A SUPERSTRUCTURE MODELING FRAMEWORK FOR PROCESS SYNTHESIS USING SURROGATE MODELS

By

Carlos A. Henao

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2012

Date of final oral examination: 04/27/12

The dissertation is approved by the following members of the Final Oral Committee:

Christos T. Maravelias, Associate Professor, Chemical engineering
Ross E. Swaney, Associate Professor, Chemical engineering
James A. Dumesic, Professor, Chemical engineering
Thatcher W. Root, Professor, Chemical engineering
Jeffrey T. Linderoth, Professor, Industrial and Systems Engineering

ABSTRACT

One of the fundamental problems in chemical engineering regards the design of processing facilities tailored for the production of specific chemical products from a specific set of raw materials. The conceptual development of these facilities requires the integration of elemental processing units, each of them designed to perform specific physicochemical transformations, into an optimal processing system capable of fulfilling production requirements while acknowledging a number of constraints including raw material availability, operational safety, environmental regulations etc.

In order to address this conceptual "process synthesis problem", several strategies have been developed over the years. In a traditional approach, the entire process is decomposed into a number of sub-systems (e.g. reaction network, separation network, heat exchanger network) whose design is addressed separately following a natural hierarchy [1]. For each subsystem, heuristic rules can be used to guide the generation of proper solutions. This approach is popular since it reduces the problem complexity (via decomposition), while incorporating engineering judgment (heuristics). However, it has been shown that more systematic methodologies can lead to better designs. One of such approaches is called "Superstructure Optimization" [2]. Here, a complex process diagram including all potentially useful processing units and interconnections is initially proposed, and an optimization model featuring a suitable objective function as well as mathematical models for all units and connections is developed. The solution of this model identifies the best process structure and operational conditions. Since all aspects of the design (both structural and operational) are considered simultaneously, better solutions are generally identified.

This thesis is focused on the development of a novel methodology for the formulation of superstructure optimization models to support the solution of process synthesis problems. It includes a new way to represent and generate process superstructures, as well as all the elements required to formulate complete superstructure optimization models using an entirely modular approach and standard processing unit models. The models for all superstructure elements (i.e. processing units and connectivity elements) are created from detailed simulation models, where complex non-linear equation blocks are replaced by general non-linear surrogate mappings capable of fitting realistic process data. This approach leads to realistic superstructure models which can be solved to optimality given their reduced mathematical complexity. A novel approach to the design of these hybrid surrogate models and the implementation details in the generation of process data, as well as the fitting required to obtain such surrogate mappings are also discussed.

Several illustrative examples and case studies are included to demonstrate how the proposed surrogate modeling approach can be superior to traditional modeling based on simplifications (e.g. linear approximations, the use of idea thermodynamics, etc.), producing more realistic and significant results. This combination of realism and reduced mathematical complexity allows the formulation of tractable optimization modes which are generally larger than tractable models based exclusively on detailed unit models.

ACKNOWLEDGMENTS

With this research project I have had the privilege and the delight of working closely with my advisor, Professor Christos Maravelias. No words can express my gratitude to him for his support, understanding and patience during some difficult times, as well as the leadership and technical insight which greatly contributed to the completion of this work. More than a mentor to me, Christos is the perfect example of what you can accomplish trough dedication and the hard work that comes naturally from loving profoundly what you do.

Thanks to all the excellent professors whose courses I had the opportunity to take. Specially Professors Thatcher Root, James Rawlings, Michael Ferris, Stephen Wright and Jeffrey Linderoth , who made taking classes a real pleasure with their contagious enthusiasm inside and outside the classroom and their profound dedication to teaching. To my classmates, in particular Pratik Pranay, Mahdi Namazifar, and Dejit Roy, for their friendship and all the very productive and interesting conversations we had regarding research.

The constructive comments and critiques provided by Professor Ignacio Grossman, Professor Larry Biegler and Mark Diachendt are most appreciated.

I also would like to thank Professors James Dumesic and Manos Mavrikakis, along with their students Drew Braden and Lars Grabow for all the help and insights provided during the development of some joint projects. Many thanks to all the UW Chemical Engineering Department

staff, especially Mary Diaz and Donna Bell for their diligence and their willingness to help with all the administrative details.

Of course, none of this would have been possible without the encouragement of my parents Guillermo and Maria Elena, as well as the love patience and moral support of my wife Katerina who decided to leave her Job in our country and follow me in this adventure. Finally, I would like to thank my parents-in-law Nelson and Edilia who have been invaluable in helping my wife and me with our newborn baby Isabel during the last couple of months, allowing me to dedicate the necessary time to finish this thesis.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

NOTATION

Indices and sets

c	$\in$	$\mathbf{C}$	: Components
k	$\in$	$\mathbf{K}$	: Selection set, or network layer
p	$\in$	$\mathbf{P}$	: Ports
r	$\in$	$\mathbf{R}$	: Reaction
s	$\in$	$\mathbf{S}$	: Streams
u	$\in$	$\mathbf{U}$	: Units
$\mathbf{C}*$	$\subset$	$\mathbf{C}$	: Components in $\mathbf{C}$ excluding the last one
$\mathbf{P}^{\mathbf{I}}$	$\subset$	$\mathbf{P}$	: Inlet ports
$\mathbf{P}^{\mathbf{O}}$	$\subset$	$\mathbf{P}$	: Outlet ports
$\mathbf{P}_u^{\mathbf{I}}$	$\subset$	$\left(\mathbf{U},\mathbf{P}^{\mathbf{I}}\right)$	: Inlet ports of unit $u \in \mathbf{U}$
$\mathbf{P}_u^{\mathbf{O}}$	$\subset$	$\left(\mathbf{U},\mathbf{P}^{\mathbf{O}}\right)$	: Outlet ports of unit $u \in \mathbf{U}$
$\mathbf{P}_u$	$\subset$	$\left(\mathbf{U},\mathbf{P}\right)$	: Ports of unit $u \in \mathbf{U}$
$\mathbf{P}_s$	$\subset$	$\left(\mathbf{S},\mathbf{P}\right)$	: Ports connected by stream $s \in \mathbf{S}$
$\mathbf{R}_u$	$\subset$	$\left(\mathbf{U},\mathbf{R}\right)$	: Reactions $r \in \mathbf{R}$ taking place in unit $u \in \cup\left(\mathbf{U}^{\mathbf{SR}},\mathbf{U}^{\mathbf{PR}}\right)$

$\mathbf{S} \subset \left(\mathbf{P^O}, \mathbf{P^I}\right)$: Streams connecting units

$\mathbf{S^C} \subset \mathbf{S}$: Conditioning streams

$\mathbf{S^{NC}} \subset \mathbf{S}$: Non conditioning streams

$\mathbf{S}^{\mathbf{I}}_{p} \subset \left(\mathbf{P^I}, \mathbf{S}\right)$: Streams connected to inlet port $p \in \mathbf{P^I}$

$\mathbf{S}^{\mathbf{O}}_{p'} \subset \left(\mathbf{P^O}, \mathbf{S}\right)$: Streams connected to outlet port $p' \in \mathbf{P^O}$

$\mathbf{S}^{\mathbf{I}}_{u} \subset \left(\mathbf{U}, \mathbf{S}\right)$: Streams entering unit $u \in \mathbf{U}$

$\mathbf{S}^{\mathbf{O}}_{u} \subset \left(\mathbf{U}, \mathbf{S}\right)$: Streams leaving unit $u \in \mathbf{U}$

$\mathbf{U^{SC}} \subset \mathbf{U}$: Raw material source units

$\mathbf{U^{SK}} \subset \mathbf{U}$: Final product sink units

$\mathbf{U^{MX}} \subset \mathbf{U}$: Mixer-Splitter units

$\mathbf{U^{EV}} \subset \mathbf{U}$: Expansion valves

$\mathbf{U^{CE}} \subset \mathbf{U}$: Compression-Expansion units

$\mathbf{U^{HC}} \subset \mathbf{U}$: Heating-Cooling units

$\mathbf{U^{MS}} \subset \mathbf{U}$: Membrane separator

$\mathbf{U^{FV}} \subset \mathbf{U}$: Flash vessels

$\mathbf{U^{AC}} \subset \mathbf{U}$: Absorption columns

$\mathbf{U^{EC}} \subset \mathbf{U}$: LL extraction columns

$\mathbf{U^{DC}} \subset \mathbf{U}$: Distillation columns

$\mathbf{U^{SR}} \subset \mathbf{U}$: Stirred tank reactors

$\mathbf{U^{PR}} \subset \mathbf{U}$: Plug flow reactors

Variables

Am_u	[m^2]	: Membrane area in unit $u \in \mathbf{U}^{\mathbf{MS}}$ (≥0)
AFm_u	[m^2.h/kmol]	: Membrane area to inlet flow ratio in unit $u \in \mathbf{U}^{\mathbf{MS}}$ (≥0)
Br_u	[---]	: Boil-up ratio unit $u \in \mathbf{U}^{\mathbf{DC}}$ (≥0)
Cc_u	[USD]	: Capital cost of unit $u \in \mathbf{U}$ (≥0)
Cc_s	[USD]	: Capital cost of stream $s \in \mathbf{S}$ (≥0)
Co_u	[USD/yr]	: Operative cost of unit $u \in \mathbf{U}$ (≥0)
Co_s	[USD/yr]	: Operative cost of stream $s \in \mathbf{S}$ (≥0)
$\underline{\Delta P}_u$	[kPa]	: Pressure drop of unit $u \in \bigcup(\mathbf{U}^{\mathbf{EV}}, \mathbf{U}^{\mathbf{MS}}, \mathbf{U}^{\mathbf{FV}}, \mathbf{U}^{\mathbf{SR}}, \mathbf{U}^{\mathbf{PR}})$ (≥0). Tightly constrained for all $u \in \bigcup(\mathbf{U}^{\mathbf{MS}}, \mathbf{U}^{\mathbf{FV}}, \mathbf{U}^{\mathbf{SR}}, \mathbf{U}^{\mathbf{PR}})$.
ΔP_u	[kPa]	: Pressure increase of unit $u \in \mathbf{U}^{\mathbf{CE}}$ (--)
ΔP_s	[kPa]	: Pressure increase in stream $s \in \mathbf{S}^{\mathbf{C}}$ (--)
ξ_s	[---]	: Split fraction for stream $s = (p', p) \in \mathbf{S}$ leaving port $p' \in \mathbf{P}^{\mathbf{O}}$ (≥0, ≤1)
F_p	[kmol/h]	: Molar flow in port $p \in \mathbf{P}$ (≥0)
$F_{c,p}$	[kmol/h]	: Molar flow of component $c \in \mathbf{C}$ in port $p \in \mathbf{P}$ (≥0)
$F^k_{c,p,s}$	[kmol/h]	: Disaggregated flow of component $c \in \mathbf{C}$ in port $p \in \mathbf{P}$ (≥0)
F_s	[kmol/h]	: Molar flow in stream $s \in \mathbf{S}$ (≥0)
$F_{c,s}$	[kmol/h]	: Molar flow of component $c \in \mathbf{C}$ in stream $s \in \mathbf{S}$ (≥0)
$F^k_{c,s}$	[kmol/h]	: Disaggregated flow of component $c \in \mathbf{C}$ in stream $s \in \mathbf{S}$ (≥0)

$\phi_{c,u}$	[---]	: Splitting fraction of component $c \in \mathbf{C}$ in unit $u \in \cup\left(\mathbf{U^{MS}}, \mathbf{U^{FV}}, \mathbf{U^{AC}U^{EC}}, \mathbf{U^{DC}}\right)$ (≥0)
H_p	[kJ/h]	: Enthalpy flow of port $p \in \mathbf{P}$ (--)
h_p	[kJ/kmol]	: Molar enthalpy of port $p \in \mathbf{P}$ (--)
$H_{p,s}^{k}$	[kJ/h]	: Disaggregated enthalpy flow of port $p \in \mathbf{P}$ (--)
H_s^{S1}	[kJ/h]	: Enthalpy flow of stream $s \in \mathbf{S}$ before conditioning (--)
h_s^{S1}	[kJ/kmol]	: Molar enthalpy of stream $s \in \mathbf{S}$ before conditioning (--)
$H_s^{\mathrm{S1},k}$	[kJ/h]	: Disaggregated enthalpy flow of stream $s \in \mathbf{S}$ before conditioning (--)
H_s^{S2}	[kJ/h]	: Enthalpy flow of stream $s \in \mathbf{S^C}$ after temperature conditioning (--)
h_s^{S2}	[kJ/kmol]	: Molar enthalpy of stream $s \in \mathbf{S^C}$ after temperature conditioning (--)
H_s^{S3}	[kJ/h]	: Enthalpy flow of stream $s \in \mathbf{S}$ after pressure conditioning (--)
h_s^{S3}	[kJ/kmol]	: Molar enthalpy of stream $s \in \mathbf{S}$ after pressure conditioning (--)
N_u	[---]	: Number of stages in unit $u \in \cup\left(\mathbf{U^{AC}}, \mathbf{U^{EC}}\right)$ (≥0, integer)
Nr_u	[---]	: Number of rectifying stages in unit $u \in \mathbf{U^{DC}}$ (≥0, integer)
Ns_u	[---]	: Number of striping stages in unit $u \in \mathbf{U^{DC}}$ (≥0, integer)
P_u	[kPa]	: Outlet pressure of unit $u \in \cup\left(\mathbf{U^{EV}}, \mathbf{U^{CE}}, \mathbf{U^{HC}}, \mathbf{U^{MS}}, \mathbf{U^{FV}}, \mathbf{U^{AC}}, \mathbf{U^{EC}}\right.$ $\left.\mathbf{U^{DC}}, \mathbf{U^{SR}}, \mathbf{U^{PR}}\right)$ (≥0)
P_p	[kPa]	: Pressure of port $p \in \mathbf{P}$ (≥0)

P_s^{S1}	[kPa]	: Pressure of stream $s \in \mathbf{S}$ before conditioning (≥0)
P_s^{S2}	[kPa]	: Pressure of stream $s \in \mathbf{S^C}$ after temperature conditioning (≥0)
P_s^{S3}	[kPa]	: Pressure of stream $s \in \mathbf{S}$ after pressure conditioning (≥0)
Qc_u	[kJ/h]	: Condenser duty of unit $u \in \mathbf{U^{DC}}$ (≥0)
qc_u	[kJ/kmol]	: Specific condenser duty of unit $u \in \mathbf{U^{DC}}$ (≥0)
Qr_u	[kJ/h]	: Reboiler duty of unit $u \in \mathbf{U^{DC}}$ (≥0)
qr_u	[kJ/kmol]	: Specific reboiler duty of unit $u \in \mathbf{U^{DC}}$ (≥0)
Q_u	[kJ/h]	: Heating duty of unit $u \in \bigcup\left(\mathbf{U^{HC}}, \mathbf{U^{SR}}, \mathbf{U^{PR}}\right)$ (--)
q_u	[kJ/kmol]	: Specific unit heating duty of unit $u \in \bigcup\left(\mathbf{U^{HC}}, \mathbf{U^{SR}}, \mathbf{U^{PR}}\right)$ (--)
Q_s	[kJ/h]	: Heating duty required in stream $s \in \mathbf{S^C}$ for temperature conditioning (--)
q_s	[kJ/kmol]	: Specific heating duty required in stream $s \in \mathbf{S^C}$ for temperature conditioning (--)
$\bar{r}_{r,u}$	[kmol/h.m^3]	: Average reaction rate of reaction $r \in \mathbf{R}_u$ in unit u (--)
S_p	[kJ/h.K]	: Entropy flow of port $p \in \mathbf{P}$ (--)
s_p	[kJ/kmol.K]	: Molar entropy of port $p \in \mathbf{P}$ (--)
$S_{p,s}^k$	[kJ/h.K]	: Disaggregated entropy flow of port $p \in \mathbf{P}$ (--)
S_s^{S1}	[kJ/h.K]	: Entropy flow of stream $s \in \mathbf{S}$ before conditioning (--)
s_s^{S1}	[kJ/kmol.K]	: Molar entropy of stream $s \in \mathbf{S}$ before conditioning (--)
$S_s^{S1,k}$	[kJ/h]	: Disaggregated entropy flow of stream $s \in \mathbf{S}$ before

conditioning (--)

S_s^{S2} [kJ/h.K] : Entropy flow of stream $s \in \mathbf{S^C}$ after temperature conditioning (--)

s_s^{S2} [kJ/kmol.K] : Molar entropy of stream $s \in \mathbf{S^C}$ after temperature conditioning (--)

S_s^{S3} [kJ/h.K] : Entropy flow of stream $s \in \mathbf{S}$ after pressure conditioning (--)

s_s^{S3} [kJ/kmol.K] : Molar entropy of stream $s \in \mathbf{S}$ after pressure conditioning (--)

T_u [K] : Outlet temperature of unit $u \in \bigcup(\mathbf{U^{EV}}, \mathbf{U^{CE}}, \mathbf{U^{HC}}, \mathbf{U^{MS}}, \mathbf{U^{FV}}, \mathbf{U^{SR}}, \mathbf{U^{PR}})$ (≥0)

T_p [K] : Temperature of port $p \in \mathbf{P}$ (≥0)

T_s^{S1} [K] : Temperature of stream $s \in \mathbf{S}$ before conditioning (≥0)

T_s^{S2} [K] : Temperature of stream $s \in \mathbf{S^C}$ after temperature conditioning (≥0)

T_s^{S3} [K] : Temperature of stream $s \in \mathbf{S}$ after pressure conditioning (≥0)

$Tcond_u$ [K] : Condenser temperature of unit $u \in \mathbf{U^{DC}}$ (≥0)

V_u [m^3] : Internal equipment volume in unit $u \in \bigcup(\mathbf{U^{SR}}, \mathbf{U^{PR}})$ (≥0)

W_u [kJ/h] : Power consumption of unit $u \in \mathbf{U^{CE}}$ (--)

W_s [kJ/h] : Power consumption in stream $s \in \mathbf{S^C}$ for pressure conditioning (--)

$x_{c,p}$ [---] : Molar fraction of component $c \in \mathbf{C}$ in port $p \in \mathbf{P}$ (≥0, ≤1)

$x_{c,s}$ [---] : Molar fraction of component $c \in \mathbf{C}$ in stream $s \in \mathbf{S}$ (≥0, ≤1)

$\chi_{r,u}$ [kmol/h] : Extent rate for reaction $r \in \mathbf{R}$ in unit $u \in \bigcup(\mathbf{U^{SR}}, \mathbf{U^{PR}})$ (--)

y_u	[---]	: Activation binary for unit $u \in \mathbf{U}$
y_s^k	[---]	: Selection binary for discrete split faction ξ^k in stream $s \in \mathbf{S}$
fr_u	[---]	: Feed flow ratio for unit $u \in \cup\left(\mathbf{U}^{\mathbf{AC}}, \mathbf{U}^{\mathbf{EC}}\right)$ (≥0)
ψ_u	[---]	: Internal variables of general unit $u \in \mathbf{U}^{\mathbf{G}}$ (≥0)

Parameters

ξ^k	[---]	: "kth" discrete split fraction value (≥0, ≤1)
T^{Lo}	[K]	: Temperature lower bound (≥0)
T^{Up}	[K]	: Temperature upper bound (≥0)
F^{Lo}	[kmol/h]	: Molar flow lower bound (≥0)
F^{Up}	[kmol/h]	: Molar flow upper bound (≥0)
H^{Lo}	[kJ/h]	: Enthalpy flow lower bound (--)
H^{Up}	[kJ/h]	: Enthalpy flow upper bound (--)
S^{Lo}	[kJ/K.h]	: Entropy flow lower bound (--)
S^{Up}	[kJ/k.h]	: Entropy flow upper bound (--)
$\nu_{c,r}$	[---]	: Stoichiometric coefficient of component $c \in \mathbf{C}$ in reaction $r \in \mathbf{R}$ (--)

Chapter 1

INTRODUCTION

1.1 Process synthesis problem

One of the fundamental problems in Chemical Engineering regards the conceptual design of continuous processing facilities, henceforth called the process synthesis problem (PSP). This problem is solved by making a series of engineering decisions leading to the definition of a processing system that allows an efficient transformation of available raw materials into a number of valuable products, while satisfying different types of constraints such as environmental and safety regulations, operational flexibility requirements etc. Such engineering decisions can be classified in two broad classes: structural decisions (including the selection of processing units and interconnections defining the process topology) and operational decisions (including the selection of unit operating conditions i.e. temperatures, pressures, compositions, flows, etc.)

The PSP is very complex. In fact, it is combinatorial in nature due to the large number of decisions involved and the number of alternatives available for each decision. For example, considering only the discrete structural decisions, it is common to have 10^{15} alternatives [2]. In order to deal with this

level of complexity, several systematic approaches have been developed over the years. The work presented here is framed in the context of the so called superstructure optimization approach.

1.2 Superstructure optimization

Solving the PSP using superstructure optimization (SO) begins with the creation of a "superstructure", that is, a process diagram including all potentially useful processing units and all relevant interconnections. This is in fact a highly redundant process diagram where, for each processing function required to transform raw materials into products, several competing units, and even entire processing subsystems, have been included and interconnected with the rest. Based on this superstructure, a proper optimization model is then formulated, including an adequate objective function and a set of equations and inequalities that establishes relationships between the variables used to describe the behavior of the unit operations and interconnections (e.g. unit operation modes, interconnection equations, raw material availability constraints, product quality requirements, emission requirements, etc.) In general, such optimization model includes both continuous variables (e.g. temperatures, pressures, component flows, equipment size, etc.) used to describe the way the units behave, and discrete variables (mostly binaries) used to select a subset of the units and interconnections within the original superstructure, hence defining a structural alternative. The selected and excluded units are respectively referred to as "activated" and "deactivated" units. By solving this optimization model, all process structures (i.e. structural alternatives) embedded in the original superstructure are evaluated, and the best one is identified along with the optimal operational conditions for its units. This solution can be seen as a simplified version of the original superstructure, capable of performing the required transformation of raw material to products in the most convenient way (as defined by the objective function) while satisfying all required constraints. These ideas are represented in Figure 1.

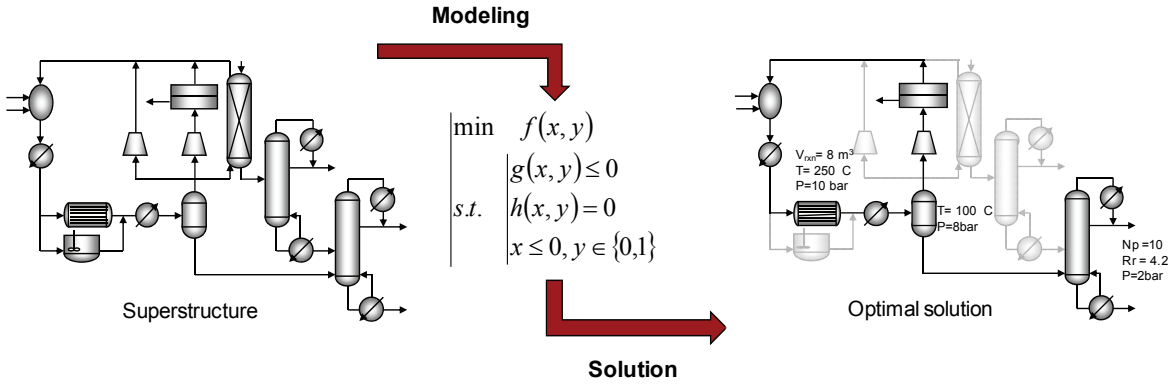


Figure 1: Superstructure optimization approach to the solution of process synthesis problems. The final solution identifies a subset of units and interconnections in the original superstructure, as well as the optimal process operating conditions.

The obvious advantage of this approach is its ability to consider all complex interaction between all design decisions, since all of them (structural and operational) are made simultaneously. However, this methodology also has a couple of important drawbacks. First, the "absolute" optimal solution cannot be obtained if its structure is not covered by the original superstructure; hence in principle, rich and complex superstructures have to be used in order to define a wide search space. Second, in many cases, realistic unit operation models have to be used as part of the optimization model in order to obtain valuable results. This increases significantly the mathematical complexity of the resulting optimization model (generally a large scale non-convex Mixed Integer Non Linear Program -MINLP- as described in section 2.1), making it computationally intractable.

1.3 Challenges in superstructure optimization and scope of this work

The advantages of SO have been successfully used to find enhanced solutions for many practical process synthesis problems. Such success stories mainly regard the study of relatively homogeneous processing systems for which systematic superstructure representations have been devised (i.e.

systems composed by one type of units, such reactor networks [3],[4], separator networks [5],[6],[7], heat exchanger networks [8]) and/or systems using simplified unit operation models (e.g. stoichiometric reactors with fixed advances, separation units using ideal thermodynamics, linear approximations etc.) However, the practical application of SO techniques has been somehow limited in the case of general reaction-separation PSPs, where large and complex superstructures as well as detailed unit operation models are required to obtain realistic and significant solutions. This is due to the mathematical complexity inherent to detailed unit models, and the use of ad hoc modeling strategies as opposed to a modular modeling approach in which standardized unit models can be integrated using general connectivity elements. In this way, the main challenge in SO still regards the *systematic* creation of *rich* and *realistic* superstructure models that are also *computationally tractable*.

To obtain rich superstructures for a given set of unit operations, we propose the use of extended connectivity. Here, it is important to include only potentially useful connections, or equivalently, to exclude counterproductive ones. Figure 2 two different superstructures for an amine based CO_2 capture system.

The split flow connectivity shown in Figure 2 a) includes interconnections between the mid sections of the absorber and the regeneration towers. This has shown to have thermodynamic advantages, hence reducing the utility consumption. To explore other potentially useful interactions between the columns, the superstructure Figure 2 b) is created by using an extended connectivity between four key points in each of the columns.

To formulate realistic but still solvable superstructure models, we propose the use of surrogate models based on detailed processing unit models. Here, the selection of the proper surrogate type is

very import, affecting both the model prediction capabilities and its mathematical complexity. In general, the selection should balance these aspects, providing a mathematical form which can be exploited by current numerical solvers or tailored solution algorithms.

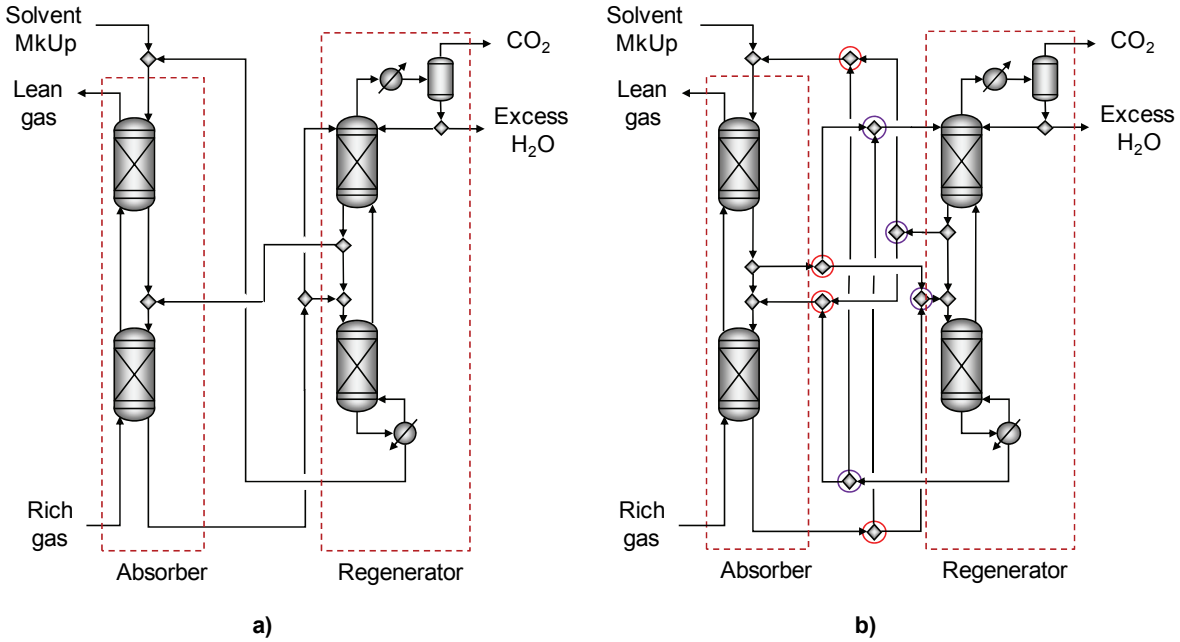


Figure 2: Amine based CO_2 capture system. **a)** Traditional split flow configuration **b)** Superstructure with extended connectivity. Absorber interconnection ports (red circled), regenerator interconnection ports (purple circled).

Consider the detailed model for a packed bed reactor presented in Figure 3. From the solution point of view, most of the complexity regards the kinetics, and the calculation of transport and thermodynamic properties. For illustration purposes, consider the problem of finding the optimal operating conditions of a methanol reactor using the following stoichiometry:

$$CO_2 + 3 \cdot H_2 \Rightarrow MeOH + H_2O \qquad (\text{Methanol Synthesis reaction})$$
$$CO + H_2O \Rightarrow CO_2 + H_2 \qquad (\text{Water Gas Shift})$$

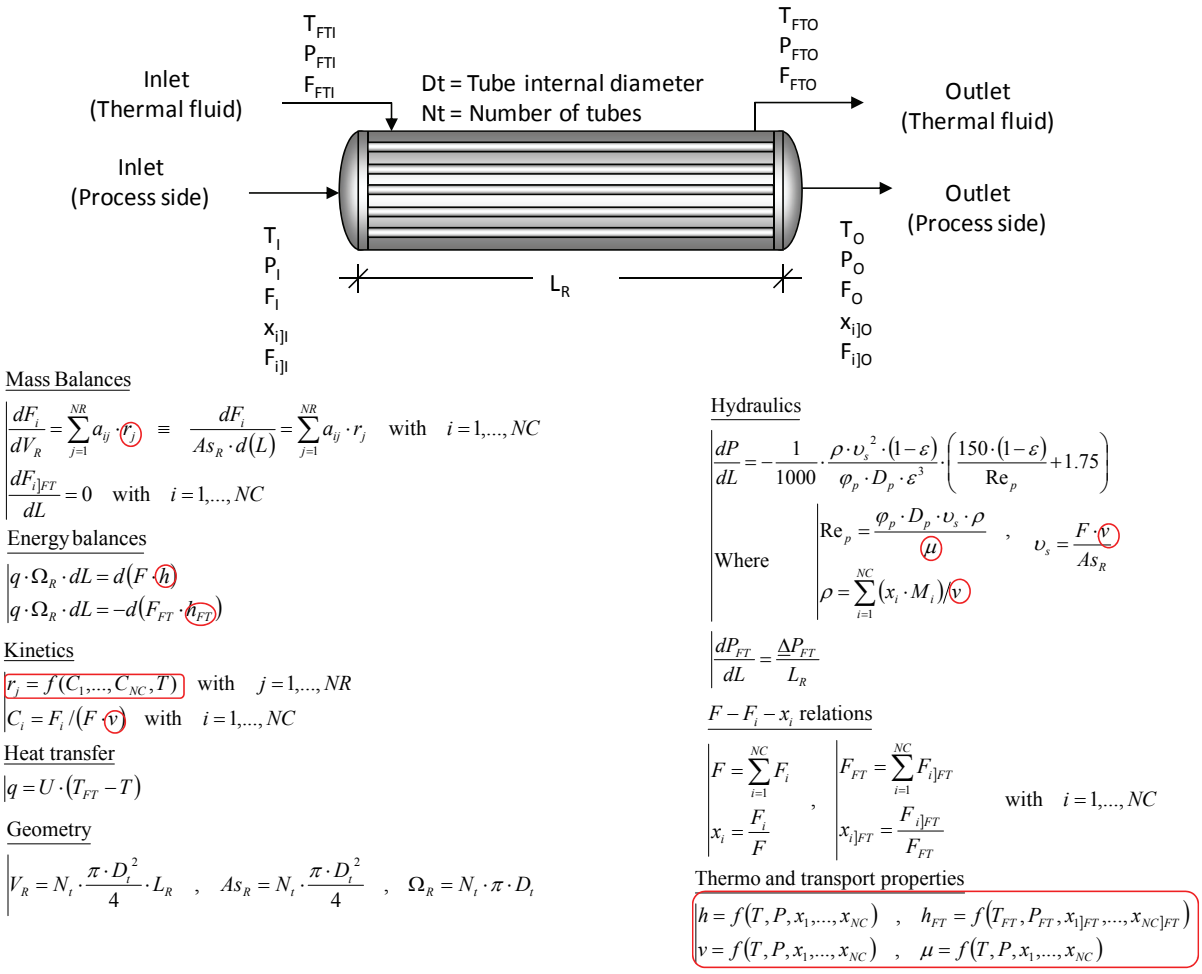


Figure 3: Detailed model for a packed bed reactor. The model includes realistic kinetics, thermodynamics and hydraulics. The most complex equations in the model are highlighted.

From the optimization point of view, we are mostly interested in the way important variables such as revenue (a function of the methanol production), capital cost (a function of the unit size) and operating cost (a function of the utility and raw material consumption) are linked to the model degrees of freedom. The values taken by many of the model dependent variables are irrelevant (e.g. internal temperatures, pressures and component flow profiles.).

The optimization model could be simplified by replacing the equations in Figure 3 with a small set of mappings relating only relevant variables. Figure 4 shows a set of these mappings for a simple case where the reactor is adiabatic with a fixed geometry, and the feed component flows are fixed. Here, the only degrees of freedom left are the feed temperature and pressure, while important variables include the outlet stream temperature pressures and flows.

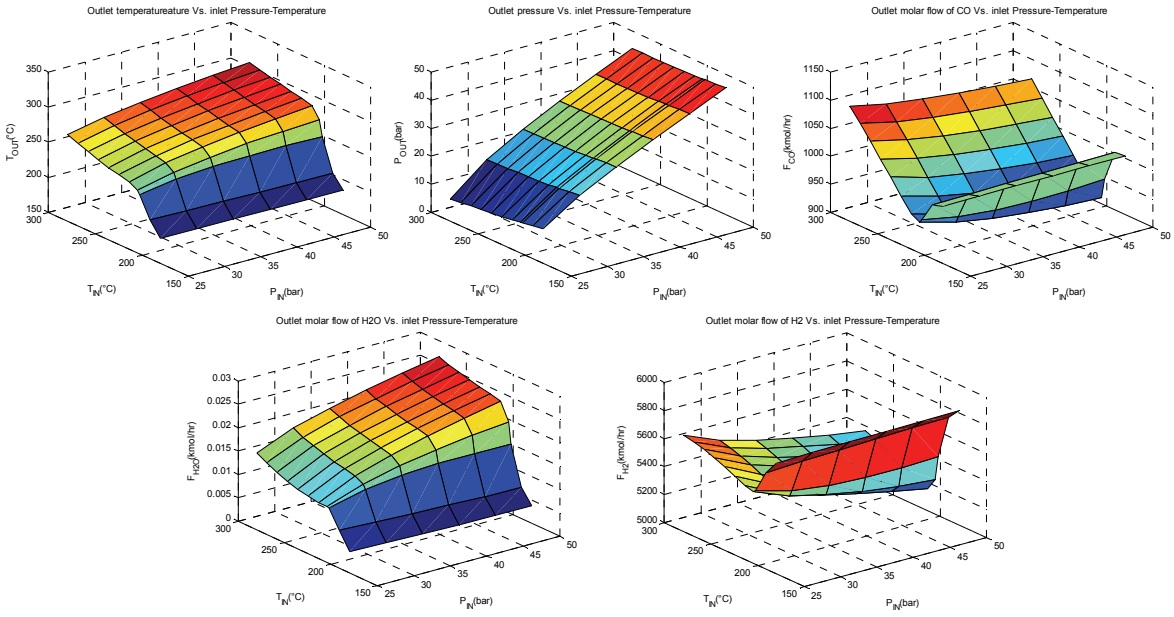


Figure 4: Key mappings relating the model degrees of freedom and important variables for a catalytic reactor used in the production of methanol from CO_2 and H_2.

As discussed in the rest of this thesis, the most important aspects in the successful application of SO techniques include: the use of a proper superstructure representation, the systematic generation of superstructures, the proper modeling of superstructure elements (unit operations, connectivity etc.), and the use of proper numerical techniques to solve the resulting models. In this thesis we have developed a procedure to represent and generate superstructures with extended but meaningful connectivity as part of a framework to solve general reaction-separation PSPs. We also

present a novel modular approach to chemical process superstructure modeling, which allows a fast initial formulation and later modification of complete models. In addition, the proposed strategy integrates non-linear surrogate mappings to replace complex unit equation blocks, reducing the mathematical complexity of the entire superstructure model while imposing a convenient mathematical form which can be exploited by current numerical solvers or tailored solution algorithms.

1.4 Thesis outline

After a short literature review in chapter 2 regarding important aspects of SO, chapter 3 presents a new approach to superstructure representation and generation. The purpose of this methodology is the creation of superstructures which are rich (improving the chances of finding the optimal solution) while excluding unnecessary complexity trough the incorporation of engineering judgment.

Chapter 4 presents our approach to superstructure modeling, including the general form of a SO model and all the elements that make it modular and flexible. These characteristics, simplify the formulation of large superstructure models from scratch, as well as the modification of existing models. Superstructure element models based on molar flows (F_c formulation) are presented, along with the general reformulation required to pose the SO model as a Mixed Integer Nonlinear Program (see section 2.1).

Chapter 5 introduces a new methodology for the design of superstructure element surrogate models. These surrogates are based on detailed element models where all complex nonlinear equation blocks have been replaced by non-linear surrogate mappings with a convenient mathematical form. A special methodology to design such hybrid surrogates is introduced, including

the selection of dependent and independent variables. For surrogate mappings base on Artificial Neural Networks, a reformulation following the ideas presented in chapter 4 is also discussed.

Chapter 6 presents how surrogate mappings are generated, including the sampling of values for independent variables, the generation of values for dependent variables, the data fitting with Artificial Neural Networks, and the validation of the resulting surrogate mappings.

Chapter 7 presents some final remarks and complementary information, including the implementation details of superstructure modeling in GAMS (General Algebraic Modeling System), the data generation using the commercial process simulator ASPEN PLUS, surrogate mapping fitting using the scientific computing environment MATLAB, an outline of the automatic updating of surrogate mapping domains, extensions on the use of surrogate models, and an outline of solution strategies based on the surrogate mapping mathematical form.

Chapter 8 includes some case studies illustrating the use of the methodologies presented in this thesis as well as their capabilities in the solution of PSPs. Finally, chapter 9 presents the main conclusions as well as some issues which could be addressed in the future as a continuation of this work.

Chapter 2

STATE OF THE ART IN SUPERSTRUCTURE OPTIMIZATION

The development of mathematical programming has made possible the use of SO as an alternative to more traditional heuristic based approaches to solve process synthesis problems. For more than three decades, contributions made by mathematicians, scientists and engineers have improved significantly the capabilities of SO techniques by addressing the key aspects mentioned at the end of section 1.3, namely, superstructure representation, superstructure generation, superstructure modeling, and numerical solution algorithms. However, as discussed in chapter 1, SO techniques have important limitations since the formulation of general modes to solve complex PSPs are often intractable.

This chapter includes a succinct presentation of important mathematical programming concepts, as well as a review on the state of the art of fundamental aspects of SO techniques.

2.1 Mathematical programming techniques

Mathematical programming as a field of study has evolved since the 1940's, providing theoretical foundations and practical numerical algorithms for the solution of different kind of problems. Important sub-areas include Linear Programming (LP), Mixed Integer Programming (MIP), Non Linear Programming (NLP), Mixed Integer Non Linear Programming (MINLP), and Global Optimization (GO).

Linear Programming studies the properties and solution of problems where a linear objective function is to be minimized (or maximized) subject to a set of linear constraints (equations and/or inequalities). Important milestones in LP includes the initial theoretical work by Leonid Kantorovich, the development of the "Simplex" solution algorithm published in 1947 by George B. Dantzig, the duality theory developed by John von Neumann, as well as an alternative interior-point solution method introduced by Narendra Karmarkar in 1984.

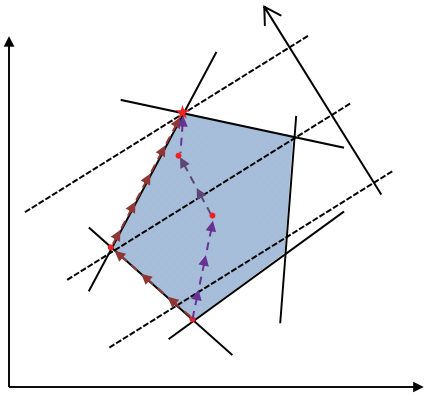


Figure 5: Graphic representation of a minimization LP problem: Feasible region (blue), objective contours (dotted lines) minimization direction (black arrow), solution (red star). Simplex path (brown), interior method path (purple).

The Simplex method is based on the fact that the solution of a feasible LP lies is one of the extreme points of its feasible region (set of points satisfying the constraints). It uses a pivoting technique to evaluate a sequence of extreme points leading to the solution [9]. In general, the total number of points evaluated is only a fraction of the finite number of extreme points in a feasible region defined by linear constraints. In interior point methods, optimality conditions for the primal and dual problems are posed and solve iteratively, producing a series of iterates lying at the interior of the feasible region. This strategy is often faster than the Simplex method when dealing with very large problems.

Mixed Integer Programming studies problems featuring a linear objective function and linear constraints that includes both continuous variables x and integer variables y. A number of important problems in management, engineering and science can be solved using this technique, including the facility location problem, scheduling and planning of batch plants, and even design of processing facilities for which linear models are applicable. Here, the basic solution technique is the "Branch and Bound" algorithm developed by Land and Doig in 1960. The algorithm is an implicit enumeration technique which solves the problems very effectively by dividing and discarding vast portions of the search space without evaluating them in detail. Consider a minimization problem. Here, the algorithm starts by solving an LP relaxation problem, also called problem 0 or root problem, which is identical to the original problem but considers variables y to be continuous (i.e. integrality constraints are dropped). The resulting solution $\left(x^0, y^0\right)$ and objective value z^0 are now analyzed. Particularly, z^0 provides a lower bound to the solution of the original one (since the feasible region of the root problem contains entirely that of the original problem) and, if y^0 is integer, the algorithm terminates. However, if at least one of the components of y^0 is not integer,

a variable among those violating the integrality condition, say y_j, is selected and the root problem is "branched" generating two sub-problems sharing the same objective and constraints in the root, but having one additional constraint each: $y_j \leq floor\left(y_j^0\right)$ for sub-problem 1, $y_j \geq ceiling\left(y_j^0\right)$ for sub-problem 2. In this way, the root problem is replaced by its branch sub-problems, which now define a combined search space excluding a region containing the solution $\left(x^0, y^0\right)$ (see Figure 6).

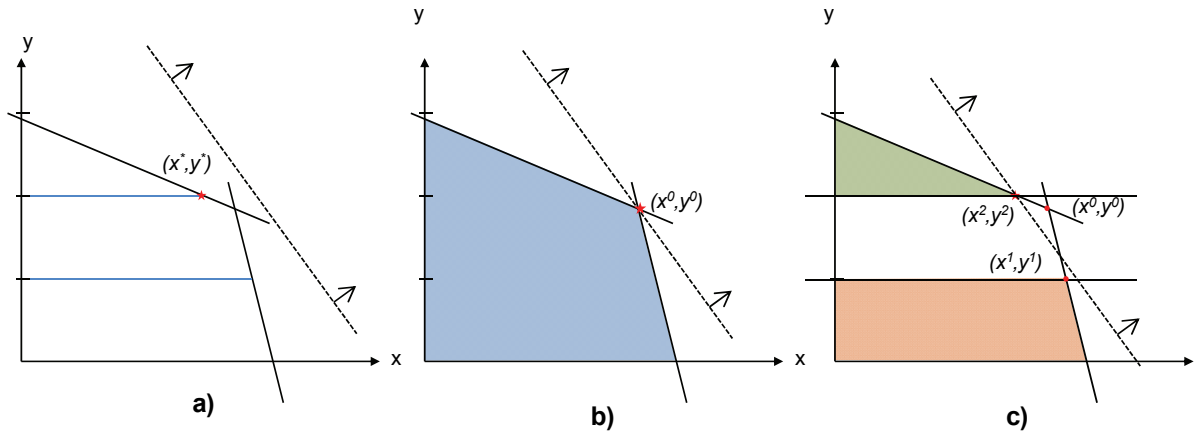


Figure 6: Graphic representation of a simple MIP and its solution using the Branch and Bound algorithm. **a)** Original problem: Constraint boundaries (solid lines), objective contour (dotted line), Feasible region (blue lines), and solution (res star). **b)** Root problem: Feasible region (blue). **c)** Branch problems: Feasible region sub-problem 1(green), feasible region sub-problem 2(orange).

The process is repeated over and over by selecting and solving the open (i.e. unevaluated) sub-problems. Graphically, this process is represented as a "search tree", where every problem is represented by a node connected by an edge to its parent problem. When the solution $\left(x^k, y^k\right)$of a node k is found to be integer (no branch sub-problems are generated) its objective value z^k is an upper bound to the real solution (since it is also a feasible solution to the original problem), and can

be used to eliminate the branches of every node whose solution objective value z^j is bigger than the best upper bound found up to that point (the smaller one). Infeasible sub-problems (nodes) can also be eliminated from the search. Normally, the algorithm terminates after running out of open sub-problems, and identifies the integer solution with the smallest objective as the solution of the original problem. Generally, the total number of LP relaxation sub-problems evaluated to find the final solution is a very small fraction of the total number of feasible integer points. This basic procedure is complemented with the use of cutting planes first proposed by Gomory in 1958, which reduce the search space in each relaxed sub-problem without eliminating integer feasible solutions, hence reducing the number of evaluated nodes [10].

Non Linear Programming includes the development of the theory and algorithms to deal with problems featuring any type of non-linearity in either the objective function or the constraints. These techniques have important applications in many fields where optimization has to be applied to nonlinear systems. For example, in protein folding, potential energy functions are minimized subject to a number of geometric and other constraints to determine the 3D structure of proteins under certain conditions. Another example is the design of aerodynamic surfaces, where engineers look for a shape with the best combination of drag and lift. In chemical engineering, process equipment design problems are also addressed using NLPs, where the dimensions and operating conditions that maximize equipment performance can be determined. In this field, theoretical foundations such as the Karush-Kuhn-Tucker (KKT) optimality conditions and duality theory have supported the development of solution techniques such as penalty methods, sequential quadratic programming and interior point methods [11]. In the case of penalty methods, constraints are brought into the objective function as penalty terms, producing an unconstrained problem for which Newton methods (Newton and Quasi-Newton), conjugate gradient, and trust region techniques can

be used. In the case of sequential quadratic programming, an auxiliary problem is solved recurrently. Such problem includes a quadratic approximation of the Lagrangian function as objective, subject to linear approximations of the original constraints around the current solution. This procedure converges to the original problem solution under certain conditions. With interior point methods, a perturbed version of KKT conditions is formulated and recurrently solved using a Newton type method. The central idea is to use a decreasing sequence of values for the perturbation parameter, so that the solution of the perturbed KKT system converges in the limit to the solution of the original KKT system identifying the final solution.

Mixed Integer Nonlinear Programming deals with nonlinear optimization problems including both continuous and integer variables. Some of the most representative MINLP solution algorithms borrow ideas from MIP and NLP, and include Branch and Bound (BB) [12],[13], Generalized Benders Decomposition (GBD) [14], Outer approximation (OA) [15], and LP/NLP [16].

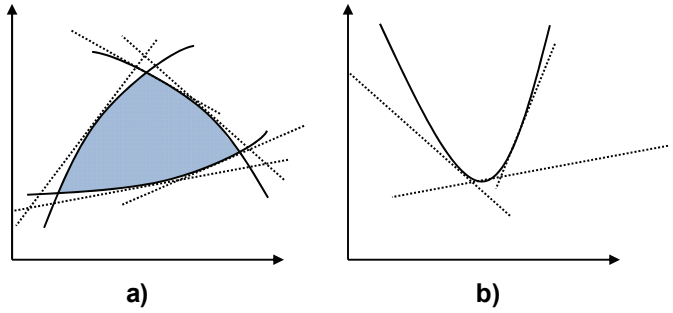


Figure 7: Linearization in the Master MINLP for OA. **a)** Feasible region. **b)** Objective function.

BB is a direct extrapolation of the method used in MIP, where every node in the search tree defines a NLP instead of an LP. OA is based on the alternate solution of a MIP master problem (generated from the accumulation of linear approximations of the objective function and constraints) which produces values for the integer variables, and an NLP sub-problem (created by fixing the integer

variables in the original problem to those values provided by the solution of the master) which provide new values for the continuous variables. LP/NLP integrates the solution tree of the master MIP and the NLP sub problems. Basically, a search tree is used to solve an initial MIP, and every time an integer solution is found, an NLP sub-problem is formulated and solve, providing a solution for which linearization of the objective function and the constraints are produced and incorporated to the MIP master without resetting its solution tree (i.e. evaluated nodes are not re-evaluated even after adding new constraints to the MIP master).

Global optimization deals with the solution of non-convex problems, but differently to other techniques which search for a local solution (i.e. a solution contained in a region for which no other feasible solution yields a better objective value), "global optimization algorithms" search for global solutions to a problem (i.e. those within the feasible region with the best possible objective value). One of the key elements in Global Optimization Algorithms is the so called spatial Branch and Bound[17] . Here, minimization models with non-convex objective functions and constraints, which usually have many local minima, are replaced by a convex relaxation models where the objective function is replace with a convex under estimator, and all non convex constraints are replaced with convex inequalities defining a convex feasible region containing the original feasible region. Similar to the Branch and Bound algorithm for MIP problems, the solution of this root relaxation problem is analyzed to produce a number of sub-problems. For simplicity, consider the simple minimization problem presented in Figure 8.

The objective function is non-convex and two local minima can be identified. The convex relaxation problem is posed by replacing the original objective function with a convex under-estimator. Solving this problem, an initial solution x^0 is found. Now, the original feasible region can be divided in two sub-regions: region 1 = $\lfloor x^L, x^0 \rfloor$ and region 2 = $\lfloor x^0, x^U \rfloor$. Two branch sub-problems can be posed by

creating objective function under-estimators for each region. The solution of these sub-problems indicates that the global solution can not be in region 1 since $z^{1,L}$ (the lower bound on the optimum solution objective in region 1) is still higher than $z^{2,U}$ (a particular solution objective and an upper bound to the optimal solution in region 2). Hence, region 1 can be discarded. The search continues in region 2, which now can be divided in region 3 = $\lfloor x^0, x^2 \rfloor$ and region 4 = $\lfloor x^2, x^U \rfloor$.

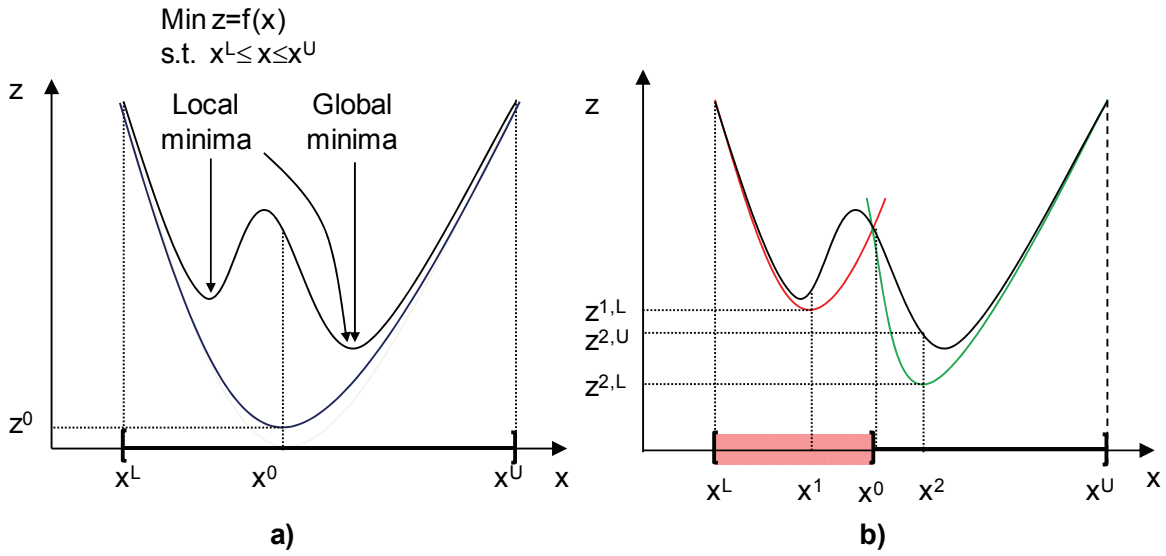


Figure 8: Simple unconstrained non-convex minimization problem and its solution using spatial branch and bound. **a)** Objective function (black line), feasible region (highlighted interval), objective convex under-estimator (blue line). **b)** First branching: under-estimator for region 1 (red line), under-estimator for region 2 (green line). Region 1 eliminated since a lower bound for the objective is greater than an upper bound for the function in region2.

2.2 Superstructure representations

During the development of SO techniques, researchers have found a clear connection between the way superstructures are represented and the way the corresponding optimization models are formulated, which ultimately affects their numerical tractability [18]. This is why the use of a correct representation is crucial to the successful implementation of SO.

Two commonly used superstructure representations, which have been found to be complementary of each other, are the State Task Networks (STN) and the State Equipment Networks (SEN) [19]. Three fundamental elements are used to build these networks: States, Tasks and Equipment. 'States' are the set of physicochemical properties defining process streams. 'Tasks' are the processing steps required to transform states into other states. For example, the reaction task A→B+C, allows a state where only component A is present to be transformed into states where potentially all three components are present. The separation task (A)/(B,C) allows a state with components A,B and C to be transformed in two distinct states: one where only component A is present, and one where only components B,C are present. Finally, 'equipment' corresponds to specific physical artifacts required to perform the tasks (e.g. PFR reactor, Tray distillation column, etc.)

STNs where initially used to represent batch process scheduling problems [20], but they are equally useful in the representation of continuous processes. Here, tasks can be seen as intermediate processing steps between two or more adjacent states. States can be seen as connections between adjacent tasks, and equipment regards the hardware required to fulfill such tasks [20]. In order to build an STN, a set of tasks required to define one or more paths connecting raw material states with final product states is initially identified. Later, a set of equipment able to perform the whole set of tasks is also identified and included in the representation. When a task can be performed by

more than one piece of equipment, the resulting representation is referred to as “State Task Network – Variable Task Equipment” (STN-VTE). When each task can be performed only by one piece of equipment, the representation is referred to as “State Task Network – One Task /One Equipment” (STN-OTOE). The SO model generated from these representations are conceived to support the systematic identification of a subset of tasks to be included in the final process design, and the equipment required to perform such tasks.

The main limitation in many implementations of this approach resides in the modeling restrictions imposed when states (hence tasks) are defined. That is, having a specific state-task in mind (e.g. the separation of a specific binary mixture to a certain degree), results in a unit model which do not allow room for alternative operability (e.g. the presence of a third component in the feed stream, or the use of a variable separation degree). For example, this is the case when the modeling of a unit is undertaken using a set of linear approximations, each of them valid within a narrow operating region (i.e. operating mode). Each approximation and its narrow operating region define a distinct task and associated states. Each task model is very simple (i.e. linear) but very restrictive, requiring the use of large numbers of tasks and states to obtain a flexible representation of each unit, and hence, an operationally flexible superstructure model (i.e. with a wide operational search space). This can lead to overly large superstructure as illustrated in Figure 9. SENs where inspired by the idea of using detailed processing unit models along with general unit connectivity to formulate process synthesis problems in a more general and realistic fashion [21]. In SEN representations, a number of pieces of equipment are fully interconnected using process streams. In a way, traditional process flow diagrams are closer to SENs than STNs. In fact, an SEN can be seen as process diagram with full connectivity, where every outlet port of every unit is connected to every inlet port of every unit. Here, contrary to what happens in STN representations, every unit (i.e. equipment) is assumed

to be operationally flexible, being able to perform a variety of tasks depending on its design and the way it is operated. There is no pre-assignment of tasks to equipment or states to process streams. In order to build an SEN, a set of equipment capable of performing all required tasks to transform raw materials into final product is initially selected, and fully connected. The corresponding SO model is posed to support the systematic identification of a subset of equipment and interconnections to be included in the final process design, as well as the proper assignments of tasks to equipment.

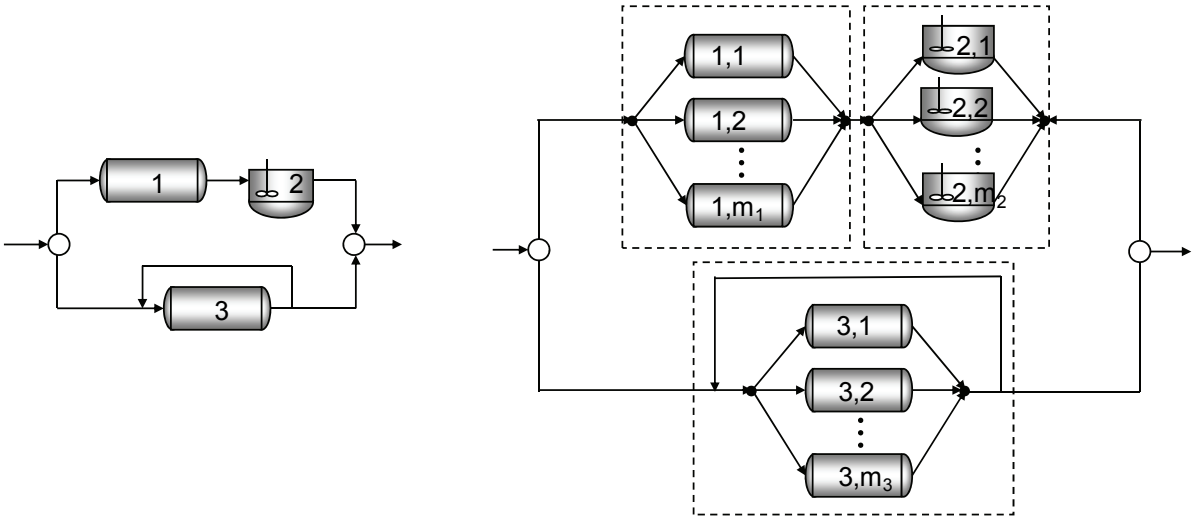


Figure 9: STN superstructure. **a)** Simple processing system. **b)** STN based on a number of possible tasks performed by each unit.

Due to the full connectivity and the flexible unit operability, a particular piece of equipment in an SEN is indistinguishable from any other of the same type (e.g. any CSTR reactor looks identical to any other CSTR in the SEN) since each of them have the same connectivity, and they all are described by an identical sets of equations. This indistinguishability reduces the combinatorial explosion characteristic of other representations, since the structural decisions are reduced to the selection of

the number of units of each type to include in the final solution, instead of the selection of units to include. In other words, for a superstructure with 'N_j' units of type 'j', an SEN representation includes $\prod_j(N_j+1)$ structural alternatives (since you can include 0,1,...,N_j units of type 'j' in your solution), and every other representation where units of type 'j' are distinguishable leads to $2^{\sum j(Nj+1)}$ alternatives (since the selection or exclusion of each unit has to be decided) [21].

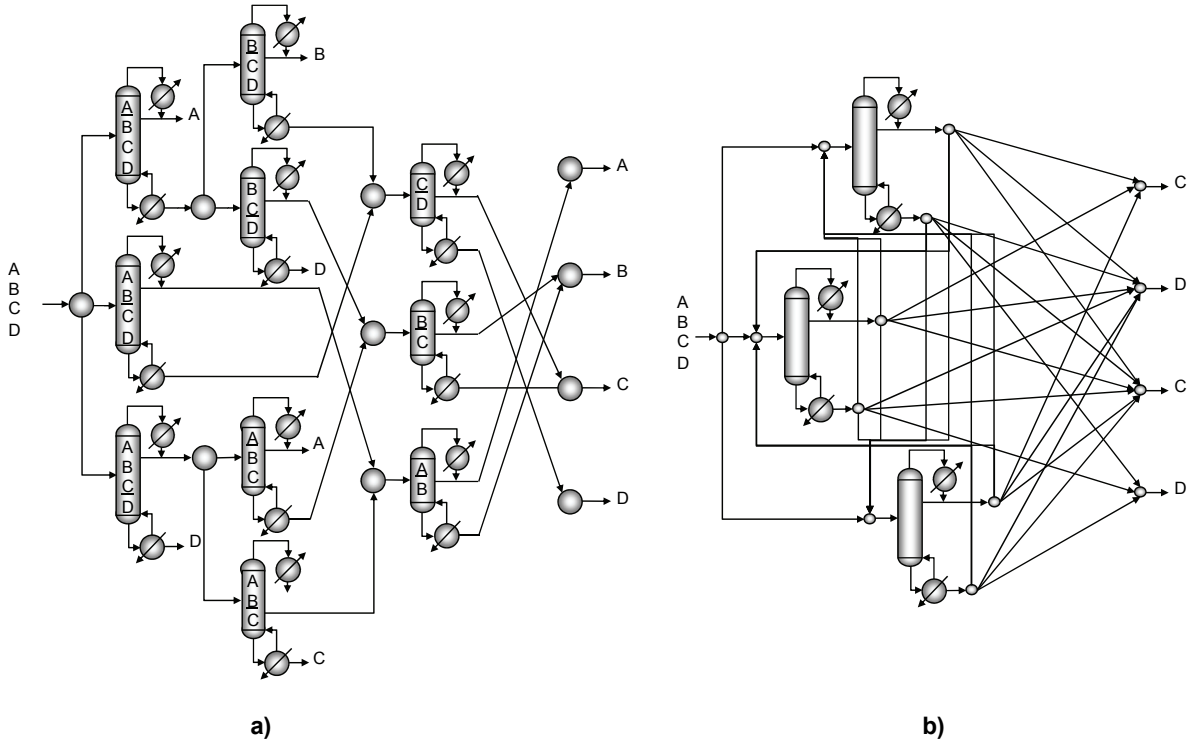


Figure 10: Separation system for a mixture of 4 components. **a)** STN representation. **b)** Highly connected SEN representation.

SENs seem to have an advantage over STNs, since they generally lead to optimization formulations which are more compact from the unit modeling point of view; that is, the model for every piece of equipment is more flexible, covering a number of tasks. However, the full connectivity required in SENs to properly account for all possible ways a particular piece of equipment could be connected to the rest, leads to significantly more connecting equations.

Other researchers have proposed alternative representations which can be seen as derivations or enhancements of the two fundamental approaches just discussed. Of particular interest is the P-graph representation [22] composed by two types of nodes, material and operation nodes. Here, the concept of 'material' is similar to the concept of 'state' but the former is fully defined by the intensive properties of the mixture it represents, excluding any of the extensive quantities included in the latter. The concept of 'operation' is very close to the concept of 'task', and each operation is defined by a pair (α,β) where α is the subset of materials entering the operation and β is the subset of materials leaving the operation. Due to the way operations are defined, the connectivity is implicitly determined by two types of arcs: connections of materials to operations and connections of operations to materials. The advantage of P-graphs over other representations come from an increased structural semantic and syntactic content, being able to uniquely represent any process structure, as opposed to traditional process diagram representations where a particular diagram can represent a number of structures [22]. To illustrate this point consider two different processes. In the first one, raw material A is transformed by unit 1 into material C, and raw material B is transformed by unit 2 into material D. Then, materials C and D are transformed by unit 3 into final product E. In the second process, both units 1 and 2 generate only material C, which is now transformed by unit 3 into final product E. A conventional directed graph representation (where nodes are units and streams are arcs) is identical for these processes (see Figure 11-a). On the other hand, P-graphs (where materials and units are both nodes of different types) provide unique representations for these two processes (see Figure 11-b and c).

As discussed in the next section, P-graphs are also advantageous since they are part of a systematic approach to generate superstructures. In this approach, the PSP, individual structures and superstructures are formally defined in mathematical terms, while a set of axioms and theorems

establish a strong foundation for a systematic algorithm to create compact superstructures for a given PSP [22],[23].

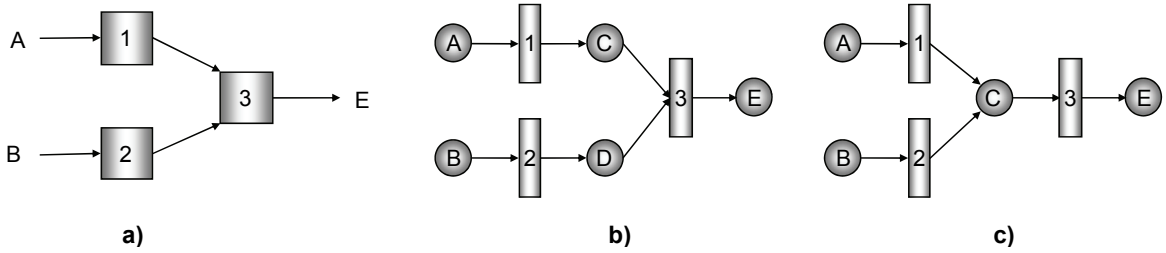


Figure 11: Simple process representation **a)** Directed graph representation. **b)**, **c)** P-graph representations.

Another interesting approach is the R-graph representation [24], a type of bipartite graph where vertices represent the inlet and outlet ports to the processing units in the superstructure, and directed arcs represent process streams going from outlet ports to inlet ports. This graphical representation looks very similar to traditional process diagrams. Here, every outlet port is regarded as a general stream divider element and every inlet port is regarded as a general mixing element.

These graphs have been used in the development superstructure representations where the phenomena of structural multiplicity (i.e. the existence of many sub-graphs in a superstructure graph which represent the same structural alternative)[24],[25] is minimized. In particular, structural multiplicity is common in traditional representations, especially in SENs, where units of the same type are only distinguishable from each other though labels, while sharing the same connectivity and the form of the models used to describe them. Obviously, it is desirable to avoid such phenomena as they generally lead to SO formulations which are unnecessarily complex. Figure 12 presents a simple example illustrating the concept of structural multiplicity.

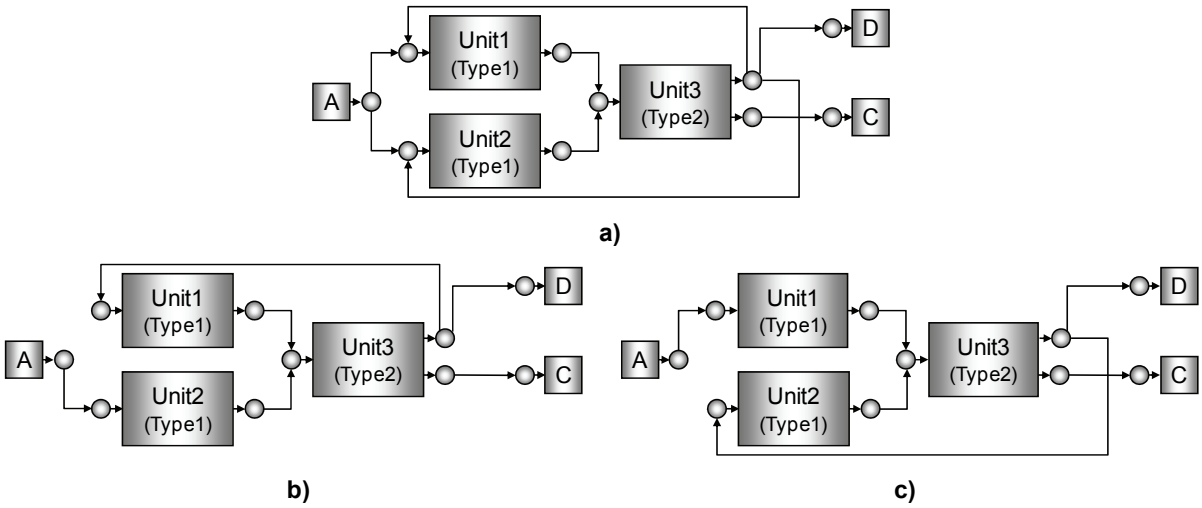


Figure 12: R-graph representation of a process diagram. **a)** Process superstructure graph. **b), c)** Two sub-graphs representing the same structural alternative (multiplicity).

2.3 Superstructure generation approaches

One of the main challenges in SO is the creation of superstructures that are as simple as possible but still include the optimal structure. Several approaches have been developed to create rich superstructures covering wide portions of the structural space, thus increasing the chances of finding the real optimum. However, only a few authors have tried to develop a systematic methodology to generate superstructures which, for a particular problem, include exclusively the set of feasible alternatives, thus producing the simplest superstructure which contains the optimal solution.

The early approaches used to create superstructures for general reaction-separation problems include the combination of simple promising structures formulated in advance using engineering judgment [18] (see Figure 13) and the combination of superstructures for plant subsystems (e.g. reaction networks, separation networks, heat recovery networks, etc.) each created via dedicated

methodologies [26]-[33]. In the first approach, the number of alternatives is limited by the experience and creativity of the designer formulating the initial set of structures; however, it is deeply rooted in engineering judgment and considers proven technologies for the most part, resulting in effective solutions but not revolutionary ones. In the second case, many alternatives are included in each of the plant subsystems, but the structural options are also limited as a result of the independent development of each substructure. In fact, this approach eliminates many of the alternatives that engineers would normally consider, as traditional limits between plant sections (i.e. reaction network, separation network, heat integration network) fade away in search for highly integrated and more efficient processes. This kind of integration leads, for example, to use a reactor in support of a separation operation by consuming a substance whose presence makes a particular mixture more difficult to process.

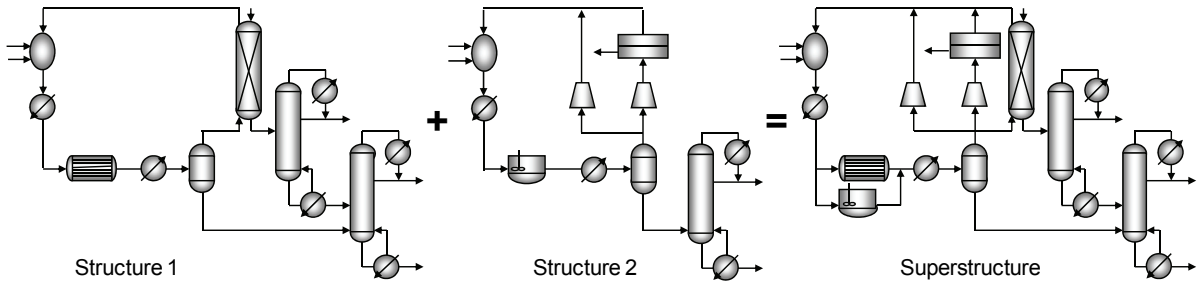


Figure 13: Superstructure created by combining two simpler structures.

The use of SEN representations reduces the problem of superstructure generation to the selection of a maximum amount of units per unit type to include, since it assumes full unit connectivity. However, even for the simpler question of identifying the most convenient set of unit operations to include in the superstructure, no systematic general procedure has been developed.

Of particular interest is the development of a complete mathematical framework for the creation of superstructures [22],[23]. This approach uses the P-graph representation described in section 2.2. Here, the PSP is defined mathematically as the triplet (P,R,O) where P is the set of products to generate, R is the set of raw materials and O is the set of operations available. As expected, the materials required to describe the operations form a superset M which contains both P and R as subsets with no common elements. It is worth noting that defining the PSP in this fashion requires the selection of an initial the set of operations (O), an aspect for which no systematic guideline has been presented, hence leaving an important question unanswered. Under this framework, the properties of a feasible process structure for a particular problem (P,R,O) are presented trough axioms, and their combinatorial properties are established through a set of theorems. The framework also presents an algorithm to generate the so called "maximal structure", defined as the union of all feasible structures; that is, the simplest superstructure which contains all relevant structural alternatives to a particular PSP. The main contribution of this approach is the mathematical formalism used to define the PSP and the systematic algorithm introduced to identify its maximal structure; however, it does not address the problem of the proper selection of operations. In other words, selecting the set of units to include in the superstructure is part of how the problem is stated, instead of being part of how the problem is solved. No real advantage comes from this.

2.4 Superstructure modeling approaches

Once a superstructure has been generated, the next step is the formulation of the corresponding structural/parametric optimization model whose solution identifies the best design. For simplicity, consider a superstructure composed by units $u \in \mathbf{U}$ with one input and one output. To allow a general connectivity between units, consider each input as a stream mixer and each output as a stream splitter (see Figure 14). Here, different variables can be used to denote the operating

conditions and the structure of the process. For the mixtures going in and out of unit u, it is common to use component molar flows $F_{c,u}^{I}$, $F_{c,u}^{O}$ temperatures T_u^{I}, T_u^{O} and pressure P_u^{I}, P_u^{O} as standard state variables. To model the units, these states variables and some internal unit variables ψ_u are used. To model stream mixers, splitters and connections between units u and u', it is common to use stream split fractions $\zeta_{u,u'} \in [0,1]$ or stream component flows $F_{c,u,u'}$. Finally, to identify structural alternatives via activation/deactivation (i.e. selection/exclusion) of units u, binary variables $y_u \in \{0,1\}$ are used.

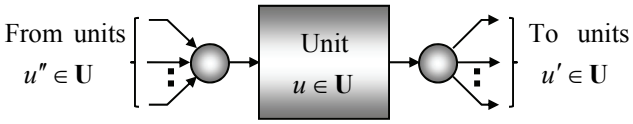


Figure 14: Graphic representation of a generic unit (one input, one output) in a superstructure, including a general inlet mixer and a general outlet splitter.

Table 1: Two alternative model formulations for a general unit including an inlet mixer and an outlet splitter

	Using stream component flows and activation binaries	**Using stream split fractions and activation binaries**
Unit Model	$\left(F_{c,u}^{O}, T_u^{O}, P_u^{O}\right) = f\begin{pmatrix} F_{c,u}^{I}, T_u^{I}, P_u^{I}, \\ \psi_u, y_u \end{pmatrix} \quad , \quad \forall u \in \mathbf{U}$	$\left(F_{c,u}^{O}, T_u^{O}, P_u^{O}\right) = f\begin{pmatrix} F_{c,u}^{I}, T_u^{I}, P_u^{I}, \\ \psi_u, y_u \end{pmatrix} \quad , \quad \forall u \in \mathbf{U}$
Unit Inlet Mixer	$F_{c,u}^{I} = \sum_{u' \in \mathbf{U}} F_{c,u',u} \quad , \quad \begin{vmatrix} \forall c \in \mathbf{C} \\ \forall u \in \mathbf{U} \end{vmatrix}$ $\left(T_u^{I}, P_u^{I}\right) = f\begin{pmatrix} F_{c,u',u}, T_{u'}^{O}, \\ P_{u'}^{O}, y_{u'} \end{pmatrix} \Big/ \forall u' \in \mathbf{U} \quad , \quad \forall u \in \mathbf{U}$	$F_{c,u}^{I} = \sum_{u' \in \mathbf{U}} F_{c,u'}^{O} \cdot \zeta_{u',u} \quad , \quad \begin{vmatrix} \forall c \in \mathbf{C} \\ \forall u \in \mathbf{U} \end{vmatrix}$ $\left(T_u^{I}, P_u^{I}\right) = f\begin{pmatrix} F_{c,u'}^{O} \cdot \zeta_{u',u}, T_{u'}^{O}, \\ P_{u'}^{O}, y_{u'} \end{pmatrix} \Big/ \forall u' \in \mathbf{U} \quad , \quad \forall u \in \mathbf{U}$
Unit outlet splitter	$F_{c,u}^{O} = \sum_{u' \in \mathbf{U}} F_{c,u,u'} \quad , \quad \begin{vmatrix} \forall c \in \mathbf{C} \\ \forall u \in \mathbf{U} \end{vmatrix}$ $F_{c,u,u'} \cdot \sum_{c \in \mathbf{C}} F_{c,u}^{O} = F_{c,u}^{O} \cdot \sum_{c \in \mathbf{C}} F_{c,u,u'} \quad , \quad \begin{vmatrix} \forall c \in \mathbf{C} \\ \forall u' \in \mathbf{U} \end{vmatrix}$	$\sum_{u' \in \mathbf{U}} \zeta_{u,u'} = 1 \quad , \quad \forall u \in \mathbf{U}$

Depending on the kind of modeling, these variables can be used in different ways [34]. One approach uses of split fractions $\zeta_{u,u'}$ but no unit selection binaries y_u , leading in general to NLPs, where the best process design is defined by those units and interconnections where material flow is nonzero. Here, the use of split fractions leads to bilinear terms in the equations describing the stream mixers (see column2 Table 1). This approach has some disadvantages. First, since no binary variables are used, it is very difficult to formulate logic constraints imposing connectivity implications restricting the search space (i.e. when a particular unit is excluded, some of the units connected to it should also be excluded). Second, without binary variables it is not possible to use linear capital cost models with a binary activated independent terms; a technique applied to capture in part the concave nature of capital cots functions while avoiding the use of non-linear concave costs functions (see Figure 15) which lead optimization formulations with many local minima [34].

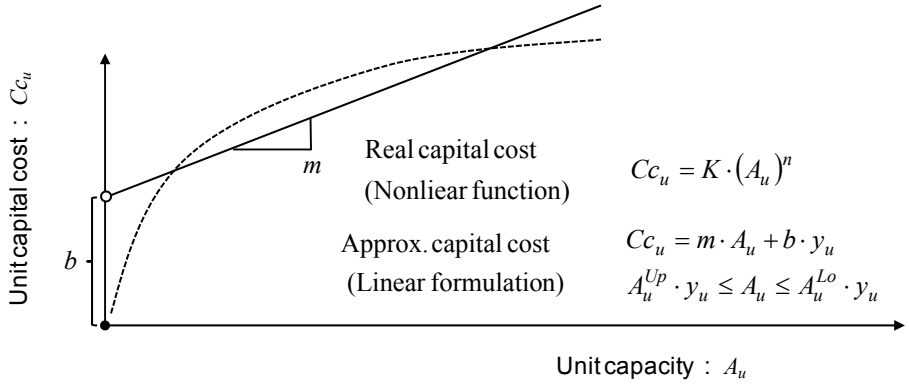


Figure 15: Unit capital cost function: Nonlinear concave formulation (doted curve), Binary activated linear formulation (solid line).

Another approach uses stream component flows $F_{c,u,u'}$, and unit selection binaries y_u leading to MINLPs, where the best design is defined by selected units (i.e. those with $y_u = 1$) as well as connections with nonzero flow. Mixers and splitters are described through linear material balances,

but enforcing composition equality among outlet splitter streams requires equations with bilinear terms in the molar flows (see column 1 Table 1). The use of unit binary variables facilitate the formulation of logic constraints and linear capital cost functions, but the use of split flows $F_{c,u,u'}$ could in general lead to a bigger number of flow variables.

Alternatively, stream split fractions $\zeta_{u,u'}$, and unit selection binaries y_u can be used, leading to MINLPs with less variables than in the previous case (see column 2 Table 1), while facilitating the incorporation of connectivity implications between process units, splitters and mixers. Here, stream split fractions $\zeta_{u,u'}$ are set to zero when either u or u' are excluded from the solution. The best design is defined again by the selected units and connections with non-zero flow.

It is important to notice that the use of split fractions and unit selection binaries brings a number of difficulties when solving the superstructure model. First, the bilinear terms in splitter equations, as well as other sources of nonlinearities, result in non-convex models subject to the existence of many local minima. Problems coming from bilinear terms have been traditionally solved by using a finite list of split values which are used to transform the bilinear terms into linear terms with binary variables (see section 4.4.1). Other alternatives include the use of convex envelopes (i.e. McCormick), special modeling approaches for splitters [18] and even the incorporation of these elements in dedicated global optimization algorithms [35]. Second, as discussed in section 2.1, a powerful solution algorithm for MINLPs such as OA, is based on the solution of a sequence of NLPs and MIPs, where the NLPs result from fixing the values of y_u in the original model to those found by solving the MIPs , while the MIPs result from linear approximations of the original model around the optimal solutions of the mentioned NLPs. Fixing some y_u to zero (i.e. excluding unit u from the solution), drives many continuous variable to zero when solving the NLPs , which in turn could lead

to numerical singularities and other problems in the formulation of the linear approximations required to build the MIPs. To solve this difficulty, special decomposition methodologies [18] or logic based modeling and solution algorithms can be used [36], [37]. Alternatively, if the superstructure model is posed as a Generalized Disjunctive Program (GDP), special logic based algorithms [38],[39] and computer codes [40] can also be used. For the two mayor superstructure representations (STNs and SENs), a comprehensive and systematic modeling methodology has been developed [19] where models can be posed as either MINLPs or GDPs.

At a lower level, the modeling of individual unit operations has been carried out in many ways, generally using some kind of simplification in order to guarantee the computational tractability of the final superstructure model. These simplified modeling includes the use of simple linear and bilinear approximations (e.g. separators with simple component split fractions [41]), shortcut models (e.g. Underwood models for distillation columns [42]) and the use of first principle models with simplified kinetics and thermodynamic property calculations (e.g. tray by tray distillation models with ideal thermodynamics [43]) . Such simplifications can be valid in instances where the chemical systems are ideal enough, and/or where the modeling is intended as a preliminary effort to investigate high level characteristics of a processing system. However, in instances where realistic and accurate results are required, the use of realistic thermodynamics and detailed unit models is mandatory. In this way, the challenge is to avoid excessive simplification (i.e. keeping a realistic model) without compromising the numerical tractability of the whole formulation.

2.5 Use of surrogate models in optimization

Surrogate models or meta-models can be used to reduce model complexity, during the formulation of design problems [44], [45], [46]. These surrogates are generally useful in situations involving iterative

solution procedures where a very complex model has to be evaluated at every iteration, hence becoming very computationally intensive. For example, consider the design of an optimal hydrodynamic shape, wing or a boat propeller, where the system under study is described by a detailed Computational Fluid Dynamic (CFD) model. A surrogate model capturing only the relationship of variables that are important can be used instead of the original detailed CFD model. This replacement potentially reduces the mathematical complexity, not only by reducing the total number of variables (since only important variables are included), but also by allowing the selection of surrogates with a convenient mathematical form. In some cases, only complicated sub-blocks of equations need to be replaced, resulting in hybrid models.

In the area of process systems engineering, surrogates have been used to support the modeling of chemical process systems [47],[48,][49],[50], as well the solution of special classes of MINLPs used in the synthesis and optimization of chemical process [51],[52],[53],[54]. These surrogates are built by fitting process data using general purpose multivariable mappings (e.g. Kriging interpolators, Artificial Neural Networks ANN, Support Vector Machines SVM, Radial Basis Functions RBF, etc.) The process data may come from different sources: Experiments, special purpose computer subroutines, commercial software, etc. In particular, commercial process simulators are very useful since they can produce realistic data for commonly used processing units through the integration of detailed unit operation models and real thermodynamic calculation subroutines.

Chapter 3

PROPOSED SUPERSTRUCTURE GENERATION FRAMEWORK

This chapter presents our approach to superstructure representation and generation, including the basic elements used, and the way to produce rich but significant connectivity.

3.1 Superstructure representation elements

Our approach uses three kinds of elements to create a superstructure representation: processing units, connection ports and general conditioning streams. Here, 'processing units' are mainly regarded as units where materials are transformed via reaction or separation. In addition, raw material source units and final product sink units are included as auxiliary elements whose function is to provide raw materials to the process and collect final products from the process. The 'connection ports' correspond to material inlet and outlet points in every processing unit (e.g. reactor inlet, absorption column solvent inlet, distillation column distillate outlet, etc). Every unit type has a predefined number of inlet and outlet ports (e.g. flash vessels have a single inlet port and two outlet ports), including the source and sink units (i.e. source units have one outlet raw material

port, while sink units have one inlet final product port). Ports allow an arbitrary connectivity between units. That is, every outlet port is regarded as a general splitting devise able to divide the material leaving a unit into fractions sent to different processing units. Every inlet port is regarded as a general mixing devise able to combine materials coming from different units into a single material flow entering a unit. Finally, 'general conditioning streams' act as connections between outlet ports and inlet ports, while performing a two-step conditioning operation: a general compression-expansion operation followed by a general heating-cooling operation. This conditioning is responsible for the temperature and pressure changes required to generate inlet port mixtures at conditions necessary for the receiving unit to operate properly.

The use of these superstructure elements offers the possibility of highly connected superstructures (as in the case of SENs) while emphasizing a distinction between the reaction/separation operations and conditioning operations. This distinction is important since the former are responsible for the main material transformations in the process, while the latter support the operation of the former. Under this perspective, superstructure modeling focuses on the main units, and provides a simple but flexible treatment of conditioning operations. In fact, the use of general conditioning streams gives the superstructure conditioning capabilities beyond what is possible with other representations and facilitates the use of aggregated models to deal with the integration of industrial utilities (e.g. Pinch Analysis).

In order to allow a more traditional treatment of conditioning operations, our methodology also considers the use of 'non-conditioning streams' and some simple 'conditioning units' (i.e. expansion valves, compressor-expanders, heater-coolers). The 'non-conditioning streams' establish simple connections between ports, and can be used to combine conditioning units into complex conditioning systems. In this way, instead of using general conditioning streams for all processing

unit connections (possibly leading to an overly flexible conditioning scheme), a portion of the connections between processing units can be replaced with non-conditioning streams, or a combination of non-conditioning streams and one or more conditioning units. The selection of which connections to treat in one way or another can be made based on the processing functions of the units to connect (i.e. if the outlet conditions of a unit are adequate for the operation of a downstream unit, a connection between them can be made using a non-conditioning stream)

3.2 Superstructure generation

As discussed in section 2.3, superstructure generation methodologies aim at the formulation of the simplest superstructure that is rich enough to guarantee the optimal solution is not overlooked. Our proposed methodology starts with the selection of a set of processing units (reaction and separation only) required to generate a series of simple structures able to produce the required products using the available raw materials. Since every unit performs a specific function in one of these structures, each unit in the superstructure is assigned a processing function. In this respect, our representation is similar to STNs, but the processing functions here are more general and not bound to a rigid description through detailed states. For example, a unit required to perform the separation of components A,B is able to admit other components in its feed, for a wide range of compositions and flows.

To complete the representation, the inlet and outlet ports of each unit are identified and then, fully interconnected (i.e. every outlet port is connected to all inlet ports). Later, a set of connectivity rules presented in the next section are used to eliminate those connections which are counterproductive. The final result is a rich superstructure with extended connectivity that contains relevant and feasible connections.

It is important to highlight that this methodology incorporates a lot of engineering judgment, not only during the selection of processing units (as a set of potentially feasible structures have to be initially proposed), but also during the identification of the final connectivity, which is based on unit processing functions, and the set of different materials which should and should not reach each unit for its function to be possible.

3.3 Connectivity simplification rules

Connections allow the transfer of materials in a process structure, making them available to each processing unit as required, according with their intended processing function (i.e. normal operation). These requirements can be expressed in terms of the materials which must be present and absent at each port for a unit to operate properly. In order to incorporate these ideas, the proposed fully connected superstructures have to be simplified. In particular, after reducing the connectivity, every unit should be reachable by the components required for its normal operation and avoidable by components whose presence can disrupt such operation. In order to present a formal statement of these connectivity rules, four basic component subsets are defined:

- *Feasible component set of inlet port* p *(*$\mathbf{C}_p^{\mathrm{FI}}$*):* Set of components whose presence in port p will not affect negatively the normal operation of its unit. For example, a component C is not feasible for the inlet port to a separator of components A,B if its presence creates an A-B azeotrope.
- *Minimal component set of inlet port* p *(*$\mathbf{C}_p^{\mathrm{MI}}$*):* Set of components whose presence in port p is required during the normal operation of its unit. For example, components A and B are minimal for the inlet port of a reactor producing component C via the reaction A+B $\rightarrow$C.

- *Feasible component set of outlet port* p' *(*$\mathbf{C}_{p'}^{\mathrm{FO}}$*):* Components present in the outlet port p' of a unit performing its intended processing function being fed with feasible components.
- *Minimal component set of outlet port* p' *(*$\mathbf{C}_{p'}^{\mathrm{MO}}$*):* Components present in the outlet port p' of a unit performing its intended processing function while fed with minimal components.

For a given connectivity, the normal operation of a particular unit u is feasible if it is possible to avoid the presence of harmful (i.e. infeasible) components in its inlet ports, and it is possible to supply all its inlet ports with all their minimal components. A process superstructure network is feasible if the normal operation of all its units is feasible.

Connectivity rule 1 (Avoiding feeds with infeasible components): The presence of infeasible components in inlet port p is avoidable if, for every incoming connection from an outlet port p', all minimal components in p' (always present during the normal operation of its unit) are feasible for p. Mathematically, such rule is expressed as:

$$\mathbf{C}_{p'}^{\mathbf{MO}} \subseteq \mathbf{C}_{p}^{\mathbf{FI}} \qquad \forall \,\mathrm{connected}(p', p) \;\equiv\; \bigcup_{p' \,\mathrm{connected\,to}\, p} \mathbf{C}_{p'}^{\mathbf{MO}} \subseteq \mathbf{C}_{p}^{\mathbf{FI}} \qquad \forall p \tag{3-1}$$

Connectivity rule 2 (Promoting feeds with all minimal components): The supply of all minimal components to a unit inlet port (or final product) p is possible if the incoming connections from all outlet ports p' might collectively contain all such components. Mathematically, such rule is expressed as:

$$\mathbf{C}_{p}^{\mathbf{MI}} \subseteq \bigcup_{p' \mathrm{connected\,to}\, p} \mathbf{C}_{p'}^{\mathbf{FO}} \qquad \forall p \tag{3-2}$$

Connectivity rules 3, 4 (Promoting useful inlet connections): In principle, a useful inlet connection to a reactor inlet port p is one which might supply at least *one* of its minimal components (i.e. reactants). A useful inlet connection to a separator inlet port p is one which might supply *all* its minimal components (i.e. key components). The condition for separators is intended to avoid unnecessary mixing & separation. For example, the connection between the source port for raw material A and a unit intended to separate A from product C (both minimal components) does not violate (3-1)-(3-2) but it is obviously counterproductive. Formally, these rules are expressed as:

$$\mathbf{C}_{p}^{\mathbf{MI}} \cap \mathbf{C}_{p'}^{\mathbf{FO}} \neq \phi \qquad \forall \,\text{connected}(p', p), \forall \,\text{Reactor inlet port } p \tag{3-3}$$

$$\mathbf{C}_{p}^{\mathbf{MI}} \cap \mathbf{C}_{p'}^{\mathbf{FO}} = \mathbf{C}_{p}^{\mathbf{MI}} \qquad \forall \,\text{connected}(p', p), \forall \,\text{Separator inlet port } p \tag{3-4}$$

While (3-1)-(3-2) are necessary for the feasible operation of the units, they are not sufficient to guarantee such feasibility. For example, (3-2) promotes the presence of all minimal components in all inlet ports, but it does not guarantee it. In addition, notice that (3-3)-(3-4) are optional and can be disregarded to allow connections which might be useful without supplying minimal components to a particular unit. For example, for a unit intended to separate key components A and B (minimal components) a feed connection supplying component C (not minimal) is useful if the presence of C facilitates the separation by reducing the utility consumption.

In general, the application of these rules can lead to a number of very different superstructures (some simpler than others) depending entirely on the way the processing functions and the port component sets are defined for each unit. This flexibility is what makes possible the incorporation of engineering judgment at this stage. It is also important to clarify that specific knowledge regarding how the components behave (e.g. thermodynamic behavior) and how the units operate (e.g. what

processing functions they can actually accomplish) is fundamental in the creation of a superstructure which covers realistic process structures. In other words, incorrect or little process information can only lead to counterproductive or overly complex connectivity, since simplification claims become either wrong or difficult to support. At the end, the connectivity simplification can be seen as a preprocessing step to the optimization calculations, where obvious disadvantageous connections are eliminated. The quality of these simplifications depends exclusively on the quality of the information available.

Compared to STN representations, our approach leads to simpler diagrams not only because the unit models are more flexible (hence they perform a number of tasks as required), but also because the conditioning operations have been assigned to the streams. Compared to SEN representations, our approach leads to simpler connectivity for a given initial set of units. However, for a given PSP, the number of units used in our representation tend to be higher than in the case of SENs. As we will discuss later, having an idea of the function each unit is supposed to perform, allows the formulation of processing unit models which are less complex (also less general) than the fully detailed models used with SEN representations. In conclusion, the proposed approach leads to representations and models which are in between SENs and STNs.

3.4 Superstructure generation: Illustrative example

Consider the synthesis problem for the Hydrodealkylation (HDA) process [55]. Here, the production of Benzene from Toluene and Hydrogen is based on the following stoichiometry:

$$C_7H_8 + H_2 \rightarrow C_6H_6 + CH_4 \quad (rxn_1)$$
$$2 \cdot C_6H_6 \leftrightarrow C_{12}H_{10} + H_2 \quad (rxn_2)$$

The main reaction (rxn_1) generates the product C_6H_6 (Benzene) and a byproduct CH_4 (Methane). The side reversible reaction (rxn_2) consumes the main product and generates a secondary product $C_{12}H_{10}$ (Diphenyl). We are interested in the production of C_6H_6 and a marginal production of $C_{12}H_{10}$, if the economics supports it.

The available process information suggests that production of Diphenyl can be inhibited by the presence of both Hydrogen and Diphenyl in the reactors. Hence, proper reactor operation involves Hydrogen excess and Diphenyl recycling. To support the reactors, other operations have to cover the separation and recycling of $H_2, C_7H_8, C_{12}H_{10}$, the separation and elimination of CH_4, and the separation and purification of C_6H_6. The component relative volatilities allow all separations to be performed using Liquid-Vapor operations exclusively. Also, with the volatility order given by $\alpha_{H_2} > \alpha_{CH_4} > \alpha_{C_6H_6} > \alpha_{C_7H_8} > \alpha_{C_{12}H_{10}}$, the specific separation cuts to perform are CH_4 / C_6H_6, C_6H_6 / C_7H_8, $C_7H_8 / C_{12}H_{10}$. To cover these processing functions the superstructure will include one plug flow reactor (PR1), one stirred tank reactor (SR1), three flash vessels (FV1-FV3) one absorption column (AC1) and one distillation column (DC-1). The process also includes two raw material sources (SC1,SC2) and three product sinks (SK1-SK3). The specific processing functions and port component sets are presented in Table 2.

Tables evaluating each of the possible connections between unit ports are presented in appendix A. To avoid counterproductive mixing, sink ports are treated as separator inlet ports, that is, no connections between source and sink units are considered here since product streams are expected to contain mainly other components.

Table 2: Processing functions and port component sets for units in the HDA process.

Unit	Processing function [1]	Port[2]	CM [1]	CF[1]
SC1	Supply raw material 1	O1	1	1
SC2	Supply raw material 4	O1	4	4
SK1	Retrieve waste material 2	I1	2	1,2
SK2	Retrieve main product 3	I1	3	3
SK3	Retrieve secondary product 5	I1	5	5
PR1, SR1	Consume 1 and 4, to produce 3 while minimizing the production of 5	I1	1,4,5	1,2,4,5
		O1	1,2,3,4,5	1,2,3,4,5
FV1	Separate lights 1,2 from heavies 3,4,5 from a mixture containing significant amounts of all of them.	I1	2,3	1,2,3,4,5
		O1	2 [3]	1,2 [3]
		O2	2,3	2,3,4,5
FV2	Strip 2 from a mixture containing traces of 2 and significant amounts 3,4,5.	I1	2,3	2,3,4,5
		O1	2,3	2,3
		O2	3 [4]	3,4,5 [4]
FV3	Strip 4 from a mixture containing 4 and 5.	I1	4,5	4,5
		O1	4,5	4,5
		O2	5 [4]	5 [4]
AC1	Separate incondensable 1 highly volatile 2 using a selective absorbing 2 in heavy component 4	I1	1,2	1,2,3,4,5
		I2	4	3,4,5
		O1	1 [5]	1 [5]
		O2	2,4	2,3,4,5
DC1	Separate component 3 from heavier components 4 and 5	I1	3,4	3,4,5
		O1	3 [6]	3
		O2	4 [6]	4,5 [6]

Notes:

(1) Component identification number: H_2=1, CH_4=2, C_6H_6=3, C_7H_8=4, $C_{12}H_{10}$=5

(2) Port naming: Inlet 1(or light inlet) = I1, Inlet 2 (heavy outlet) = I2
Outlet 1 (or light out) = O1, Outlet 2 (heavy outlet) = O2

(3) Assuming complete rectification (i.e. no heavies in the light product)

(4) Assuming complete stripping (i.e. no lights in the heavy product)

(5) Assuming complete stripping (i.e. no lights in the heavy product)

(6) Assuming sharp separation (i.e. no heavies in the light product, no lights in the heavy product)

The unit set used for this superstructure includes 14 outlet ports and 11 inlet ports. These ports define 154 streams in the fully connected superstructure. However, the number of streams is reduced to 36 when the connectivity rules are enforced. The final superstructure is presented in Figure 16.

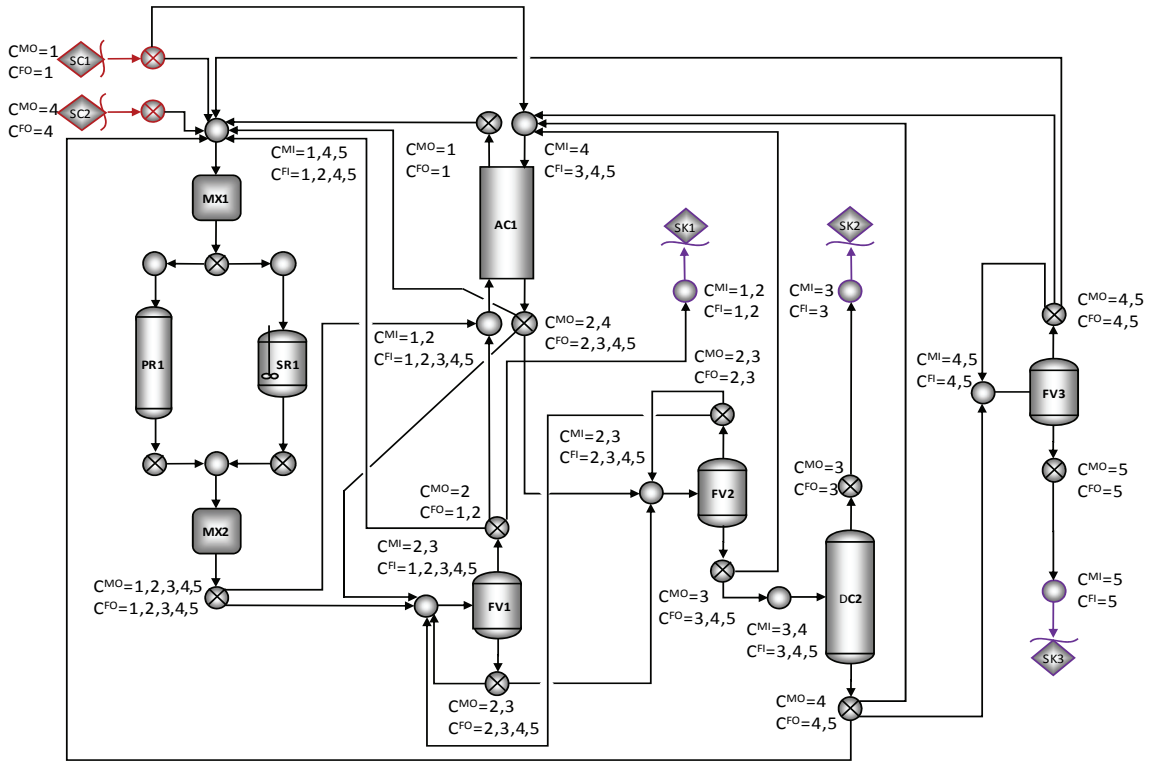


Figure 16: HDA process superstructure generated with connectivity rules.

In order to have a point of comparison, Figure 17 shows an adaptation of a typical superstructure for the HDA process. This figure uses the connectivity suggested in the literature [18] for the limited set of units used in this example. It is evident that our approach brings a richer connectivity, hence a better chance of including the real optimal structure for the given set of units.

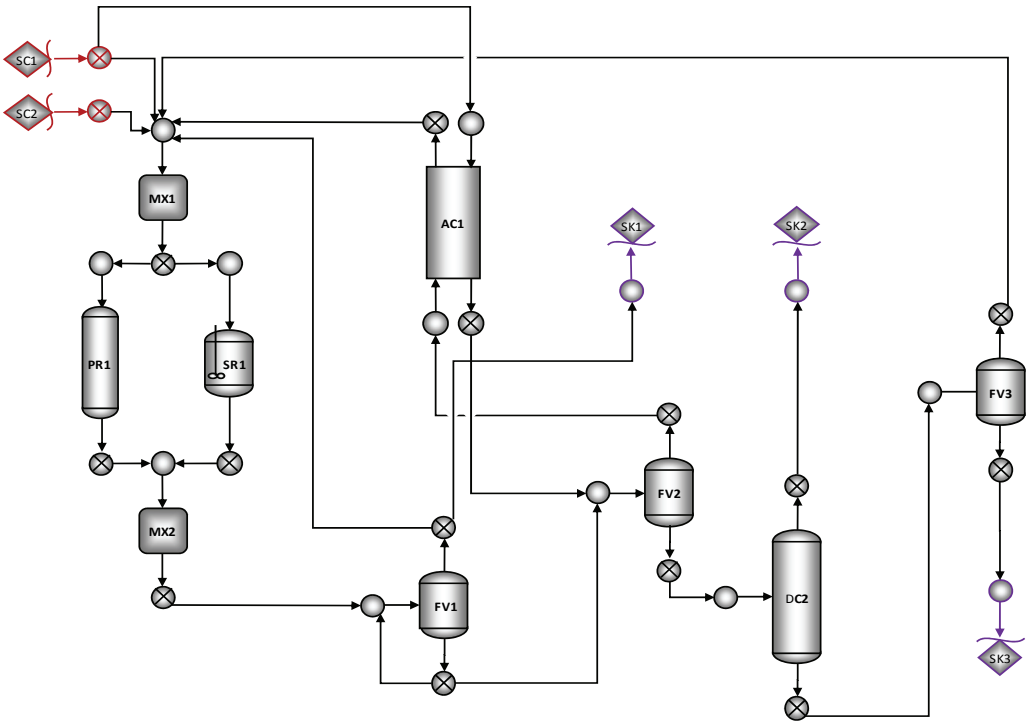


Figure 17: Traditional HDA process superstructure [18] .

Chapter 4

PROPOSED SUPERSTRUCTURE MODELLING APPROACH

This chapter presents the detailed superstructure model formulation, including the way basic sets are defined and used to index variables and equations, the role surrogate models play, the high level structure of the final SO model, as well as the specific models used for each superstructure element.

4.1 Indices and sets

Six basic sets are used to index the model variables and constraints:

- *Unit types* $(ut \in \mathbf{UT})$: Includes the following labels: *sc* = source or raw material unit, *sk* = sink or final product unit, *mx* = Mixer-Splitter, *hc* = Heater-Cooler, *ce* = Compressor-Expander, *ev* = Expansion valve, *ms* =membrane separator, *fv* = flash vessel, *ac* = absorption column, *ec* = extraction column, *dc* = distillation column, *sr* = stirred tank reactor, *pr* = plug flow reactor.
- *Unit consecutive numbers* $(uc \in \mathbf{UC})$*:* List of consecutive integers used to distinguish units of the same type.

- *Port types* $(pt \in \mathbf{PT})$: Includes the following labels: I = inlet, O=outlet.
- *Port consecutive numbers* $(pc \in \mathbf{PC})$: List of consecutive integers used to distinguish ports of the same type in the same unit.
- *Compounds* $(c \in \mathbf{C})$: Includes labels to denote the different chemicals used in the process.
- *Reactions* $(r \in \mathbf{R})$: List of consecutive integers used to denote different reactions.

From these fundamental sets, several multidimensional sets can be declared. The unit set is declared as $u \in \mathbf{U}(ut,uc)$; that is, units are indexed by unit type and unit consecutive number. The port set is declared as $p \in \mathbf{P}(ut,uc,pt,pc)$; that is, ports are indexed by the unit they belong to, the port type and the port consecutive number. Finally, the stream set is declared as $s \in \mathbf{S}(ut,uc,pt,pc,ut',uc',pt',pc')$; that is, streams are indexed by the origin and destination ports they connect.

From supersets $\mathbf{U},\mathbf{P},\mathbf{S}$ several subsets are defined to facilitate the modeling. Unit subsets include: Raw material sources $\mathbf{U}^{\mathbf{SC}}$, final product sinks $\mathbf{U}^{\mathbf{SK}}$, mixer-splitters $\mathbf{U}^{\mathbf{MX}}$, heater-coolers $\mathbf{U}^{\mathbf{HE}}$, compressor-expanders $\mathbf{U}^{\mathbf{CE}}$, expansion valves $\mathbf{U}^{\mathbf{EV}}$, membrane separators $\mathbf{U}^{\mathbf{MS}}$, flash vessels $\mathbf{U}^{\mathbf{FV}}$, absorption columns $\mathbf{U}^{\mathbf{AC}}$, LL extraction columns $\mathbf{U}^{\mathbf{EC}}$, distillation columns $\mathbf{U}^{\mathbf{DC}}$, stirred tank reactors $\mathbf{U}^{\mathbf{SR}}$ and plug flow reactors $\mathbf{U}^{\mathbf{PR}}$. Port subsets include: Raw material ports $\mathbf{P}^{\mathbf{RM}}$, final product ports $\mathbf{P}^{\mathbf{FP}}$, inlet ports $\mathbf{P}^{\mathbf{I}}$, outlet ports $\mathbf{P}^{\mathbf{O}}$, inlet ports of unit u $\mathbf{P}_u^{\mathbf{I}}$, outlet ports of unit u $\mathbf{P}_u^{\mathbf{O}}$, and unit ports $\mathbf{P}_u = \mathbf{P}_u^{\mathbf{I}} \cup \mathbf{P}_u^{\mathbf{O}}$. Stream subsets include: Conditioning streams $\mathbf{S}^{\mathbf{C}}$, non-conditioning streams $\mathbf{S}^{\mathbf{NC}}$ streams connected to inlet port p $\mathbf{S}_p^{\mathbf{I}}$, streams connected to outlet port p' $\mathbf{S}_{p'}^{\mathbf{O}}$, inlet streams to a unit u $\mathbf{S}_u^{\mathbf{I}}$ and outlet streams of a unit u $\mathbf{S}_u^{\mathbf{I}}$. For convenience in the

formulation of some special constraints, the set $\mathbf{U}_s$ of units connected by stream s, and the set $\mathbf{U}_p$ of units a particular port p belongs to, are also declared.

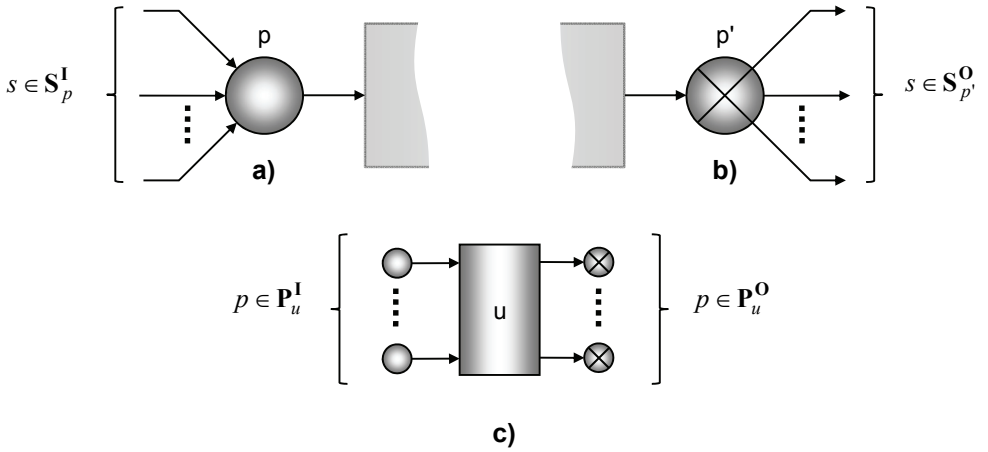


Figure 18: Graphic representation of port and stream sets. **a)** Inlet port streams, **b)** Outlet port streams. **c)** In/out unit ports.

4.2 Modeling elements

Modular modeling is an important characteristic of our approach, allowing fast superstructure model creation, model enhancement and code recycling. To satisfy this requirement, standard models have been developed for all the basic superstructure elements, including the most common processing units. In addition, to reduce the mathematical complexity of each element model, entire blocks of nonlinear equations have being replaced with multivariable mapping functions generated by fitting realistic process data.

In this hybrid surrogate modeling approach, most of the linear equations in the original detailed superstructure element models are maintained (e.g. material an energy balances), and the rest are replaced by non-linear surrogate mapping functions. The models are presented here in a general

form, highlighting the fact that different types of surrogate mappings can be used (e.g. radial basis functions, support vector regressions, Kriging interpolators, etc.)

4.3 High level modeling

Posed as a Generalized Disjoint Program (GDP), a component flow based formulation of the complete SO model has the structure shown in (4-1)-(4-6) [56],[57],[36]:

$$\max \quad \sum_{p\in\mathbf{P^{FP}}} \pi_p \cdot \left(\sum_{c\in\mathbf{C}} F_{c,p}\right) - \sum_{p'\in\mathbf{P^{RM}}} \pi_{p'} \cdot \left(\sum_{c\in\mathbf{C}} F_{c,p'}\right) - \sum_{u\in\mathbf{U}} \left(\left(\frac{A}{P}\right)\cdot Cc_u + Co_u\right) \tag{4-1}$$

s.t:

$$\begin{bmatrix} \overset{Y_u}{} \\ h_u\left(\left[F_{c,p},T_p,P_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u}},\boldsymbol{\psi}_u\right)=0 \quad , \quad g_u\left(\left[F_{c,p},T_p,P_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u}},\boldsymbol{\psi}_u\right)\le 0 \\ Cc_u = \mathrm{f}_u^{Cc}\left(\left[T_p,P_p\right]_{p\in\mathbf{P}_u},\boldsymbol{\psi}_u\right) \quad , \quad Co_u = \mathrm{f}_u^{Co}\left(\boldsymbol{\psi}_u\right) \end{bmatrix} \vee \begin{bmatrix} \neg Y_u \\ F_{c,p}=0 \;\; \vert \forall c\in\mathbf{C} \\ T_p=0 \;\; \vert \forall p\in\mathbf{P}_u \\ P_p=0 \\ \boldsymbol{\psi}_u=0 \\ Cc_u=0 \quad , \quad Co_u=0 \end{bmatrix} \quad \forall u\in\mathbf{U} \tag{4-2}$$

$$\begin{bmatrix} \overset{Y_{p'}}{} \\ h_{p'}\left(\begin{matrix}\left[F_{c,p},T_{p'},P_{p'}\right]_{c\in\mathbf{C}},\boldsymbol{\psi}_{p'} \\ \left[F_{c,s},T_s^{S1},P_s^{S1}\right]_{\substack{c\in\mathbf{C}\\ s\in\mathbf{S}_{p'}}},\left[\boldsymbol{\psi}_s\right]_{s\in\mathbf{S}_{p'}}\end{matrix}\right)=0 \quad , \quad g_{p'}\left(\begin{matrix}\left[F_{c,p'},T_{p'},P_{p'}\right]_{c\in\mathbf{C}},\boldsymbol{\psi}_{p'} \\ \left[F_{c,s},T_s^{S1},P_s^{S1}\right]_{\substack{c\in\mathbf{C}\\ s\in\mathbf{S}_{p'}}},\left[\boldsymbol{\psi}_s\right]_{s\in\mathbf{S}_{p'}}\end{matrix}\right)\le 0 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{p'} \\ F_{c,p'}=0 \;\; \vert \forall c\in\mathbf{C} \\ T_{p'}=0 \\ P_{p'}=0 \\ \boldsymbol{\psi}_{p'}=0 \end{bmatrix} \quad \forall p'\in\mathbf{P^O} \tag{4-3}$$

$$\begin{bmatrix} \overset{Y_p}{} \\ h_p\left(\begin{matrix}\left[F_{c,p},T_p,P_p\right]_{c\in\mathbf{C}},\boldsymbol{\psi}_p \\ \left[F_{c,s},T_s^{S3},P_s^{S3}\right]_{\substack{c\in\mathbf{C}\\ s\in\mathbf{S}_p}},\left[\boldsymbol{\psi}_s\right]_{s\in\mathbf{S}_p}\end{matrix}\right)=0 \quad , \quad g_p\left(\begin{matrix}\left[F_{c,p},T_p,P_p\right]_{c\in\mathbf{C}},\boldsymbol{\psi}_p \\ \left[F_{c,s},T_s^{S3},P_s^{S3}\right]_{\substack{c\in\mathbf{C}\\ s\in\mathbf{S}_p}},\left[\boldsymbol{\psi}_s\right]_{s\in\mathbf{S}_p}\end{matrix}\right)\le 0 \end{bmatrix} \vee \begin{bmatrix} \neg Y_p \\ F_{c,p}=0 \;\; \vert \forall c\in\mathbf{C} \\ T_p=0 \\ P_p=0 \\ \boldsymbol{\psi}_p=0 \end{bmatrix} \quad \forall p\in\mathbf{P^I} \tag{4-4}$$

$$\begin{bmatrix} \overset{Y_s}{} \\ h_s\left(\left[F_{c,s},T_s^{st},P_s^{st}\right]_{\substack{c\in\mathbf{C}\\ st\in\{S1,..,S3\}}},\boldsymbol{\psi}_s\right)=0 \quad , \quad g_s\left(\left[F_{c,s},T_s^{st},P_s^{st}\right]_{\substack{c\in\mathbf{C}\\ st\in\{S1,..,S3\}}},\boldsymbol{\psi}_s\right)\le 0 \\ Cc_s = \mathrm{f}_s^{Cc}\left(\left[T_s^{st},P_s^{st}\right]_{st\in\{S1,..,S3\}},\boldsymbol{\psi}_s\right) \quad , \quad Co_s = \mathrm{f}_s^{Co}\left(\boldsymbol{\psi}_s\right) \end{bmatrix} \vee \begin{bmatrix} \neg Y_s \\ F_{c,s}=0 \;\; \vert \forall c\in\mathbf{C} \\ T_s^{st}=0 \;\; \vert st\in\{S1,..,S3\} \\ P_s^{st}=0 \\ \boldsymbol{\psi}_s=0 \\ Cc_s=0 \quad , \quad Co_s=0 \end{bmatrix} \quad \forall s\in\mathbf{S} \tag{4-5}$$

$$\Omega\left(\left[Y_u, Y_p, Y_s\right]_{\substack{u\in\mathbf{U}\\ p\in\mathbf{P}\\ s\in\mathbf{S}}}\right) = \text{True}, Y_i \in \{True, False\} \tag{4-6}$$

The objective function (4-1) is an approximate operative profit = (product revenue - raw material cost - capital cost rate - operative cost). Here, π_p is the price/cost of the material in port p. $F_{c,p}$ and $F_{c,s}$ denote the molar flow of component c in port p and stream s, respectively. Cc_u and Co_u are the capital and operative costs of unit u, and (A/P) is a proper inverse annuity factor. The disjunctive constraints (4-2) to (4-5) enforce either the element models (i.e. unit models $h_u(...)=0, g_u(...)\le 0$, port models $h_p(...)=0, g_p(...)\le 0$, stream models $h_s(...)=0, g_s(...)\le 0$), or set all their variables to zero. The element models are presented in terms of state variables (component flows F_c, temperatures T and pressures P), as well as internal unit, port and stream variables, grouped in the vectors $\boldsymbol{\psi}_u$, $\boldsymbol{\psi}_p$ and $\boldsymbol{\psi}_s$, respectively. Some arrays are used to group variables. For example $\left[F_{c,p}, T_p, P_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u}}$ groups all component molar flows, temperatures and pressures for all ports in unit u. When a unit, port or stream is activated (i.e. the corresponding Boolean Y_u, Y_p, Y_s is true), the element model is enforced (contained in the left disjuncts); otherwise, all its variables are set to zero (as indicated by the right disjuncts).

The super index st denotes different thermodynamic states required to describe stream conditioning (see section 4.4.2). As discussed earlier, conditioning streams are regarded in our methodology as general conditioning operations, hence they have associated capital costs Cc_s and operative costs Co_s. The final set of constraints (4-6) includes connectivity implications and other logic constraints [58],[59], where selection Booleans Y_u, Y_p, Y_s are interrelated.

The general GDP formulation in (4-1)-(4-6) includes a couple of key modeling ideas supporting the modular approach. First, unit models are presented exclusively in terms of internal variables and port variables. This makes it very easy to review, modify and implement new unit models. The only condition for such new models to be compatible with our formulation is that they use the standard set of port variables shown previously. Second, port models serve as interfaces between unit models and stream models. This characteristic, along with the fact that streams are indexed by port pairs, simplifies the definition and modification of the superstructure connectivity. In fact, as discussed in chapter 7 the superstructure connectivity is defined by a single model parameter called the connectivity matrix. Updating the model when a change is made in the superstructure connectivity is reduced to updating this single parameter.

4.4 Low level modeling

This section presents the F_c-based models for ports, streams and a general processing unit, as well as the logic connectivity constraints. The corresponding $F_c - x_c$ based models are included in appendix B. Specific models for unit operations are presented in appendix D.

4.4.1 Unit ports

Figure 19 includes the graphic representations for inlet and outlet unit ports.

Inlet ports $\left(p \in \mathbf{P}^{\mathbf{I}}\right)$*:* These ports are modeled as adiabatic stream mixers enforcing pressure balance between all incoming streams:

$$F_{c,p} = \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} F_{c,s} \quad , \quad \forall c \in \mathbf{C} \tag{4-7}$$

$$H_p = \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} H_s^{S3} \tag{4-8}$$

$$P_p = P_s^{S3} \quad , \quad \forall s \in \mathbf{S}_p^{\mathbf{I}} \tag{4-9}$$

$$H_p = \mathrm{f}^H\left(\left[F_{c,p}\right]_{c \in \mathbf{C}}, T_p, P_p\right) \quad , \quad S_p = \mathrm{f}^S\left(\left[F_{c,p}\right]_{c \in \mathbf{C}}, T_p, P_p\right) \tag{4-10}$$

Equation (4-7) and (4-8) are material and energy balances, where F_c and H denote component flow and enthalpy flow. Equation (4-9) is the pressure balance condition between inlet streams. Finally, equations (4-10) are relationships between port state variables and its thermodynamic properties: enthalpy flow H and entropy flow S. Here, the nonlinear mapping functions $\mathrm{f}^H(\cdot)$ and $\mathrm{f}^S(\cdot)$ are built from process data to serve as a surrogate thermodynamic model.

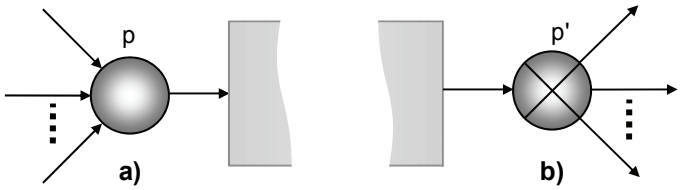


Figure 19: Unit ports: **a)** Inlet port p. **b)** Outlet port p'.

Unit outlet ports $\left(p' \in \mathbf{P}^{\mathbf{O}}\right)$*:* These ports are modeled as general adiabatic and isobaric splitters:

$$\sum_{s \in \mathbf{S}_{p'}^{\mathbf{O}}} \xi_s = 1 \tag{4-11}$$

$$F_{c,s} = \xi_s \cdot F_{c,p'} \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-12}$$

$$H_s^{S1} = \xi_s \cdot H_{p'} \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-13}$$

$$S_s^{S1} = \xi_s \cdot S_{p'} \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-14}$$

$$T_s^{S1} = T_{p'} \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-15}$$

$$P_s^{S1} = P_{p'} \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-16}$$

$$H_{p'} = \mathrm{f}^H\left(\left[F_{c,p'}\right]_{\forall c \in \mathbf{C}}, T_{p'}, P_{p'}\right) \quad , \quad S_{p'} = \mathrm{f}^S\left(\left[F_{c,p'}\right]_{\forall c \in \mathbf{C}}, T_{p'}, P_{p'}\right) \tag{4-17}$$

Equation (4-11) is the basic relationship between stream split fractions ξ_s for those streams $s \in \mathbf{S}_{p'}^{\mathrm{O}}$ leaving outlet port p'. Equations (4-12), (4-13) and (4-14) are the basic operating equations of an adiabatic splitter. Equation (4-15) is a consequence of the adiabatic-isobaric operation, and (4-16) is the isobaric operating condition. Finally, equations (4-17) are relationships between port state variables and its thermodynamic properties.

The model (4-11)-(4-17) is mostly linear, except for the non-linear mappings and the bilinear terms in (4-12), (4-13) and (4-14). These bilinear terms are important sources of non-convexities which can be eliminated by using a discrete set of fraction values ξ^k and a set of binary variables y_s^k instead of variable stream split fractions ξ_s. In this way, the bilinearities are replaced with a mixed integer linear formulation. Using a convex hull reformulation [38], equations (4-11)-(4-14) can be replaced by:

$$\sum_{k \in \mathbf{K}} y_s^k = 1 \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-18}$$

$$\sum_{s \in \mathbf{S}_{p'}^{\mathbf{O}}} \sum_{k \in \mathbf{K}} y_s^k \cdot \xi^k = 1 \tag{4-19}$$

$$F_{c,p'} = \sum_{k \in \mathbf{K}} F_{c,p',s}^k \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-20}$$

$$F^{Lo} \cdot y_s^k \le F_{c,p',s}^k \le F^{Up} \cdot y_s^k \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall k \in \mathbf{K} \tag{4-21}$$

$$F_{c,s} = \sum_{k \in \mathbf{K}} \xi^k \cdot F_{c,p',s}^k \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-22}$$

$$H_{p'} = \sum_{k \in \mathbf{K}} H_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-23}$$

$$H^{Lo} \cdot y_s^k \le H_{p',s}^k \le H^{Up} \cdot y_s^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall k \in \mathbf{K} \tag{4-24}$$

$$H_s^{S1} = \sum_{k \in \mathbf{K}} \xi^k \cdot H_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-25}$$

$$S_{p'} = \sum_{k \in \mathbf{K}} S_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-26}$$

$$S^{Lo} \cdot y_s^k \le S_{p',s}^k \le S^{Up} \cdot y_s^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall k \in \mathbf{K} \tag{4-27}$$

$$S_s^{S1} = \sum_{k \in \mathbf{K}} \xi^k \cdot S_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{4-28}$$

Equation (4-18) enforces the selection of only one fraction value for each outlet stream. Equation (4-19) enforces the basic relationship enforced by equation (4-11) (i.e. slip fractions must add up to one), this time for the selected stream split fractions. Equations (4-20), (4-23) and (4-26) are disaggregation of port component flows $F_{c,p'}$, port enthalpy flows $H_{p'}$, and port entropy flows $S_{p'}$, into $F_{c,p',s}^k$, $H_{p',s}^k$ and $S_{p',s}^k$. Equations (4-21), (4-24) and (4-27) activate and deactivate disaggregated port component flows, port enthalpy flows and port entropy flows, where F^{Lo}, F^{Up} are global lower and upper bounds on all component flows, H^{Lo}, H^{Up} are lower and upper bounds on enthalpy flows, and S^{Lo}, S^{Up} are lower and upper bounds on entropy flows. Finally, equations

(4-22), (4-25) and (4-28) are the disaggregated versions of the splitter operating equations (4-12), (4-13) and (4-14).

4.4.2 Streams

Figure 20 includes the graphic representations of streams. Conditioning streams include a general heating-cooling unit and a general compression-expansion unit, creating three distinct states: Initial state (S1) defined by the conditions of the origin port; intermediate state (S2) result of the general heating-cooling operation; and final state (S3) result of the general compression-expansion operation. Non conditioning streams are simple connections between outlet and inlet ports.

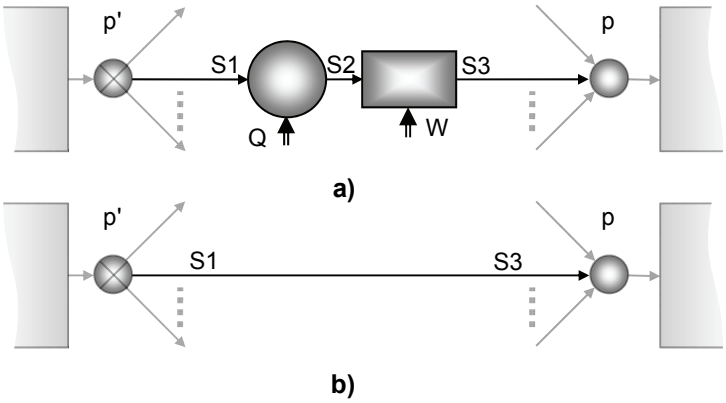


Figure 20: Stream (p', p). **a)** Conditioning stream. **b)** Non conditioning stream.

Conditioning streams $(s \in \mathbf{S}^{\mathbf{C}})$*:* These are modeled by combining equations describing a general heating-cooling operation and a general compression-expansion operation:

$$0 \le \xi_s \le 1 \tag{4-29}$$

$$H_s^{S2} = H_s^{S1} + Q_s \quad , \quad H_s^{S3} = H_s^{S2} + W_s \tag{4-30}$$

$$P_s^{S2} = P_s^{S1} \quad , \quad P_s^{S3} = P_s^{S2} + \Delta P_s \tag{4-31}$$

$$S_s^{S3} = S_s^{S2} \tag{4-32}$$

$$\begin{aligned} H_s^{S2} &= \mathrm{f}^{H}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S2}, P_s^{S2}\right) \quad , \quad H_s^{S3} = \mathrm{f}^{H}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S3}, P_s^{S3}\right) \\ S_s^{S2} &= \mathrm{f}^{S}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S2}, P_s^{S2}\right) \quad , \quad S_s^{S3} = \mathrm{f}^{S}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S3}, P_s^{S3}\right) \end{aligned} \tag{4-33}$$

$$Cc_s = \mathrm{f}^{Cc,HC}\left(Q_s, T_s^{S1}, P_s^{S1}, T_s^{S2}\right) + \mathrm{f}^{Cc,CE}\left(W_s, T^{S3}{}_s, P^{S3}{}_s\right) \tag{4-34}$$

$$Co_s = \mathrm{f}^{Co,HC}\left(Q_s, T_s^{S1}, T_s^{S2}\right) + \mathrm{f}^{Co,CE}\left(W_s\right) \tag{4-35}$$

Equation (4-29) bounds the stream split fractions. Equations (4-30) are the energy balances of the heating-cooling operation (connecting state S1 an S2) and the compression-expansion operation (connecting states S2 and S3), where Q_s and W_s are the stream heating utility and compression power consumptions. Equations (4-31) are hydraulic equations, where ΔP_s is the stream pressure increase. Equation (4-32) is the isentropic condition in the compression-expansion operation. Equations (4-33) relate stream state variables with thermodynamic properties using the same nonlinear mappings used in port models (i.e. surrogate thermodynamics). Equation (4-34) uses nonlinear mappings $\mathrm{f}^{Cc,HC}(\cdot)$ and $\mathrm{f}^{Cc,CE}(\cdot)$ to calculate capital cost of the heating-cooling and the compression-expansion operations included in the stream. Equation (4-35) uses mappings $\mathrm{f}^{Co,HC}(\cdot)$ and $\mathrm{f}^{Co,CE}(\cdot)$ to calculate operative cost of stream conditions operations.

Non conditioning streams $\left(s \in \mathbf{S}^{\mathbf{NC}}\right)$*:* These are modeled as a simple set of equalities:

$$0 \le \xi_s \le 1 \tag{4-36}$$

$$H_s^{\mathrm{S3}} = H_s^{\mathrm{S1}} \tag{4-37}$$

$$S_s^{\mathrm{S3}} = S_s^{\mathrm{S1}} \tag{4-38}$$

$$P_s^{S3} = P_s^{S1} \tag{4-39}$$

$$T_s^{S3} = T_s^{S1} \tag{4-40}$$

$$Cc_s = 0 \tag{4-41}$$

$$Co_s = 0 \tag{4-42}$$

Since no conditioning operation takes place, the intensive and extensive properties of states S1 and S3 have to be the same as indicate in (4-37) to (4-40). For the same reason, capital and operating costs are zero as indicated in (4-41) and (4-42).

4.4.3 General unit operation

For the general unit $\left(u \in \mathbf{U}^{\mathrm{G}}\right)$ presented in Figure 21, including an arbitrary number of inlet and outlet ports, the model is as follows:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{4-43}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + \sum_{l \in \{1,\ldots,L\}} Q_{l,u} + \sum_{o \in \{1,\ldots,O\}} W_{o,u} \tag{4-44}$$

$$P_u = P_p - \underline{\Delta P}_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{4-45}$$

$$P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{4-46}$$

$$F_{c,p'} = \mathrm{f}_{c,p'}^{F,\mathrm{G}}\left(\left[F_{c,p}, T_p\right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta P}_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord\left(p'\right) \neq 1, \forall c \in \mathbf{C} \tag{4-47}$$

$$T_{p'} = \mathrm{f}_{p'}^{T,\mathrm{G}}\left(\left[F_{c,p},T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}},\boldsymbol{\psi}_u,P_u,\underline{\Delta}P_u\right) \quad , \quad \forall p'\in \mathbf{P}_u^{\mathbf{O}} / ord\left(p'\right)\neq 1 \tag{4-48}$$

$$Q_{l,u} = \mathrm{f}_{l,u}^{Q,G}\left(\left[F_{c,p},T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}},\boldsymbol{\psi}_u,P_u,\underline{\Delta}P_u\right) \quad , \quad \forall l \in \left\{1,...,L\right\} \tag{4-49}$$

$$W_{o,u} = \mathrm{f}_{o,u}^{W,G}\left(\left[F_{c,p},T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}},\boldsymbol{\psi}_u,P_u,\underline{\Delta}P_u\right) \quad , \quad \forall o \in \left\{1,...,O\right\} \tag{4-50}$$

$$Cc_u = \mathrm{f}^{Cc,\mathrm{G}}\left(\left[F_{c,p},T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}},\boldsymbol{\psi}_u,P_u,\underline{\Delta}P_u\right) \tag{4-51}$$

$$Co_u = \mathrm{f}^{Co,G}\left(\left[Q_{l,u}\right]_{l\in\left\{1,...,L\right\}},\left[W_{o,u}\right]_{o\in\left\{1,...,O\right\}}\right) \tag{4-52}$$

Equations (4-43) and (4-44) are component and energy balances, where $Q_{l,u}$ and $W_{o,u}$ represent different heat and work additions to unit u. Equation (4-45) represents the unit hydraulics, relating inlet port pressures, the internal unit pressure and the unit pressure drop $\underline{\Delta}P_u$. Equation (4-46) is an internal pressure balance condition. Equations (4-47), (4-48), (4-49), and (4-50) include nonlinear mappings describing how the unit operates by relating the unit model degrees of freedom (i.e. independent inlet port state variables and unit internal variables) with the component flows and temperatures of all but one of the unit outlet ports, as well as the unit utility consumption. Equations (4-51) and (4-52) include nonlinear mappings to estimate unit capital and operative costs.

To illustrate the logic behind the model, consider the degrees of freedom and the following calculation sequence. As indicated by the surrogate mappings, the degrees of freedom for this model include the component flows and temperatures of all inlet streams $\left[F_{c,p},T_p\right]_{c\in\mathbf{C},p\in\mathbf{P}_u^{\mathbf{I}}}$, the unit internal pressure P_u, the unit pressure drop $\underline{\Delta}P_u$, and other unit internal variables $\boldsymbol{\psi}_u$. Using (4-45) and (4-46), all port pressures are calculated. Using (4-47), (4-43) all outlet port component flows are

calculated. Using (4-49), (4-50) all heat and work additions to the units are calculated. Using the enthalpy equations in (4-10), all inlet enthalpy flows are calculated. Using (4-48) and (4-44) and the enthalpy equations in (4-17) an equation system can be posed to find all outlet port enthalpies and temperatures. Finally capital and operating costs can be calculated with (4-51) and (4-52).

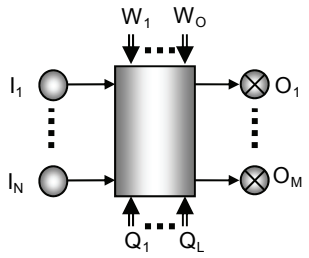


Figure 21: General processing unit.

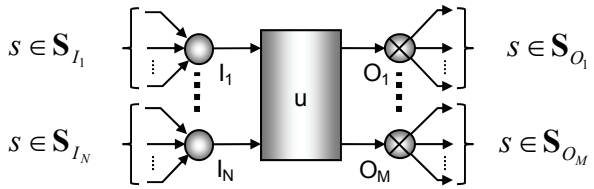
4.4.4 Logic constraints

As presented in section 4.3, Boolean variables Y_u, Y_p, Y_s are used to activate/deactivate different elements in the GDP formulation of a SO problem. Now, let $\left[\mathbf{Y^U} \quad \mathbf{Y^P} \quad \mathbf{Y^S}\right]$ be the array where $\mathbf{Y^U}, \mathbf{Y^P}, \mathbf{Y^S}$ are vectors grouping unit, port and stream Booleans. Fixing the Booleans to values $\left[\mathbf{Y_*^U} \quad \mathbf{Y_*^P} \quad \mathbf{Y_*^S}\right]$ define a process structure within the superstructure. However, not every combination of $True/False$ values leads to a feasible structure or even a candidate worth considered while looking for the optimal solution.

In order to limit the combinatorial complexity in SO formulations, the need for logic constraints has been previously recognized [58],[59], especially when dealing with superstructures featuring a high number of units and/or a rich connectivity. The idea here is to pose logic constraints giving additional structural information in the form of connectivity implications. In addition, some logic constraints based on engineering heuristics can be incorporated. These constraints eliminate from

the structural search space all process configurations which are either infeasible or just unsound from the engineering point of view. For example, all structures with at least one unconnected element (e.g. an active unit with no active upstream or downstream units).

Here we present the logic connectivity constraints to be included in the GDP formulation (4-1)-(4-6), in a form compatible with our superstructure representations and models. This approach is in part based on ideas presented in the literature [58].



$$\mathbf{P}_u^I = \{I_1,...,I_N\} \qquad \{O_1,...,O_M\} = \mathbf{P}_u^O$$

Figure 22: Sets involved in connectivity implications for units and streams.

First of all, our modeling approach assumes a specific number of inlet and outlet ports in each unit. That is, no unit ports are optional. All the ports in a unit are required for its normal operation, and every port in the superstructure belongs to a particular unit. In other words, no port can function by itself (without the unit it is supposed to be attached to) and a unit cannot function if any of its ports is missing. Hence, if a unit port is activated then the unit should be activated and, if a unit is activated then all its ports should be activated. Formally:

$$Y_p \rightarrow Y_u \quad \forall u \in \mathbf{U}, \forall p \in \mathbf{P}_u \qquad , \qquad Y_u \rightarrow Y_{p'} \quad \forall u \in \mathbf{U}, \forall p' \in \mathbf{P}_u \tag{4-53}$$

This in turn can be posed as:

$$Y_u \leftrightarrow Y_p \quad \forall u \in \mathbf{U}, \forall p \in \mathbf{P}_u \tag{4-54}$$

This means that the activation/deactivation status of a unit and all its ports is identical. In other words, port activation can be handled using appropriate unit Booleans, eliminating the need for port Booleans.

Equivalence (4-54) also suggests that the only connectivity implications required in the model relate the activation/deactivation of units and streams. To this respect, when a unit u is activated, at least one stream per unit port should be activated as well, so the unit has all the required connections to operate properly. Formally:

$$Y_u \rightarrow \bigvee_{s \in \mathbf{S}_p^{\mathbf{I}}} Y_s \quad \forall u \in \mathbf{U}, \forall p \in \mathbf{P}_u^{\mathbf{I}} \qquad , \qquad Y_u \rightarrow \bigvee_{s \in \mathbf{S}_{p'}^{\mathbf{O}}} Y_s \quad \forall u \in \mathbf{U}, \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{4-55}$$

The same way, if a stream is activated, the units connected to it should be activated.

$$Y_s \rightarrow Y_u \quad \forall s \in \mathbf{S}, \forall u \in \mathbf{U}_s \tag{4-56}$$

Logic constraints other than these simple connectivity implications can be used in the model to account for engineering judgment, technology limitations or even to adjust the model as to perform a specific type of technology evaluation. For example if we want to evaluate the potential of a series of competing technologies for a particular plant subsystem, it could be useful to enforce the selection of at most one of them. On the other hand, if we want to evaluate the interaction of a number of technologies, such constraint could be relaxed to allow a maximum number of technologies to appear in the solution.

4.5 MINLP reformulations of superstructure element models

Most modeling languages allow the implementation of optimization models in their algebraic form, but it is not common for them to have special provisions allowing a direct implementation of models

in the GDP form (4-1)-(4-6). Notably, the module LogMIP [40] in the environment GAMS (General Algebraic Modeling System) is one of the exceptions. To facilitate the implementation of our modeling approach, we present in this section a general MINLP reformulation of all superstructure element models and the algebraic version of the logic connectivity constraints presented in section 4.4.4. Specific reformulations for superstructure element models are included in appendix C.

4.5.1 General reformulation of superstructure element models

Presented as a disjunctive constraint, the flow based formulation for the superstructure element models presented in section 4.4 have the following general form:

$$\begin{bmatrix} \overset{Y}{} \\ \mathbf{A_1} \cdot x_1 + \mathbf{A}_2 \cdot x_2 + \mathbf{A}_3 \cdot x_3 = 0 \\ \mathbf{B_1} \cdot x_1 + \mathbf{B}_2 \cdot x_2 + \mathbf{B}_3 \cdot x_3 = b \\ x_2 = \mathrm{f}(x_3) \\ x_3^L \le x_3 \le x_3^U \end{bmatrix} \vee \begin{bmatrix} \neg Y \\ x = 0 \end{bmatrix} \tag{4-57}$$

where the Boolean Y is used to activate/deactivate the element. The left disjunct of constraint (4-57) contains an algebraic formulation for a superstructure element model. To explain its structure, compare (4-57) with the unit model introduced in 4.4.3. In this case, the first equation block groups all homogeneous linear equations including component and energy balances, the unit hydraulics and the internal pressure balance condition. The second equation block groups all non-homogeneous linear equations in the model (empty in this case, but required in other models e.g. the outlet port model). The third block groups all nonlinear mappings including those used in the calculation of utility consumption, outlet port temperatures and component flows. The last block delimits the domain for which the nonlinear mappings and the unit model are valid. The model independent variables x_3 include state variables for inlet ports ($\lfloor F_{c,p}, T_p \rfloor_{c \in \mathbf{C}, p \in \mathbf{P}_u^{\mathbf{I}}}$) and internal unit

variables ($\underline{\Delta}P, P_u, \mathbf{\Psi}_u$). Dependent variables x_2 include the component flows and temperatures for all but one of the outlet ports ($\left\lfloor F_{c,p'}, T_{p'} \right\rfloor_{c\in\mathbf{C}, p'\in\mathbf{P}_u^{\mathbf{O}*}}$) as well as utility consumption $\left\lfloor Q_l, W_o \right\rfloor_{l\in\{1,\ldots L\}, o\in\{1,\ldots O\}}$. The remaining variables x_1 include the inlet and outlet port pressures $\left\lfloor P_p \right\rfloor_{p\in\mathbf{P}_u^I\cup\mathbf{P}_u^{\mathbf{O}}}$, as well as the component flows and enthalpy flows of the outlet port not covered in x_2.

As indicated above, the first linear equations block is homogeneous and, by design, it has a non-singular square matrix $\mathbf{A_1}$ which guarantees the implication $\left(\left[x_2 \quad x_3\right]=0\right)\rightarrow\left(x_1=0\right)$. There are general convex hull reformulations for convex disjunctive constraints [60],[61],[62]; however as discussed in section 5.3 the kind of nonlinear mappings $x_2=\mathrm{f}\left(x_3\right)$ used in this work can be reformulated as $x_2=\bar{\mathrm{f}}\left(x_3, y\right)$ to satisfy $0=\bar{\mathrm{f}}(0,0)$ and $x_2=\bar{\mathrm{f}}(x_3,1)$, where y is the binary equivalent of Boolean Y. This reformulation makes it possible to enforce $x=0$ solely by enforcing $x_3=0$. In this way, when such mapping reformulation exists, disjunctive constraint (4-57) can be reformulated into the following mixed-integer form:

$$\begin{bmatrix} \mathbf{A_1}\cdot x_1+\mathbf{A}_2\cdot x_2+\mathbf{A}_3\cdot x_3=0 \\ \mathbf{B_1}\cdot x_1+\mathbf{B}_2\cdot x_2+\mathbf{B}_3\cdot x_3=b\cdot y \\ x_2=\bar{\mathrm{f}}\left(x_3, y\right) \\ x_3^L\cdot y\le x_3\le x_3^U\cdot y \end{bmatrix} \tag{4-58}$$

Specific MINLP reformulations for component flow based element models and mol fraction based element models are included in appendix C.

4.5.2 Logic constraints

As indicated earlier, we only need unit and stream Booleans to formulate the logic constraints. For a mixed integer reformulation, those implications are transformed into linear constraints [58],[59] involving the binary variables y_u and y_s.

The following connectivity constraints enforce the activation of at least one stream connected to every port of a unit u when it is activated:

$$y_u \le \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} y_s \quad \forall u \in \mathbf{U}, \forall p \in \mathbf{P}_u^{\mathbf{I}} \qquad , \qquad y_u \le \sum_{s \in \mathbf{S}_{p'}^{\mathbf{O}}} y_s \quad \forall u \in \mathbf{U}, \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{4-59}$$

In the same way, the following constraint enforces the activation of unit u if a stream connected to it is activated.

$$y_s \le y_u \quad \forall s \in \mathbf{S}, \forall u \in \mathbf{U}_s \tag{4-60}$$

As with the connectivity implications (4-59),(4-60), any other logic constraint can be posed as a linear algebraic constraint in terms of a number of binary variables.

Chapter 5

SURROGATE MODEL DESIGN

As part of chapter 4, models for specific superstructure elements where introduced, including a general unit operation model. As discussed, the element models are hybrid surrogates which include most linear equations found in rigorous models, as well as sets of nonlinear mappings replacing entire blocks of complicated equations relating the element independent variables with some key dependent variables.

The use of such mappings in this hybrid modeling approach generally comes with the elimination of a significant number of internal variables included in the original rigorous models, as well as a decisive reduction in the model mathematical complexity when the form of the surrogate non-linear mapping is properly selected.

In this chapter we will discuss key aspects regarding the element models presented in chapter 4, including the rationale behind their mathematical form, a systematic methodology to select the model independent variables, the form of the mappings used in this work, and the mapping reformulation according with the conditions presented in section 4.5.1. Finally, the use of operational regimes in the creation of accurate surrogate models is presented.

5.1 Basic mathematical form

As presented in the left disjunct of (4-57), all models include a block of homogenous linear equations (e.g. material and energy balances, hydraulics, isothermal and isobaric conditions, etc.), a block of non-homogeneous equations (e.g. (4-11)), a block of nonlinear surrogate mappings which relates the model independent variables x_3 with a subset of dependent variables x_2, and finally, a block of inequality constraints bounding the model domain. We use this mathematical form to balance the prediction capabilities of the resulting models and their mathematical complexity. That is, we try to incorporate in this hybrid surrogate as many of the key equations describing the behavior of the unit as possible (preferably linear in order to maintain a low mathematical complexity), and replace the rest with nonlinear mappings. Fortunately, the key conservation principles (i.e. balance equations) can be posed as homogeneous linear equations, and hence they are always included in the first equation block.

These ideas apply to both the F_c-formulations and the $F_c - x_c$-formulations; however, in the latter, the nonlinear mappings are used to calculate intensive variables (e.g. molar enthalpies and entropies) which can only be related to extensive ones (e.g. enthalpy and entropy flows) using equations with bilinear terms. These bilinearities add to the complexity of the model, but as discussed in section 2.1, special provisions can be used to deal with them.

5.2 Selection of independent variables

Selecting the independent variables is one of the key aspects of designing a proper surrogate model, since it can affect the total number of model variables and the final form of the nonlinear mappings

included in the surrogate. This section presents a systematic methodology we have developed to support the selection of independent model variables.

As previously indicated, one of the main goals of developing surrogate models for superstructure elements is to reduce the mathematical complexity of the resulting SO model; a reduction which can only come from reducing the total number of variables, equations and/or their mathematical form. These reductions can be achieved by replacing the complex blocks of equations with surrogate mappings. In fact, the methodology presented here guarantees the resulting surrogate models will include as few variables as possible.

The methodology starts with a variable analyst, where we identify the way a particular element in the superstructure is connected to the rest, and the way its rigorous model is connected to the rest of the superstructure model.

For a particular superstructure element and its associated surrogate we define:

$\mathbf{I}_{\mathbf{EM}}$: Set of independent variables. Fixing these variables transforms the element model into a structurally nonsingular equation system.

$\mathbf{D}_{\mathbf{EM}}$: Set of dependent variables (i.e. complement of $\mathbf{I}_{\mathbf{EM}}$)

$\mathbf{N}_{\mathbf{EM}}$: Set of natural independent variables. These are the variables used in commercial process simulation models to close the degrees of freedom.

$\mathbf{F}_{\mathbf{EM}}$: Set of variables fixed in the problem statement.

$\mathbf{C}_{\mathbf{EM}}$: Set of connecting variables. These are variables connecting the element model with the rest of the optimization model; that is, variables in the original element model which appear in other parts of the optimization problem: the objective function, other element models or other constraints.

$\mathbf{I_S}$: Set of surrogate independent variables.

$\mathbf{D_S}$: Set of surrogate dependent variables.

In principle, any element model establishes an implicit mapping between the space defined by the variables in $\mathbf{I_{EM}}$ (here called the model domain) and the space defined by the variables in $\mathbf{D_{EM}}$ (here called the model range). Now, at the superstructure model level, a rigorous element model can be replaced with a surrogate if such surrogate enforces the same constraints the original model enforces among variables which appear anywhere outside the element model. For this condition to be satisfied, it is necessary and sufficient that the surrogate enforces the same relationship the original model enforces between the free independent variables (i.e. $\mathbf{I_{EM}} / \mathbf{F_{EM}}$) and the connecting variables $\mathbf{C_{EM}}$. The rest of the variables are internal to the model, they are required for the original model to be complete, but irrelevant at the superstructure modeling level (i.e. they are not part of the rest of the superstructure model).

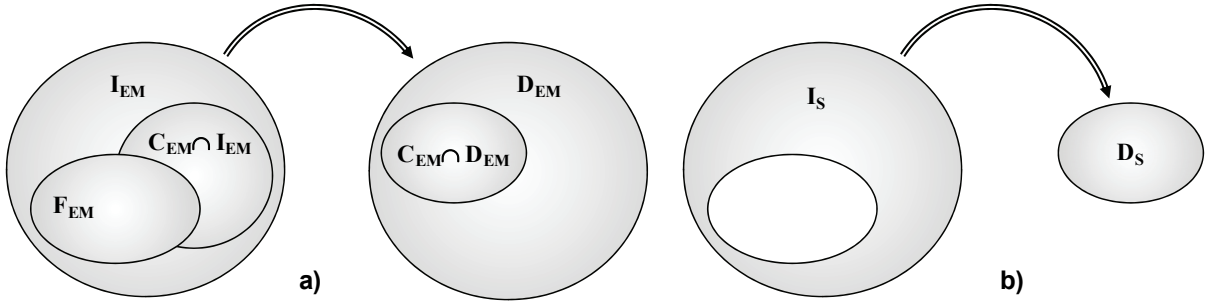


Figure 23: Venn diagrams and representation of the mappings between variable sets. **a)** Original unit model. **b)** Dimensionally reduced surrogate model.

Figure 23 presents the different variable sets and the way they are related by the original and the surrogate models. It also shows how the surrogate model can be conceived as a dimensionally reduced version of the implicit mapping, established by the original detailed element model,

between independent and dependent variables. Notice that the total number of variables in the surrogate $\left(|\mathbf{I_S}|+|\mathbf{D_S}|\right)$ can be significantly lower than in the original model $\left(|\mathbf{I_{EM}}|+|\mathbf{D_{EM}}|\right)$, facilitating the construction and implementation of the surrogate.

For a particular superstructure, and a particular process design problem, $\mathbf{F_{EM}}$ and $\mathbf{C_{EM}}$ are specified. The problem statement fixes the members of $\mathbf{F_{EM}}$, while the superstructure topology and the form of the objective function fix $\mathbf{C_{EM}}$. However, there are multiple ways to partition the variables into sets $\mathbf{I_{EM}}$ and $\mathbf{D_{EM}}$. For a partition to be appropriate, the minimum condition to satisfy is that the equation system resulted from fixing the independent variable $\mathbf{I_{EM}}$ must be structurally nonsingular. That is, an equation system with no over or under specified sub-blocks of equations. In order to systematically identify this variable partitioning, we propose the following mixed integer program:

$$\max \sum_{j\in\mathbf{J}} w_j \cdot y_j \tag{5-1}$$

$$s.t. \sum_{j:(j,k)\in\mathrm{A}} x_{jk} = 1 \quad \forall k \in \mathrm{K} \tag{5-2}$$

$$\sum_{k:(j,k)\in\mathrm{A}} x_{jk} = 1 - y_j \quad \forall j \in \mathrm{J} \tag{5-3}$$

$$x_{jk}, y_j \in \{0,1\} \tag{5-4}$$

Here, $j \in \mathrm{J}$ and $k \in \mathrm{K}$ are, respectively, indices for variables and equations in the element model. Variables y_j are selection binaries which indicate the variables selected to be part of $\mathbf{I_{EM}}$ (i.e. $\mathbf{I_{EM}} = \{z_j / y_j = 1\}$. Variables $x_{j,k}$ are binaries matching variables to equations. The parameters w_j

are preference coefficients whose values are adjusted to favor the selection of some variables over others. The set $\mathbf{A}$ contains the edges of the bipartite graph representing the variable-equation incidence in the element model (i.e. a pair (j,k) is part of $\mathbf{A}$ if variable j appears in equation k). Constraints (5-2) and (5-3) enforce a perfect matching between equations and the variables in $\mathbf{D}_{\mathbf{EM}}$ (i.e. (5-2) enforces the matching of every equation with exactly one dependent variable, and (5-3) enforces the matching of every dependent variable with exactly one equation). This perfect matching is equivalent to the structural non-singularity condition mentioned before [63]. The perfect matching idea is illustrated in Figure 24 for a hypothetic model with 8 variables and 5 equations

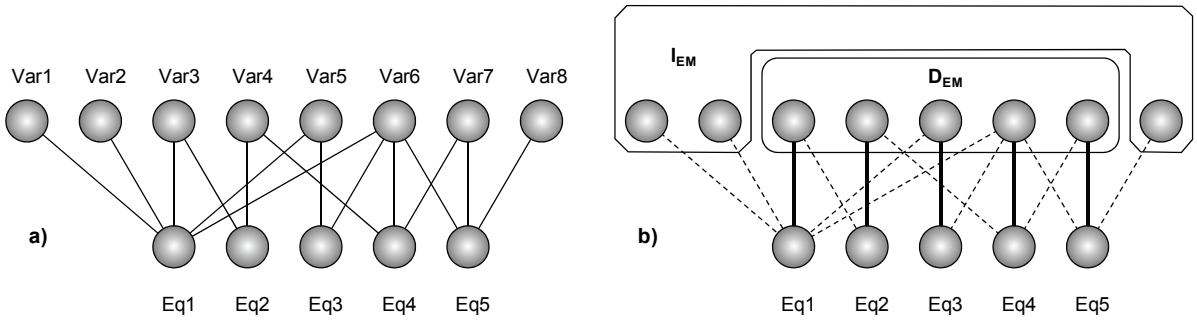


Figure 24: Equation system representations. **a)** Bipartite representation for a unit model with 8 variables and 5 equations. **b)** Perfect matching (bold edges) and identification of sets $\mathbf{I}_{\mathbf{EM}}$, $\mathbf{D}_{\mathbf{EM}}$.

To use effectively the model (5-1)-(5-4) it is important to account for the following. First, the variables in $\mathbf{F}_{\mathbf{EM}}$ have to be part of $\mathbf{I}_{\mathbf{EM}}$ since they can never be part of $\mathbf{D}_{\mathbf{EM}}$. This means that their selection binaries have to be set to 1 before solving (5-1)-(5-4). If fixing these variables makes the model infeasible, we conclude that the specification scheme (i.e. defined by $\mathbf{F}_{\mathbf{EM}}$) is inconsistent with the model structure, and it has to be modified. Second, once $\mathbf{I}_{\mathbf{EM}}$ is identified, the independent and dependent variables for the surrogate are given by:

$$\mathbf{I_S} = \mathbf{I_{EM}} \setminus \mathbf{F_{EM}} \tag{5-5}$$

$$\mathbf{D_S} = \mathbf{C_{EM}} \setminus \mathbf{I_{EM}} \tag{5-6}$$

These expressions are consistent with the representation in Figure 23. Also, they lead to:

$$\left|\mathbf{I_S}\right| = \left|\mathbf{I_{EM}}\right| - \left|\mathbf{I_{EM}} \cap \mathbf{F_{EM}}\right| = \left|\mathbf{I_{EM}}\right| - \left|\mathbf{F_{EM}}\right| \tag{5-7}$$

$$\left|\mathbf{D_S}\right| = \left|\mathbf{C_{EM}}\right| - \left|\mathbf{C_{EM}} \cap \mathbf{I_{EM}}\right| \tag{5-8}$$

It is important to notice that (5-8) shows how the selection of $\mathbf{I_{EM}}$ can affect the number of surrogate dependent variables. It also suggests that, in order to minimize the number of surrogate variables, the set $\mathbf{I_{EM}}$ should include as many of the variables in $\mathbf{C_{EM}}$ as possible. Finally, if the data required to build the surrogate models is obtained from commercial process simulators, it is convenient to include in $\mathbf{I_{EM}}$ as many of the natural degrees of freedom $\mathbf{N_{EM}}$ as possible. This reduces the use of special iterative calculation procedures common to all process simulators (e.g. HYSYS adjust functions, ASPEN PLUS process specifications, etc.) and the associated computational burden. These requirements can be satisfied by selecting large w_j for variables in $\mathbf{C_{EM}} \cup \mathbf{N_{EM}}$

The standard element models presented in section chapter 4 and 5 were developed using this methodology while assuming no fixed variables (i.e. $\mathbf{F_{EM}}$ is the empty set), and connecting variable sets composed mostly by the element input and output streams state variables and costs.

5.3 Non-linear surrogate mappings

So far, the element models have been presented using general non-linear mappings, but no details about the mappings mathematical structure have been discussed. The mappings structure is

important since it will be pervasive in the final superstructure model, determining in part how feasible it is to find significant solutions using current numerical solvers.

As indicated previously, a basic requirement for a proper mapping has to do with their reformulation as presented in section 4.5.1. Particularly, we need a mapping $z = \mathrm{f}(x)$ to allow a reformulation $z = \bar{\mathrm{f}}(x, y)$ which satisfies $z = \bar{\mathrm{f}}(x,1)$ and $0 = \bar{\mathrm{f}}(0,0)$, where y is an activation/deactivation binary. Although several mapping forms can satisfy this condition, in this work we use Artificial Neural Networks (ANNs), due to their simple structure, simple implementation and their excellent fitting capabilities. In particular we use feed forward ANNs (FFANNs), where the mapping between an independent variable vector x^0 and a dependent one x^K is given by the following equation block [64] :

$$x^k = \mathrm{f}^k\left(W^k \cdot x^{k-1} + b^k\right) \quad , \quad \forall k = 1,\ldots, K \tag{5-9}$$

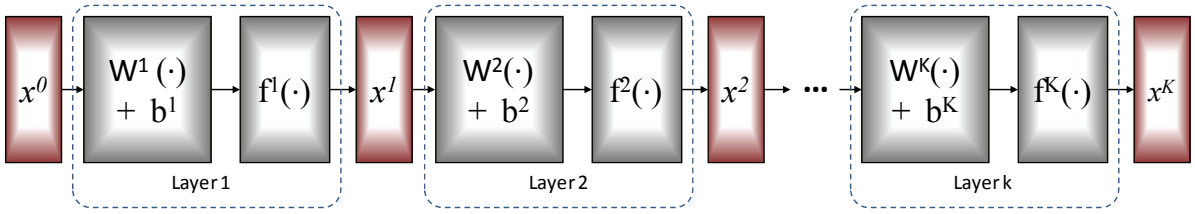


Figure 25: Graphic representation of multilayer feed forward artificial network.

Here, the artificial neural network is composed by a number of layers, denoted by super index k. For each layer, the number of components in outlet vector x^k is referred to as the number of neurons. In a feed forward configuration, each layer k maps the outlet vector of the previous layer x^{k-1} into its outlet vector x^k. The layer mapping starts with a linear transformation using a weight matrix W^k and a bias vector b^k, followed by the use of a so called "transfer" or "activation" function $\mathrm{f}^k(\cdot)$ (i.e. a scalar function applied to every component of what results from the linear

transformation). The last layer in a network is called the outlet layer, and all the rest are called hidden layers.

The non-linear fitting capabilities of the complete mapping $x^K = \mathrm{f}\left(x^0\right)$ come from the adjustable coefficients W^k, b^k and the form of the nonlinear transfer functions $\mathrm{f}^k(\cdot)$. In fact, a theoretical result shows that any nonlinear mapping with a finite number of discontinuities can be approximated with arbitrary precision by a feed forward network with two layers; the first one featuring a sigmoid transfer function, and the second one featuring a linear transfer function [64]. This reduces the problem of selecting a proper FFANNs architecture to the selection of a proper dimensionality of the hidden space (i.e. number of rows in W^1 = number of neurons in the hidden layer).

In our work we use of FFANNs with one or two hidden layers featuring $\tanh(\cdot)$ transfer functions, and an outlet layer with a linear transfer function. This form has proven appropriate in the characterization of any of the nonlinear mappings required to build hybrid surrogate models for superstructure elements, while providing a convenient mathematical structure. This allows for the development of solution strategies exclusively based on such structure. In addition, FFANNs can be reformulated to incorporate activation/deactivation binaries as follows:

$$\begin{aligned} & x^k = \mathrm{f}^k\left(v^k\right) \\ & v^k = W^k \cdot x^{k-1} + b^k \cdot y \\ & L \cdot y \le x^0 \le U \cdot y \end{aligned} \quad , \quad \forall k = 1,\ldots,K \qquad (5\text{-}10)$$

Here L and U are proper upper and lower bounds on the components of x^0.

The above reformulation has two major advantages: (a) unlike standard methods [60],[61],[62] , no variable transformation that results in potential numerical difficulties is necessary, (b) by defining a

new variable v^k, no special transformation was necessary for the nonlinear equation $x^k = \mathrm{f}^k(.)$, and instead, only its linear argument $W^k \cdot x^{k-1} + b^k$ was reformulated to incorporate the activation/deactivation binary y. The development of this reformulation exploits the fact that the transfer functions used (i.e. $\tanh(\cdot)$ and linear functions) satisfy $0 = \mathrm{f}^k(0)$. Thus, when $y = 1$ the original surrogate relation is enforced; and, when $y = 0$, the surrogate model is deactivated, leading to $x^0 = x^1 = \ldots = x^K = 0$. The effectiveness of this reformulation is another reason why we chose to use ANN surrogate mappings. Finally, note that with the proposed strategy the myriad of nonlinearity types, which can be seen in rigorous modeling of unit operations and other superstructure elements, are reduced to only one: the $\tanh(\cdot)$ function. This makes possible the development of solution methods that exploit this special structure in the final superstructure optimization model, which could be based on the convexification techniques similar to the ones presented by Westerlund [65],[66] in search for global solutions.

5.4 Unit operation regimes

With the modeling strategy presented in this work, all nonlinear behavior of a particular superstructure element is characterized by a small number of nonlinear mappings. However, the accurate fitting of surrogate models to the real nonlinear behavior of processing units can sometimes be challenging. In particular, the kind of ANNs used here, can approximate continuous smooth functions with great accuracy while using a moderate number of hidden neurons, but they require large numbers of hidden neurons to fit discontinuous and sometimes no differentiable relations. As discussed in section 6.2, it is preferred the use of ANNs with as few neurons as possible, since they bring a reduced computational burden during the fitting process as well as simpler element and superstructure models (i.e. less mapping equations and internal variables).

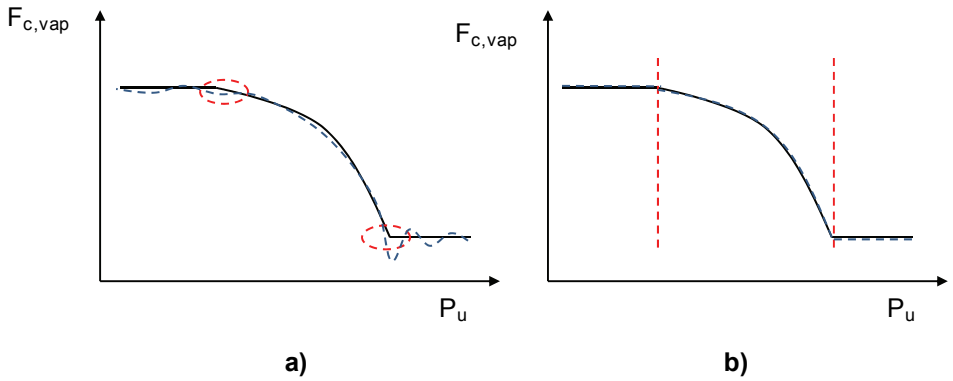


Figure 26: Vapor molar flows vs. pressure for flash vessel with constant feed: **a)** Non-smooth points and surrogate mapping fitting for the entire pressure range. **b)** Operational regions and fitting using a different mapping in each region.

In order to reduce mapping complexity, it is possible to partition the surrogate model domain in different operational regions, each one characterized by a proper set of mappings. To illustrate the idea consider a flash vessel model. Figure 26 shows how vapor molar flows depend on pressure for a flash vessel with a given feed stream. The figure highlights the non-smooth points (i.e. mixture bubble and dew points) around which the ANN does not fit the real behavior very well when few hidden neurons are used. However, if the temperature domain (the only independent variable considered here) is divided in three different sub-domains, each of them using a particular mapping, the behavior could be adjusted with better accuracy.

The same idea can be extended to a multidimensional surrogate domain where not only the unit pressure but also the feed stream state variables are independent variables. As presented in chapter 6, the surrogate mappings are built by fitting results from a number of simulation cases featuring the original detailed models for each superstructure element. Each simulation case is defined by a sample point in the domain. For a flash vessel, the sampling of the domain provides a number of operating cases which can be classified as Vapor operation, Liquid operation or Vapor-Liquid

operation. Later, using Support Vector Machines, it is possible to build approximated boundaries separating such operational regions, and specific surrogate mappings may then be built for each region. For vapor molar flows in flash units, the mappings for both Liquid operation and Vapor operation are trivial (i.e. in the first case the vapor molar flows are zero, and in the second they are equal to the feed molar flows), making the unit model very simple in those regions.

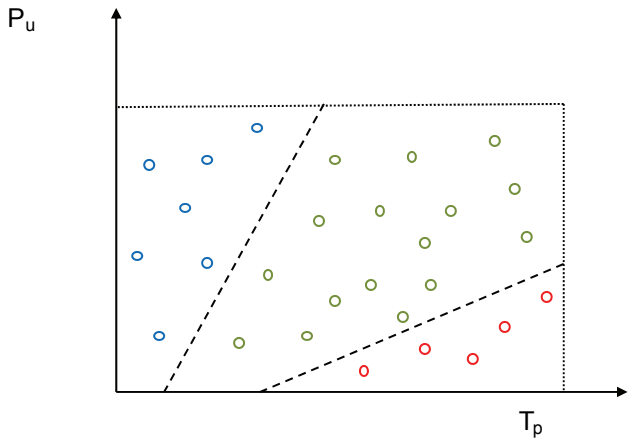


Figure 27: Illustration of operational regions dividing the model domain (Unit pressure)-(Feed temperature) for a flash vessel.

5.5 Surrogate model design: Illustrative example

To illustrate the design of a surrogate model, consider the optimal design and operation of the stirred tank reactor (CSTR) presented in Figure 28, for a fixed inlet stream and reactor pressure drop.

The optimization model has the form:

$$\max \quad f_{Obj}\left(F_{c^*,O}, V_R, Q_R\right) \qquad (5\text{-}11)$$
$$s.t.$$

$$f_{CSTR}(...) = 0 \qquad (5\text{-}12)$$

$$T_O \le T^{Up} \tag{5-13}$$

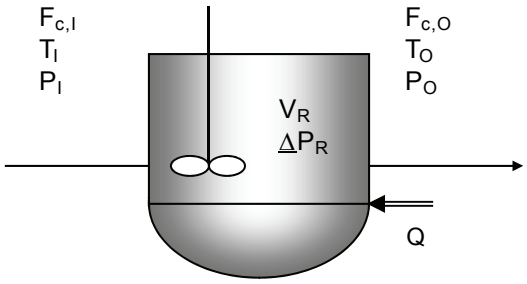


Figure 28: Simple continuous stirred tank reactor.

Here, the objective function (5-11) includes a revenue term which is a function of the outlet flow $F_{c^*,O}$ of a single valuable product c^*, a capital cost term which is a function of the reactor volume V_R, and an operating cost term which is a function of the reactor heating requirement Q_R. The constraints include the reactor model (5-12) and a safety consideration expressed by an upper bound on the reactor operating temperature (5-13).

We are interested in finding the sets of dependent and independent variables to build a surrogate which could replace the entire CSTR model (5-12). To answer this question we need to analyze an explicit form of the CSTR model, which is given by:

$$F_{c,I} + V_R \cdot \sum_{r \in \mathbf{R}} \nu_{c,r} \cdot r_r = F_{c,O} \tag{5-14}$$

$$h_I \cdot \sum_{c \in \mathbf{C}} F_{c,I} + Q_R = h_O \cdot \sum_{c \in \mathbf{C}} F_{c,O} \tag{5-15}$$

$$P_I - \underline{\Delta} P_R = P_O \tag{5-16}$$

$$r_r = \mathrm{f}_\mathrm{r}^\mathrm{r}\left(T_O, \left[C_{c,O}\right]_{\forall c \in \mathbf{C}}\right) \quad , \quad \forall r \in \mathbf{R} \tag{5-17}$$

$$x_{c,I} \cdot \sum_{c' \in \mathbf{C}} F_{c',I} = F_{c,I} \quad , \quad x_{c,O} \cdot \sum_{c' \in \mathbf{C}} F_{c',O} = F_{c,O} \quad , \quad \forall c \in \mathbf{C} \tag{5-18}$$

$$\upsilon_O \cdot C_{c,O} = x_{c,O} \tag{5-19}$$

$$\begin{aligned} h_I &= \mathrm{f}^{h_I}\left(T_I, P_I, \left[x_{c,I}\right]_{\forall c \in \mathbf{C}}\right) \\ h_O &= \mathrm{f}^{h_O}\left(T_O, P_O, \left[x_{c,O}\right]_{\forall c \in \mathbf{C}}\right) \end{aligned} \quad , \quad \upsilon_O = \mathrm{f}^{\upsilon_O}\left(T_O, P_O, \left[x_{c,O}\right]_{\forall c \in \mathbf{C}}\right) \tag{5-20}$$

Equation (5-14) and (5-15) are component and energy balances respectively. Equation (5-16) represents the unit hydraulics. Equation (5-17) contains the system kinetics, where reaction rates are expressed in terms of temperature and molar concentrations. Equations (5-18) relate component molar flows and molar fractions for the reactor inlet and outlet. Equation (5-19) calculates molar concentrations inside the reactor using molar fractions and the mixture molar volume. Finally, equations (5-20) relate state variables with thermodynamic properties: molar enthalpy and molar volume.

According to the optimization problem statement, the feed conditions and reactor pressure drop are fixed, hence:

$$\mathbf{F_{EM}} = \left\{T_I, P_I, \left[F_{c,I}\right]_{\forall c \in \mathbf{C}}, \underline{\Delta P_R}\right\} \tag{5-21}$$

The form of (5-11)-(5-13) indicates that the CSTR model is connected to the objective function (5-11) through the main product flow, the reactor volume, and heat duty. Additionally, it is also connected to (5-13) trough the reactor temperature. Hence:

$$\mathbf{C_{EM}} = \left\{T_O, F_{c^*,O}, V_R, Q_R\right\} \tag{5-22}$$

Finally, in commercial process simulators, the degrees of freedom of a CSTR model include the feed stream state variables, unit pressure drop, reaction volume and operating temperature. Hence:

$$\mathbf{N_{EM}} = \left\{T_I, P_I, \left[F_{c,I}\right]_{\forall c \in \mathbf{C}}, \underline{\Delta P}_R, V_R, T_O\right\} \tag{5-23}$$

Solving model (5-1)-(5-4) leads to:

$$\mathbf{I_{EM}} = \mathbf{N_{EM}} = \left\{T_I, P_I, \left[F_{c,I}\right]_{\forall c \in \mathbf{C}}, \underline{\Delta P}_R, V_R, T_O\right\} \tag{5-24}$$

Finally, using (5-5) and (5-6):

$$\mathbf{I}_S = \left\{V_R, T_O\right\} \quad , \quad \mathbf{D_S} = \left\{F_{c^*,O}, \underline{Q}_R\right\} \tag{5-25}$$

This means that all equations and variables in (5-14)-(5-20) can be replaced by a mapping with 2 independent variables and 2 dependent variables. Part of the significant reduction in the model complexity was possible due to the large number of fixed variables. However, in a general situation with no fixed variables it is more convenient to perform a similar analysis limited to the subset of complicated equations. This results in the kind of hybrid surrogates presented in chapter 4 and appendix D.

Chapter 6

SURROGATE MAPPING GENERATION

For an element surrogate model to be useful, it is necessary to build the required nonlinear mappings. As previously mentioned, these surrogate mappings are generated via data fitting, where sample points from the model domain (i.e. Cartesian products of the domains of variables in $\mathbf{I}_S$), are used to specify a number of simulation cases based on realistic models (i.e. using complex kinetics, thermodynamic calculations etc.) The simulation results provide the values of the dependent variables; in other words, and effective sampling of the model range (i.e. Cartesian products of the range of variables in $\mathbf{D}_S$).

Once samples are generated, an algorithm is used to fit the nonlinear mappings. Depending on the type of mapping, specific algorithms have been developed. All of them try to minimize a measure of the deviations between the data and what the mappings predict, by adjusting the mapping coefficients. In the case of ANNs the basic algorithm is called back propagation training, a name referring to the way deviation gradients are calculated and used to identify descent directions in a minimization procedure.

This chapter includes all the details regarding the generation of sample values (i.e. material and energy balance plus equipment sizing and costing), the way they are used in training the ANN mappings, and the way the resulting mappings are validated.

6.1 Data sampling

In general, the domain sampling has to provide a statistically significant set of points capturing the underlying relationship between the dependent and independent variables as characterized by the detailed model connecting them. In other words, since the generation of a large number of sampling points can be very computationally expensive, one of the main objectives here is to maximize the information gathered from a limited number of sample points. This can be done using a proper sample distribution in the domain, where every sampling point is considerably different from the rest, and the number of distinct values considered for each variable is as big as possible. From this perspective, the use of grid sampling is not very convenient since it results in groups of points sharing the values of at least one variable (i.e. all points in every grid line share all but one of the values, all points in every grid plane share all but two values, etc.) and a number of values considered for each variable which is much less than the number of sample points. Random sampling with a uniform distribution can overcome these limitations, but it can also lead to data clustering and improperly sampled sub-regions. To avoid all these problems while maximizing the sample information content for a limited number of samples, special experimental design techniques such as Latin Hypercube, Hammersley sequences, and D-optimal designs, have to be used. The classic Latin Hypercube and its modifications are very popular in general applications [67], while D-optimal designs are advantageous since they account for the type of fitting model during the generation of sampling points [68].

For this work we use Latin Hypercube Sampling (LHS). To explain how it works, consider a multivariable domain defined by upper and lower bounds for every variable. In LHS, every variable domain is divided in as many sub-intervals as the number of points Np to sample. The sub-intervals define a rectangular grid dividing the multivariable domain in Np^N hypercube sub-domains, where N is the number of variables. Next, Np of these sub-domains are selected in a way that none of them share any of the sub-intervals defining them. Finally, a point is randomly generated in each selected sub-domain. A representation of a Latin Hypercube sampling is shown in Figure 29.

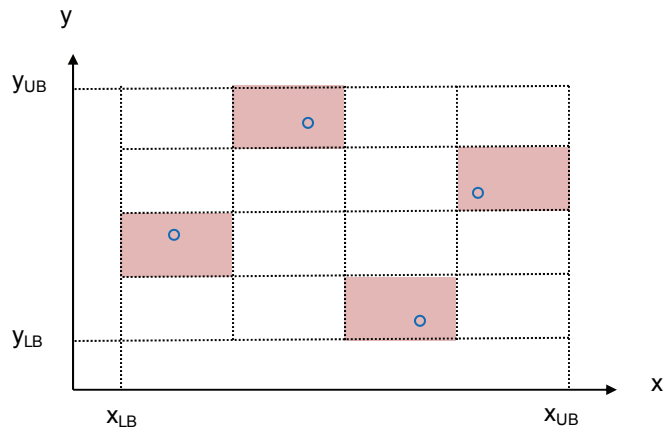


Figure 29: Latin Hypercube sampling of a 2D space with 4 sample points. Every variable domain is divided in 4 sub-intervals, 4^2 rectangular sub-domains are generated, 4 sub-domains are selected (red) and one sample point in each of them is randomly generated. Only one sub-domain is selected per row/column.

The hypercube sub-domain selection guarantees that the total number of sample values for each variable is as high as possible (i.e. equal to the number of points since no two points can share values for any of their components), while reducing the possibility of data clustering. In addition, some simple iterative algorithms can be implemented in order to enhance the sampling by

maximizing the minimum distance between points, or minimizing the sampling correlation coefficient.

Effective domain sampling becomes more complicated whenever $\mathbf{I_S}$ includes groups of variable related by constraints instead of being simply bounded by fixed values. There is a particular interest in variable groups such as component mole fractions $\{x_c, \forall c \in \mathbf{C}\}$ for which the following constraints apply:

$$\sum_{c \in \mathbf{C}} x_c = 1 \quad , \quad x_c \geq 0, \forall c \in \mathbf{C} \tag{6-1}$$

Constraints (6-1) define a simplex like region which cannot be sampled using LHS (see Figure 30 b) for a case with 3 components). A naive approach could involve the hypercube sampling of component flows $\{F_c, \forall c \in \mathbf{C}\}$ within proper lower and upper bounds, followed by the transformation:

$$x_c = F_c \,/ \sum_{c' \in \mathbf{C}} F_{c'}, \forall c \in \mathbf{C} \tag{6-2}$$

However, this leads to a non uniform distribution in the composition domain where the points tend to concentrate around the simplex geometric center. See Figure 31-a.

To obtain a truly uniform random sampling in the simplex region of the composition space, while accounting for the fact that only $|\mathbf{C}|-1$ of the $|\mathbf{C}|$ mol fractions are independent, (6-1) is rewritten as:

$$\sum_{c \in \mathbf{C}^*} x_c \leq 1 \quad , \quad x_c \geq 0, \forall c \in \mathbf{C}^* = \{c \in \mathbf{C} \,/\, ord(c) < |\mathbf{C}|\} \tag{6-3}$$

$$x_{c'} = 1 - \sum_{c \in \mathbf{C}^*} x_c \quad , \quad \forall c / ord(c') = |\mathbf{C}| \tag{6-4}$$

The equivalence of (6-1) and (6-3)-(6-4) makes it possible to reduce the sampling of all mol fractions $\{x_c, \forall c \in \mathbf{C}\}$ to the sampling of a subset of independent mol fractions $\{x_{c^*}, \forall c \in \mathbf{C}^*\}$, where (6-4) is used to calculate the remaining coordinate. Figure 30 presents this idea.

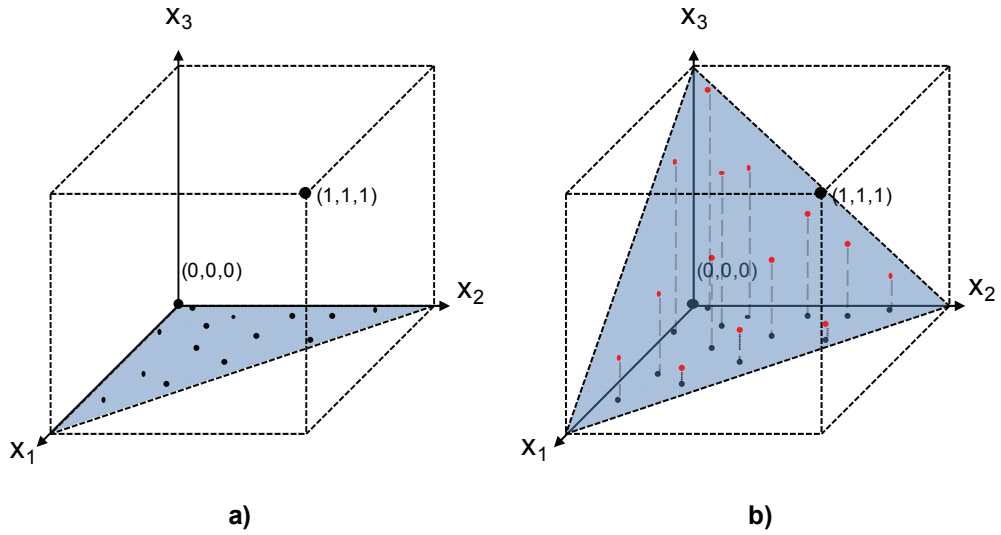


Figure 30: Sampling of a 2D simplex region (blue). **a)** Sampling of independent variables x_1-x_2 (black points). **b)** Lifting and final samples in x_1-x_2-x_3 space (red points).

The uniform sampling of the simplex region defined by (6-3) has been addressed in the literature [69]. In particular the variable $\mathbf{x} = (x_1, \ldots, x_q)$ is uniformly distributed in simplex $\mathbf{S}^q$ (i.e. $\mathbf{x} \sim U(\mathbf{S}^q)$) if $\mathbf{x}$ has the following stochastic representation:

$$x_i = (1 - \phi_{i-1}) \prod_{j=i}^{q} \phi_j \quad , \quad \phi_i = Beta(i,1) \quad , \quad i = 1, \ldots, q \quad , \quad \phi_0 \equiv 0 \tag{6-5}$$

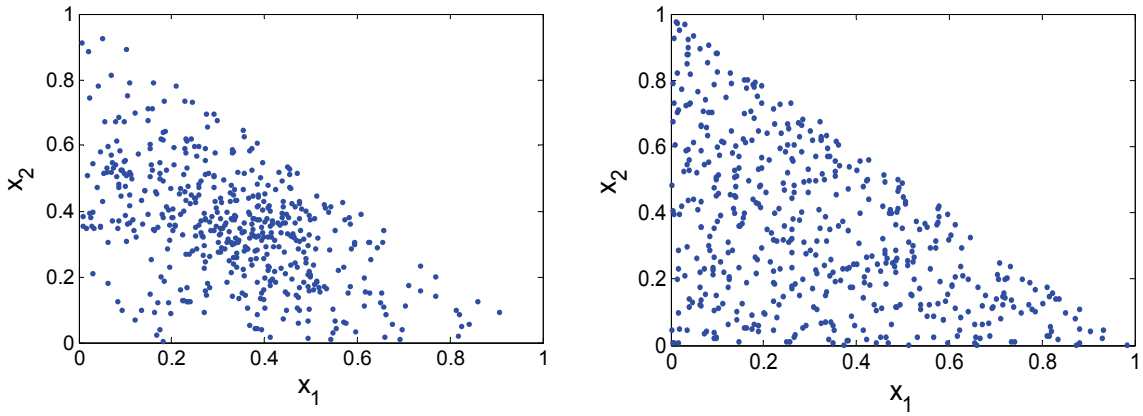


Figure 31: A 500 point sampling of composition space x_1-x_2-x_3 projected to the x_1-x_2 plane. **a)** Non uniform distribution obtained using (6-2) on a uniform sampling of component flows. **b)** Uniform sampling obtained with (6-5).

Groups of variables defining a rectangular domain trough upper and lower bounds exclusively (e.g. temperature, pressure, total flow) are henceforth called rectangular groups, while groups of variables defining a simplex like domain as the one in defined by (6-1) (e.g. mol fractions) are henceforth called simplex groups. Here we propose the following sampling strategy: rectangular domain are sampled using Latin Hypercube, simplex domains are sampled using (6-5), and hybrid domains containing one rectangular group of variables and one or more simplex groups of variables (e.g. absorption columns where we have two sets of feed stream compositions) are sampled by staking the values of samplings taken independently for each variable group domain. Figure 32 a) illustrate this idea, for a hybrid domain in $T-x_1-x_2$. For this case, if $\mathbf{x}_{(2\times Np)}$ is a matrix containing the x_1-x_2 domain sampling, and $\mathbf{T}_{(1\times Np)}$ is a matrix containing the T domain sampling, the sampling of the hybrid domain $T-x_1-x_2$ is given by, $\left[\mathbf{T}^{\mathrm{T}} \quad \mathbf{x}^{\mathrm{T}}\right]^{\mathrm{T}}$.

Other than the samples generated using the mentioned strategy, we include additional points, which correspond to all possible combinations of extreme and intermediate points in each group domain. Such points are important in improving the surrogate mapping behavior near extreme regions of the hybrid domain. Figure 32 b) illustrates the additional sampling.

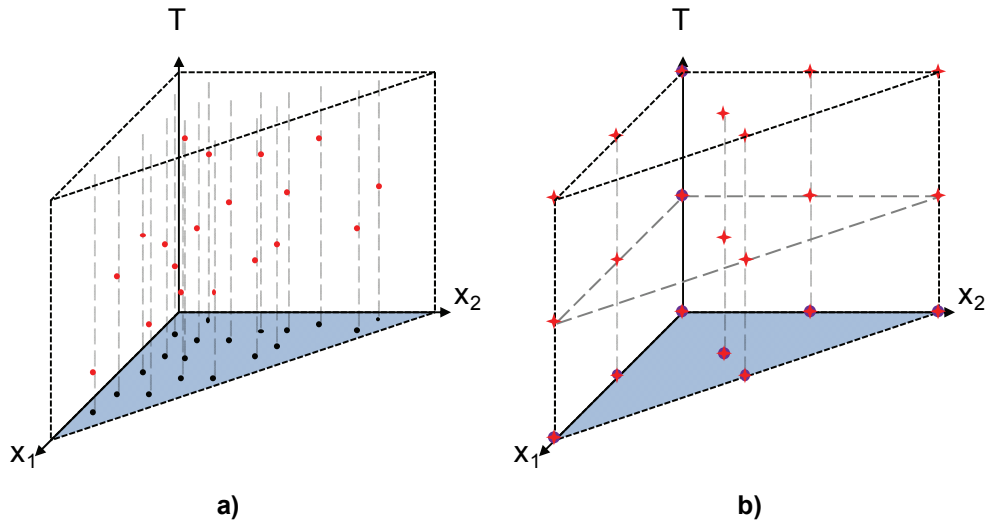


Figure 32: Sampling of a hybrid domain. Rectangular variable group ={T}. Simplex independent variable group ={x_1,x_2}. **a)** Uniform sampling in x_1-x_2 domain (black dots). Final samples in the hybrid T-X_1-X_2 domain (red dots). **b)** Extreme and intermediate points in x_1-x_2 domain and T domain (purple dots). Extreme and intermediate points in the hybrid T-X_1-X_2 domain (red stars).

Once the model domain samples are available, the corresponding model range samples can be generated. As previously mentioned, every domain sample point is used to close the degrees of freedom of a detailed simulation model. The solution of these simulation cases provides the model range sampling. This involves the solution of relatively complex equations systems including material and energy balances as well as equipment sizing and costing. Finally, with samplings for the model domain and rage, the data fitting required to build the surrogate mappings can begin.

6.2 Surrogate mapping data fitting

As mentioned above, there are different fitting algorithms for different non-linear mappings. However, all of them find the values of the surrogate adjustable parameters which minimize a metric of the surrogate deviations with respect to the available samples. For artificial neural networks, we use an iterative training procedure based on the basic back propagation algorithm complemented with some other elements. In particular, our final implementation includes data scaling and Principal Component Analysis, as well as the use of Bayesian regularization and early stopping to improve the network extrapolation capabilities.

The data scaling and a Principal Component Analysis (PCA) on the domain samples are special kinds of linear transformations which orthogonalize the components of the sample points. In some cases, the PCA can reduce the dimension the domain if the sampling indicates the presence of redundant or correlated components. As a result, the training algorithm is able to process the transformed samples more efficiently, leading to better surrogate mappings [70].

In general, the training involves minimizing the sum of squared model deviations with respect to a subset of sample points called the "training set". In any training with regularization, the objective function is a linear combination of two terms: the traditional sum of squared deviations and the mean of squared network parameters $w_{i,j}^{k}$ and b_{j}^{k}. The coefficients of these two terms are $\gamma \in (0,1)$ and $(1-\gamma)$ respectively. This objective function guides the search towards solutions with small network parameter values which are characteristic of smooth fitting surfaces. Bayesian regularization is a statistical framework that allows the automatic determination of the so called performance ratio γ, a parameter which balances of the two terms in the objective function.

Early stopping is a training termination criterion which involves the monitoring of model deviations with respect to a second subset of sample points known as the "validation set". Both regularization and early stopping reduce over-fitting, a condition in which the trained network features very small deviations with respect to the training set, but big deviations with respect to the validation set, indicating a poor network capacity to reproduce data outside of the training set (i.e. poor interpolation- extrapolation capabilities). This is particularly important when dealing with noisy data and small data sets.

Another important aspect in ANN training is the initialization algorithm for parameters $w_{i,j}^{k}$ and b_{j}^{k}. The Nguyen-Widrow initialization procedure [71] is used in this work. This algorithm provides initial parameter values for a layer, so that the active region of each neuron is distributed evenly through the layer domain. To explain this idea, consider a neuron i' in layer k' of an ANN. Following (5-10) this neuron maps the layer domain into component i' of the layer range according to:

$$x_{i'}^{k} = \mathrm{f}^{k}\left(W_{i'.}^{k'} \cdot x^{k'-1} + b_{i*}^{k'}\right) \qquad (6\text{-}6)$$

The active region for this neuron is the portion of the layer domain for which points $x^{k'-1}$ are mapped into non extreme values $x_{i'}^{k}$ for the function f^{k}. For linear transfer functions, the active region is not bound, but it is well defined for sigmoid transfer functions. For $\tanh(.)$ in particular, the active region is defined by the values of $x^{k'-1}$ for which the function argument $\left(W_{i'.}^{k'} \cdot x^{k'-1} + b_{i*}^{k'}\right)$ is in $[-2,2]$. It is worth noting that $\tanh(2) = -\tanh(-2) = 0.964$. In other words, at the boundary of the active region, the neuron provides an absolute value for $x_{i'}^{k}$ which is 96.4% of the extremum. Outside the interval $[-2,2]$ the argument is considered to drive the transfer function to saturation values, far away from the mid section where the neuron can play an active role in the fitting process.

This is why the Nguyen-Widrow initialization procedure, increases the fraction of neurons actively used in the fitting of data (i.e. non-saturated neurons), which in turn reduces the training time and improves the prediction capabilities of the network for a fixed number of neurons [71].

6.3 Surrogate mapping validation

After the data fitting, one performance parameter and two kinds of graphs are used to assess the suitability of the generated surrogate mappings. The performance parameter used here is the maximum surrogate deviation in the prediction of the values in the training set, expressed as a percentage of the model range. That is, let y_i^{Max}, y_i^{Min} be the maximum and minimum values for dependent variable y_i within the training data set. Then, the surrogate deviation for y_i at the training sample point j is given by:

$$dev_{i,j} = 100 * \left| y_{i,j}^{Surrogate} - y_{i,j}^{Data} \right| / \left(y_i^{Max} - y_i^{Min} \right) \quad (6\text{-}7)$$

This deviation metric compares the absolute surrogate deviation with the variability of y_i instead of its absolute value, resulting in a more balanced assessment for dependent variables which can take a wide range of values including very small ones. In these cases, a traditional metric would report very large relative deviations as a result of small absolute deviations around data points for which y_i takes very small values. As explained in section 7.1.1 the generation of a final surrogate mapping involves data fitting to a number of different ANNs. Among them, the one featuring the lowest maximum deviation is the one incorporated in the corresponding hybrid surrogate.

As for the final surrogate assessment, post-regression analysis graphs and validation data-fitting graphs are used. The post regression analysis graphs present $y_j^{Surrogate}$ Vs. y_j^{Data} for every

dependent variable y_i using the training data. Validation data-fitting graphs are 3D graphs overlapping surrogate predicted data $\left(y_j^{Surrogate}, x_k, x_l\right)$ and additional process data $\left(y_j^{ValData}, x_k, x_l\right)$ where x_k, x_l are two of the surrogate mapping independent variables. In this case, the surrogate and validation data are generated by sampling value of x_k, x_l between their upper and lower bounds, while keeping $x_m / \forall m \neq k, m \neq l$ fixed to a typical values x_m^*. To illustrate the idea, Figure 33 shows the three validation sampling planes used in a 3D rectangular domain.

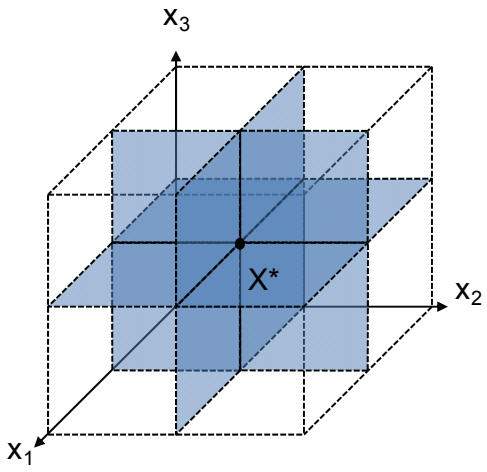


Figure 33: Validation sampling regions for a mapping with a 3D domain.

6.4 Surrogate mapping generation: Illustrative example

Consider the generation of a surrogate model to estimate thermodynamic port properties for mixtures of components H_2, CH_4, C_6H_6, C_7H_8, $C_{12}H_{10}$. For a flow based formulation, the surrogate model have the form in (4-10), including 7 independent variables $\left\lfloor F_{c,p} \right\rfloor_{c \in \mathbf{C}}, T_p, P_p$ and 2 dependent variables H_p, S_p. The surrogate domain is bound by the values presented in Table 3.

Table 3: Bounds and typical values for the domain of surrogate mappings H_p and S_p.

Independent variables		Units	Lower Bound (x^{Up})	Upper Bound (x^{Lo})	Typical values (x^*)	
					Case 1	Case 2
Flows	H_2	kmol/h	0	3905.92	1952.96	0
	CH_4	kmol/h	0	1085.18	542.59	0
	C_6H_6	kmol/h	0	255.06	127.53	127.53
	C_7H_8	kmol/h	0	391.06	195.53	0
	$C_{12}H_{10}$	kmol/h	0	10.22	5.11	5.11
Temperature		K	244.036	974.4	609.218	609.218
Pressure		kPa	90	3850	1970	1970

A sampling containing 4365 points was generated and used in the training. The data was obtained using ASPEN-Plus with the Peng-Robinson equation of state. An automatic search of the best network configuration and parameters was conducted here, considering ANNs with 1 and 2 hidden layers, 5 to 15 neurons per layer, and 3 different transfer functions for the hidden layers: $\tanh(v) = 2/(1+\exp(-2\cdot v))-1$, $\text{logsig}(v) = 1/(1+\exp(-v))$ and $\text{radbas}(v) = \exp(-v^2)$. The training was repeated 20 times for each network configuration. Notice the domain is relatively wide, since, as suggested in chapter 4, the resulting thermodynamic mapping is to be used in all ports and streams of a possible superstructure. The best training instance produced a maximum deviation of 2.9%, corresponding to a network with 2 hidden layers, 13 neurons in each hidden layers, and $\tanh(\cdot)$ as the transfer function. Post-regression graphs for this network are presented in Figure 34.

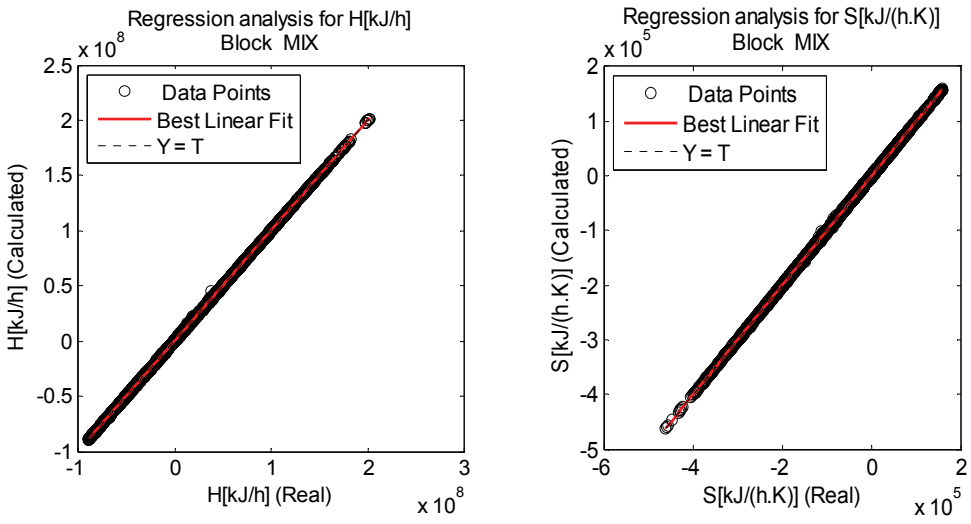


Figure 34: Post-regression graphs for a surrogate mapping used in the calculation of thermodynamic properties of mixtures containing H_2, CH_4, C_6H_6, C_7H_8, $C_{12}H_{10}$.

Figure 35 to Figure 38 present a fraction of the 42 possible validation graphs for the surrogate, where the vector x^* of typical values corresponds to "case1" in Table 3. In this case, x^* is a point in the middle of the domain, so the surrogate is supposed to perform well.

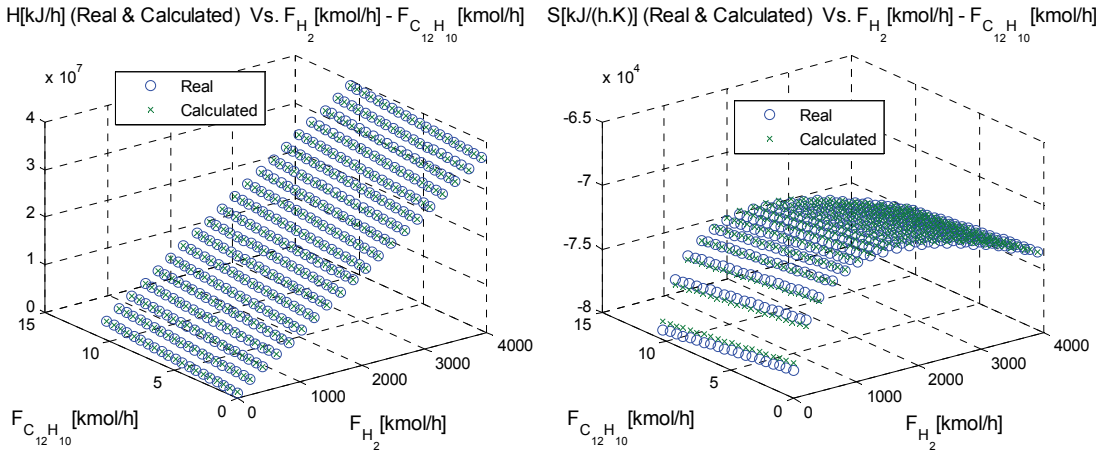


Figure 35: Surrogate validation graphs for plane F_{C12H10} - F_{H2} (Case 1).

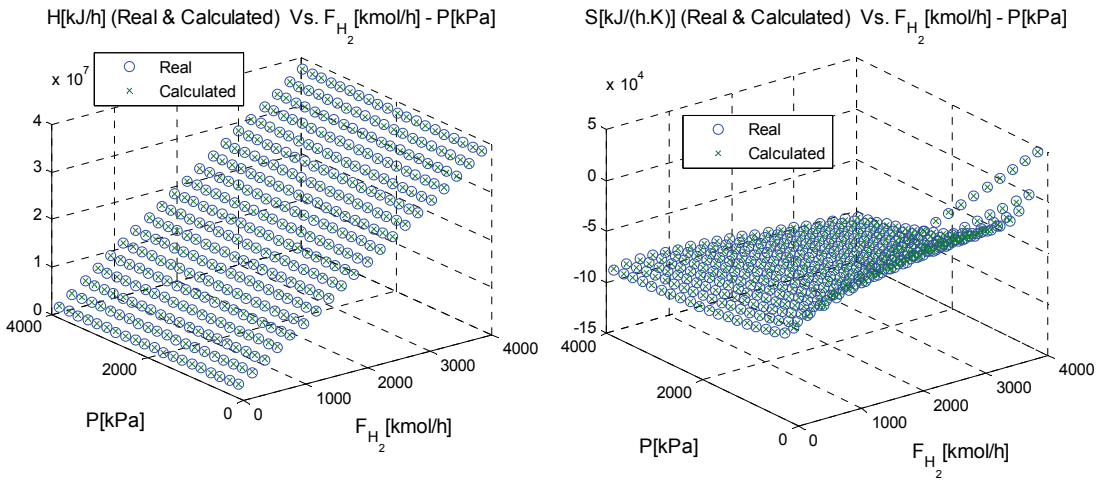


Figure 36: Surrogate validation graphs for plane P - F_{H2} (Case 1).

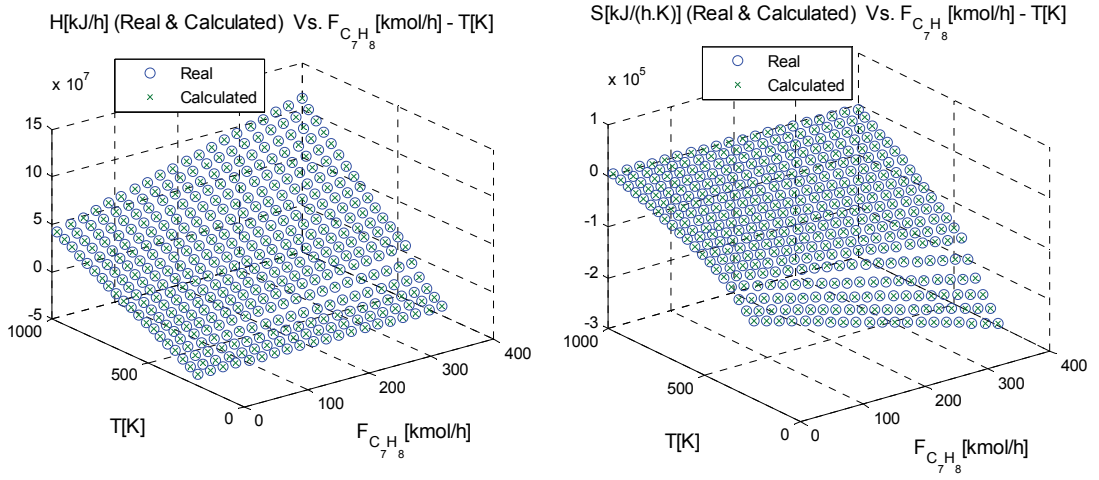


Figure 37: Surrogate validation graphs for plane T - F_{C7H8} (Case 1).

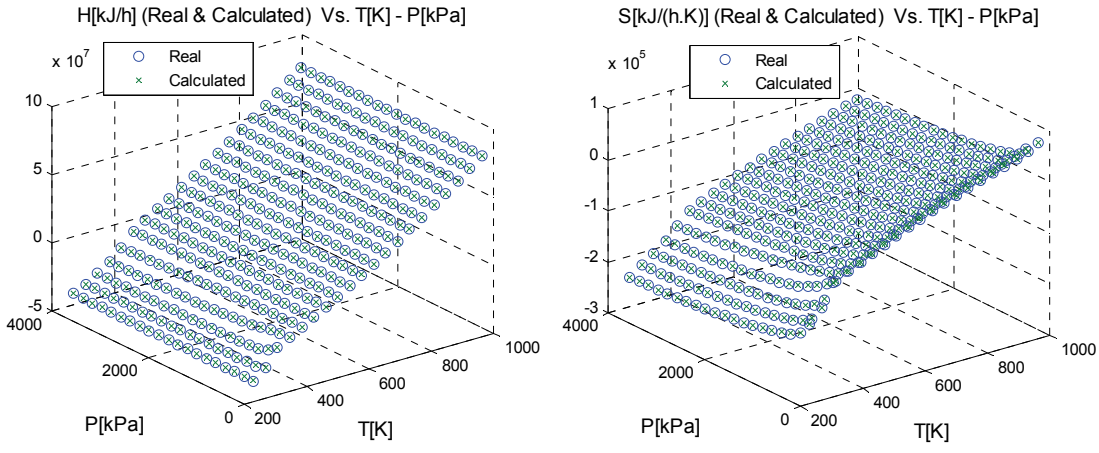


Figure 38: Surrogate validation graphs for plane P -T (Case 1).

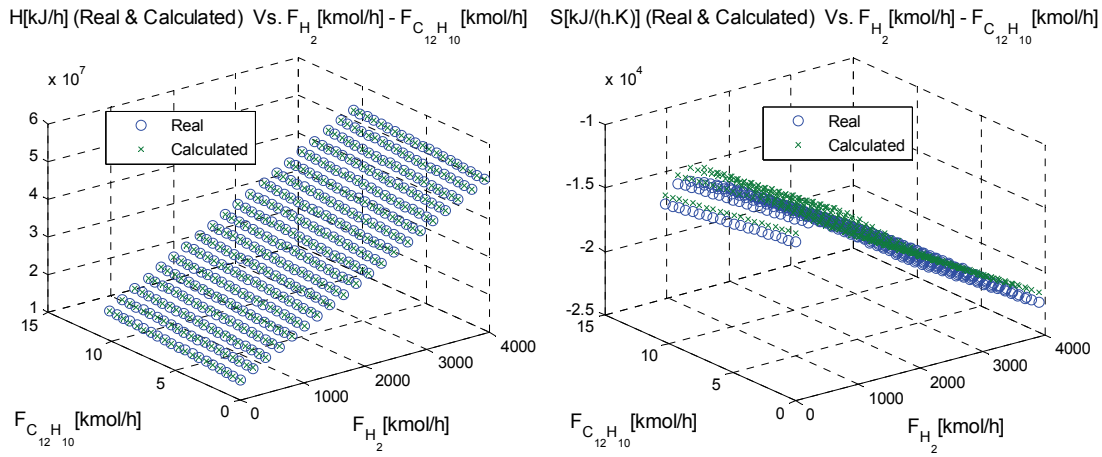


Figure 39: Surrogate validation graphs for plane F_{C12H10} - F_{H2} (Case 2).

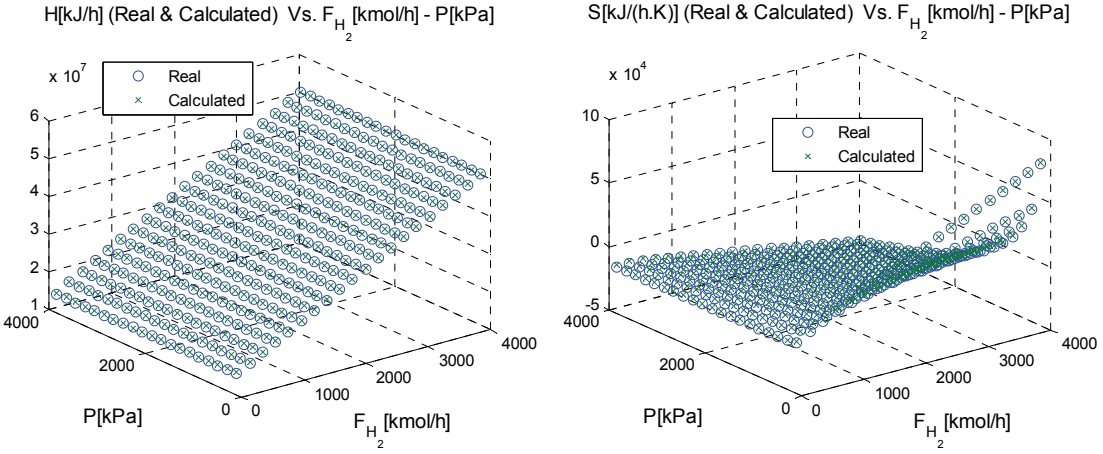


Figure 40: Surrogate validation graphs for plane P - F_{H2} (Case 2).

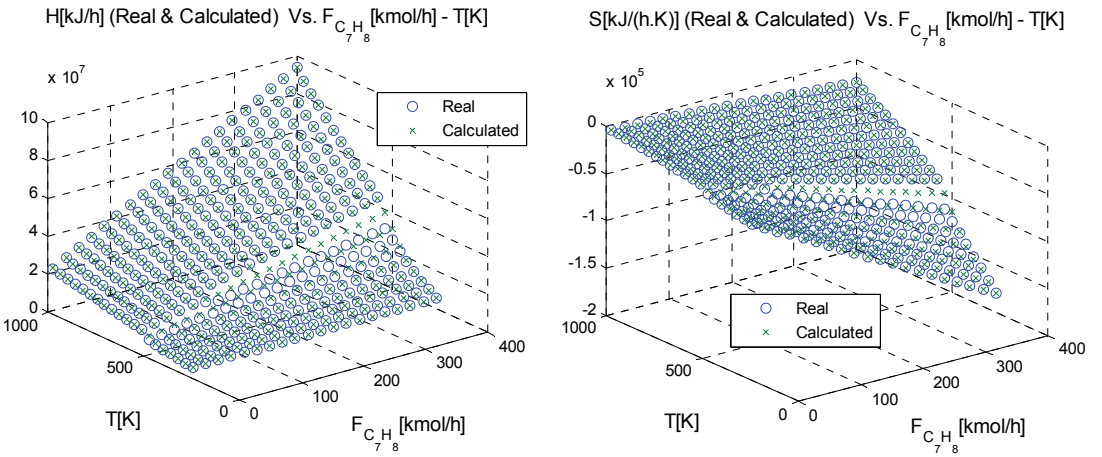


Figure 41: Surrogate validation graphs for plane T - F_{C7H8} (Case 2).

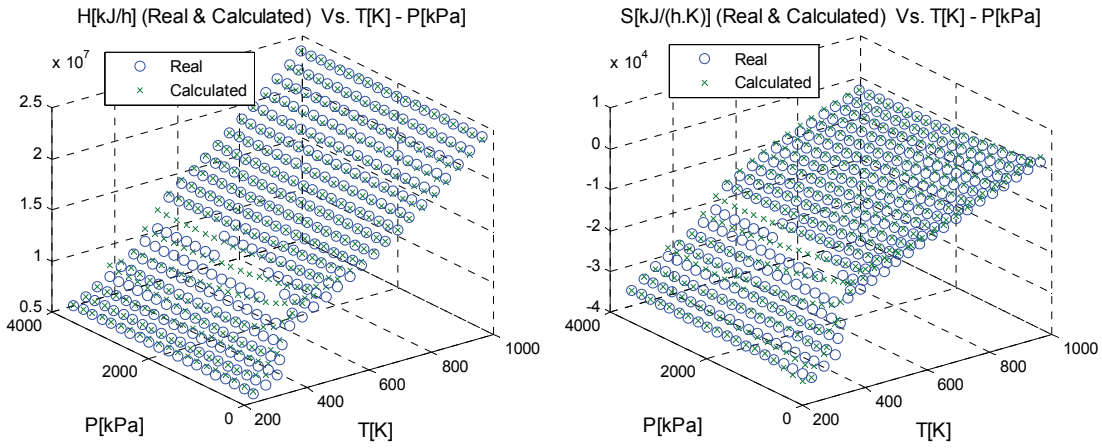


Figure 42: Surrogate validation graphs for plane P - T (Case 2).

Figure 39 to Figure 42 show the same validation graphs previously presented, but the vector $x*$ of typical values corresponds to "Case2" in Table 3. Here, $x*$ is a point at the boundary of the domain, as the flows of H_2, CH_4, and C_7H_8 take their corresponding lower bounds. Here, we see a small increase in the deviations mostly due to the presence of non smooth behavior (see Figure 41 and Figure 42) cause by phase transitions.

Chapter 7

FINAL REMARKS

There are several ways to implement and use the ideas presented in this work. This chapter includes additional information on the particular software implementation we developed to generate the results presented in chapter 8, as well as some extensions on how to use the methodology.

7.1 Software implementation

To implement our methodology we used tree different pieces of software: a process simulator to generate the realistic process data required to build the surrogate mappings, a scientific computing environment to process the data and generate the mappings, and finally, a modeling environment to formulate and solve the SO models. This section includes a description of the system of files and subroutines used and the role each one of them plays.

7.1.1 Surrogate mapping generation

Surrogate generation requires the integration of a proper scientific computing environment and a process simulator. For this work we have used ASPEN PLUS to generate process data and MATLAB

V7.12 to process the data; however, similar implementations can be developed for other languages (e.g. Octave, FORTRAN, C, etc.) and process simulators (e.g. HYSYS, PROII, CHEMCAD, etc.)

Details of our implementation and the way all the pieces work together are presented in Figure 43. The generation of surrogate mappings involves a main subroutine (MappingGenerator.m) which is supported by a data generation subroutine (SDG.m) and a mapping fitting subroutine (NNMG.m). Here, "SDG.m" takes information from a parameter file (MappingParameters.mat) and produces a sampling for the mapping domain and range (MappingTrainingData.mat). This subroutine is able to handle domains with rectangular variables and several groups of simplex variables. The domain sampling is generated using information for each independent variable, including their lower and upper bounds, the type of variable and the variable group they belong to. The domain sample matrix is parsed through the MATLAB unit subroutine (Unit.m) to the ASPEN-MATLAB interface (AspenBlock.m). The interface "AspenBlock.m" manages the information flow between the MATLAB code and the ASPEN Plus calculation engine. In a typical run, "AspenBlock.m" retrieves the proper standard ASPEN unit simulation input file (e.g. flash.inp, reactor.inp, etc.), rewrites a "sensitivity block paragraph" to specify simulation cases using the domain sample matrix, calls the ASPEN Plus calculation engine to run the modified ASPEN file, and retrieves material and energy balance information from the corresponding output file (e.g. flash.out, reactor.out, etc.) The simulation information is then parsed to sizing and costing subroutines (UnitSize.m, UnitCapCost.m, UnitOpCost.m) developed based on standard algorithms [72],[73],[74],[75]. Finally the mapping range sampling matrix is assembled from material and energy balance information complemented with cost information.

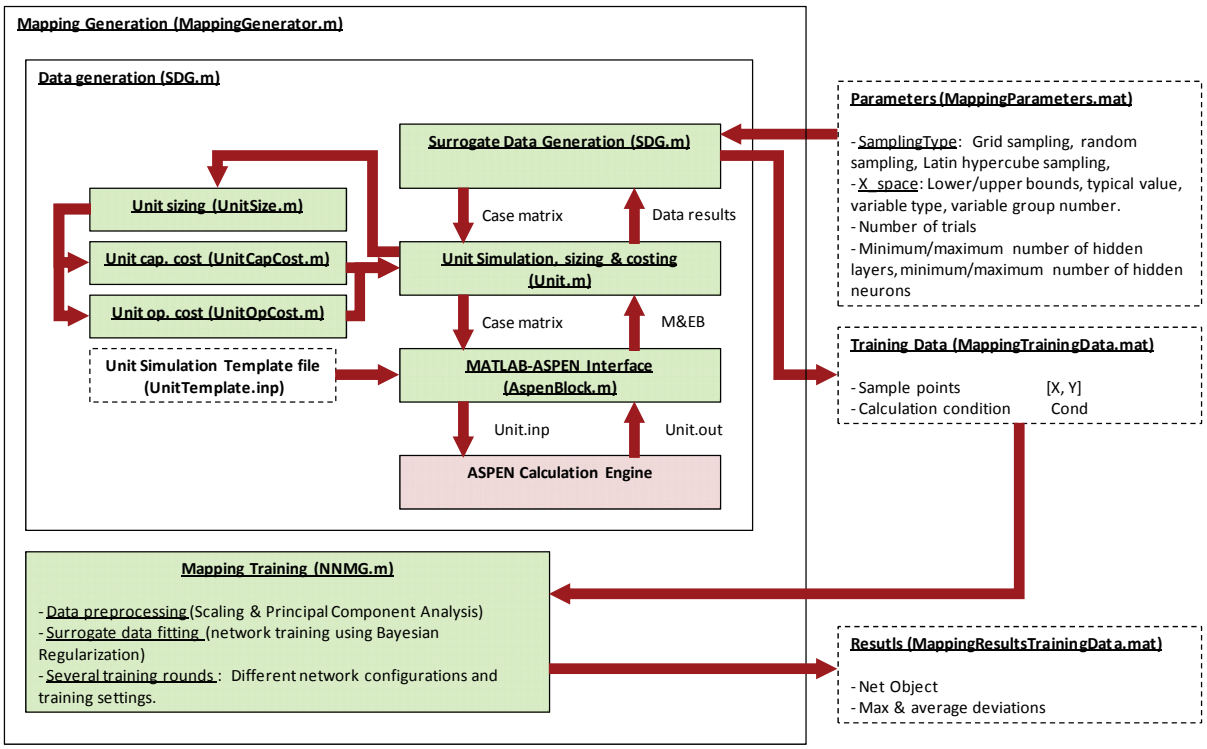


Figure 43: Data generation and surrogate mapping fitting.

Once the sampling matrices are generated, they are sent to the training subroutine (NNMG.m), where data preprocessing using scaling and Principal Component Analysis (PCA) is performed before the main network training starts. Subroutine "NNMG.m" is very flexible, allowing the use of different training algorithms and options. From a number of training cases we found that Bayesian regularization with early stopping outperforms all other training algorithms implemented in the MATLAB Neural Networks Toolbox, being the preferred algorithm when running "NNMG.m".

In order to find a suitable network architecture for a particular surrogate mapping, "NNMG.m" performs the parallel training of a number of networks with different configurations. The configurations are generated from user supplied parameters (i.e. max/min number of hidden layers, max/min number of neurons per layer, list o transfer functions). To account in part for the stochastic

nature of the fitting process (since the training set is pulled randomly from the available samples), the training for each network is repeated a number of times.

7.1.2 Superstructure modeling

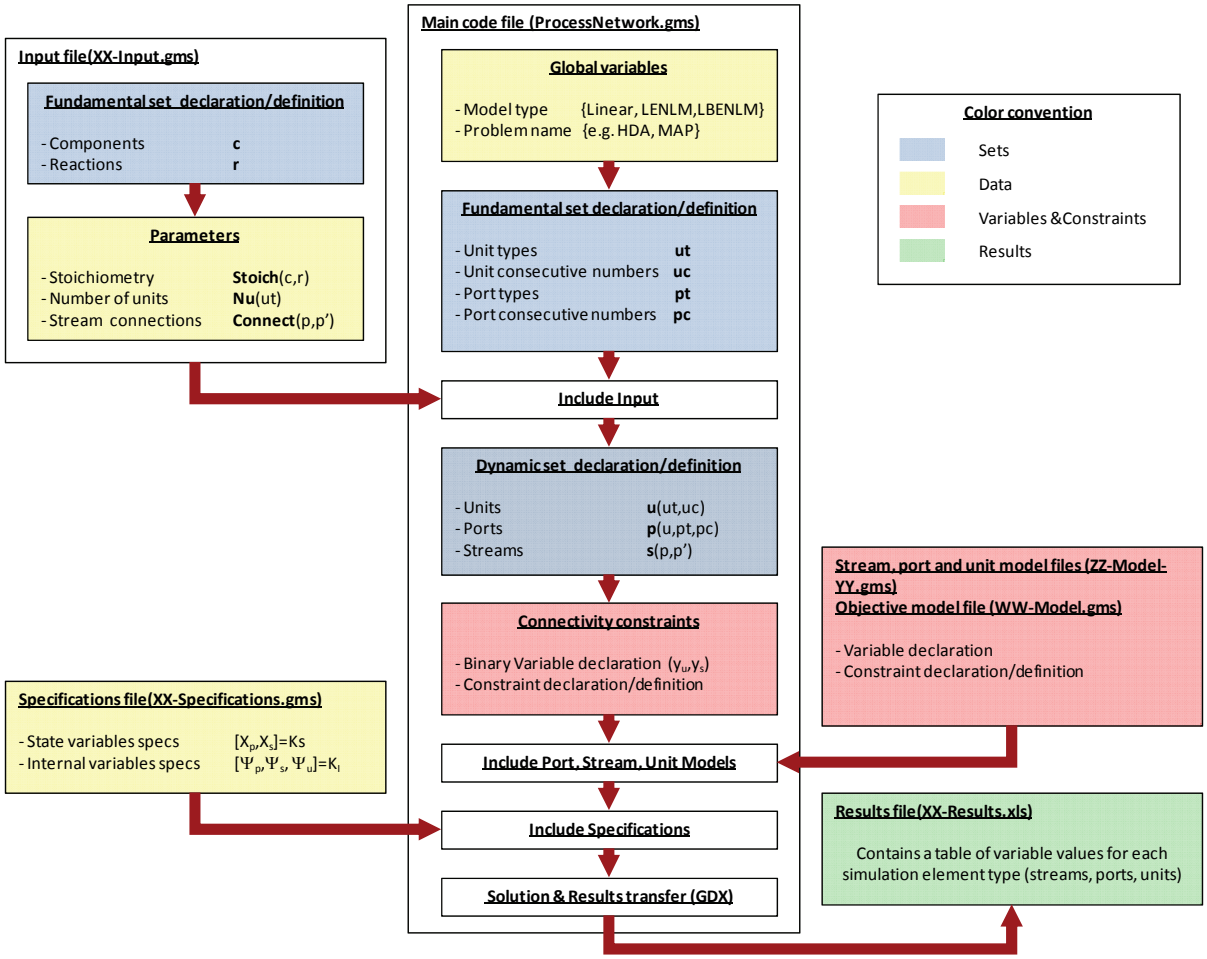


Figure 44: GAMS implementation of a general superstructure model.

The final SO model implementation requires the use of a proper modeling environment. Although the description presented here assumes the use of GAMS (General Algebraic Modeling System) [76], the same models can be easily implemented in any other modeling language having analogous capabilities. Figure 44 presents the general structure of the main model file (ProcessNetwork.gms)

as well as the required supporting files containing model data (XX-Input.gms, XX-Specifications.gms) and element models (ZZ-Model-YY.gms, WW_Model.gms) which are combined with the main model at compilation time.

The main model file is divided in eight sections. Section 1 includes the declaration and assignment of several important control variables specifying the data and model files to use. These control variables include the "problem name" and the "model type". The problem name is a string "XX" used to identify files containing problem specific information. The string is incorporated in the files names "XX-Input.gms" and "XX-Specifications.gms". The model type is a string "YY" used to distinguish between different model versions for the same element "ZZ". All element models are named as "ZZ-Model-YY.gms". Valid strings for model type include: "Linear" for entirely linear models, "LENLM" for models containing only linear equations and non-linear mappings, "LBENLM" for modes including linear and bilinear equations plus non-linear mappings.

Section 2 contains the declaration and definition of fundamental sets such as unit types $\mathbf{UT}$, unit consecutive numbers $\mathbf{UC}$, port types $\mathbf{PT}$, and port consecutive numbers $\mathbf{PC}$; as well as the declaration of all the multidimensional sets and subsets presented in section 4.1. It also includes the declaration and assignment of an important parameter, namely the matrix $Np(ut, pt)$ indicating the number of inlet and outlet ports in each unit type.

Section 3 incorporates the content of an appropriate input file XX-Input.gms. This file declares problem specific sets for components $\mathbf{C}$ and reactions $\mathbf{R}$. It also declares and assigns problem specific parameters such as $Stoich(c,r)$ which contains the problem stoichiometric coefficients $\nu_{c,r}$, the vector $Nu(ut)$ indicating the number of units per each unit type included in the superstructure,

and the connectivity binary matrix $Connect(p', p)$ indicating the pairs of ports connected by streams.

In section 4, sets $\mathbf{U,P,S}$ are automatically calculated trough conditional statements using the basic sets and the parameters defined so far. For example, $\mathbf{U}$ is calculated using parameter $Nu(ut)$, $\mathbf{P}$ is calculated from $\mathbf{U}$ and parameter $Np(ut, pt)$, and $\mathbf{S}$ is calculated from $\mathbf{P}$ and the connectivity matrix $Connect(p, p')$. Using the same approach, the rest of the subsets and multidimensional sets are calculated automatically. It is worthwhile to notice the small number of parameters required to perform these calculations, and how easy it is to modify extensively a particular model implementation in a short time. This is a consequence of the key modeling ideas used and previously discussed in section 4.3.

Following the set calculations, section 5 contains unit and stream binaries declarations, as well as the connectivity constraints presented in section 4.5.2. In section 6, appropriate stream, port and unit model files "ZZ-Model-YY.gms" are included. These files declare all variables and constraints presented in section 4.4 for each modeling element, all of which are indexed using the sets calculated in the previous sections. With this indexing strategy, only the required variables and constraints are generated based entirely on the few input parameters described earlier.

Section 7 incorporates the content of a proper specification file, where the values of certain variables are fixed (e.g. raw material characteristics, final product requirements, etc.) The final section includes the solution statement, and other commands to gather and present the numerical results in a proper output file (i.e. spreadsheet files XX-Results.xls)

7.2 Automatic surrogate domain update

Once the surrogate mappings are generated, their validity is limited to the sampled domains. In fact, as presented in section 4.5 and appendix C, domain bounds are included in the MINLP reformulation of every superstructure element model to ensure that every surrogate mapping is used correctly. However, the use of this many bounds can be tricky if not done properly, since they can limit greatly the search space and even lead to infeasible models.

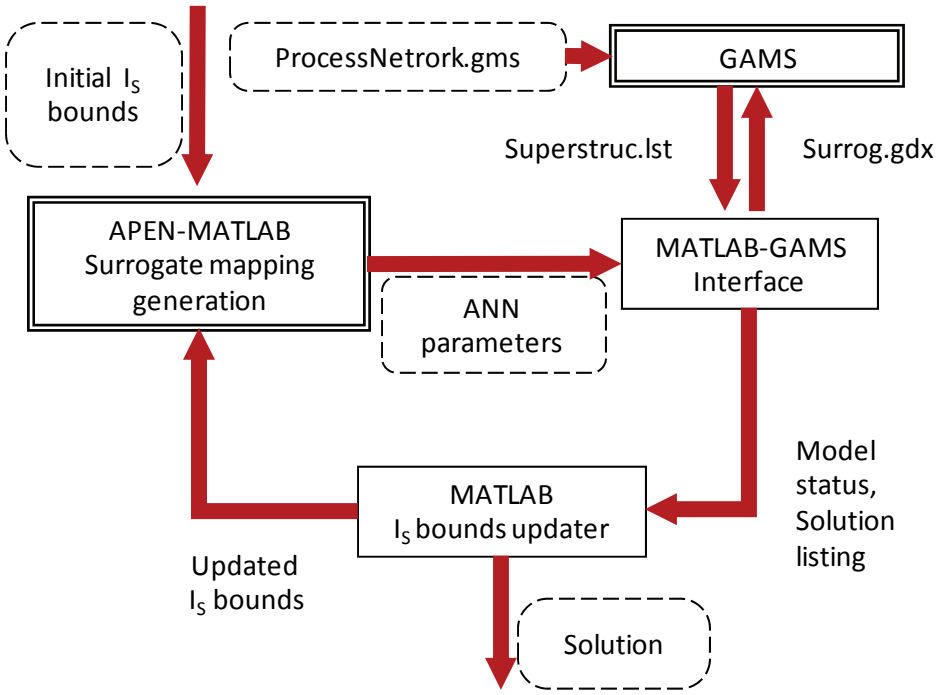


Figure 45: Automatic update of surrogate mapping domains.

To illustrate the point, consider a case where the inlet port domain does not overlap with the corresponding unit domain. Here, there is no region in the state space for which both the port model and the unit model are valid, thus making the SO model infeasible. Another interesting situation comes when the optimization algorithm finds an optimal solution for which one or several

of the mapping bounds are active. Here, there is doubt about the value of the solution, since it could be the result of an overly restrictive mapping domain. To avoid these situations some surrogate mapping domain updating procedure is required. A possible implementation of this, including a MATLAB-GAMS interface (see. MATLAB and GAMS: Interfacing Optimization and Visualization Software by: Michael C. Ferris. http://www.cs.wisc.edu/math-prog/matlab.html) is depicted in Figure 45.

In every execution of the external loop, the domain bounds are used to define the sampling region that ultimately lead to the calculation of the surrogate mapping adjustable parameters. At this point a MATLAB-GAMS interface calls GAMS to solve the superstructure MINLP model (i.e. ProcessNetwork.gms). The solution information generated by GAMS includes the solution status that indicates if the problem is feasible or not, as well as details about active-inactive, feasible-infeasible constraints. This information is used by the domain updater to either update the domain in order to start a new iteration or stop the calculations. The proposed scheme is very useful in situations where the production of accurate surrogates requires small mapping domains.

In general, guaranteeing that this procedure will converge to the optimal solution of the original model is a difficult goal. To this respect, the literature presents a theoretical discussion of the necessary and sufficient conditions for such convergence in the context of optimization using approximated models [77]; however, checking these conditions is rather hard even for simple cases. An alternative approach has been proposed by Caballero[52]. In this work, the surrogate domain updating procedure enters a contraction step every time the solver returns a feasible solution strictly interior to the surrogate domains. The idea of such domain contraction is to increase the local accuracy of the surrogate, which increases the chances of identifying a solution to the original detailed optimization problem.

7.3 Plant subsystem surrogates

The use of surrogate models at the process unit level is intuitive because unit models are the elemental components used in any process modeling effort, and the building blocks of commercial simulators. However, surrogates can also be used to describe the behavior of entire plant subsystems composed by several processing units. This is useful when the structure of such subsystems is not to be optimized. The obvious advantage is the reduction in the total number surrogate connecting variables (e.g. state variables of the streams connecting the units) and their associated surrogate mappings. In a way, we present this idea in appendix D, where absorption columns are modeled as a single unit instead of a group of interconnected flash vessels.

To illustrate the idea, consider the amine-based CO_2 capture system presented in Figure 46. Assuming 5 components (O_2, N_2, CO, H_2O, MEA) and using the standard surrogate models included in appendix D, the total number of surrogate independent and dependent variables to model the systems as presented in Figure 46 a) approaches 100. However, considering a single surrogate for the entire system as presented in Figure 46 b) brings this number close to 30. This reduction is the result of some streams becoming internal to the system, and the reduction in the total number of degrees of freedom (and hence the effective number of independent variables) due to unit coupling.

The construction of a surrogate for such *plant subsystem* requires the implementation of a simulation model where some of the original unit degrees of freedom are related to each other; e.g., the pressure increase of the pumps are related to column operating pressures and the pressure drops of the units in the streams connecting the columns. That is, these variables are no longer independent. In addition, some other variables are replaced by new variables introduced to better describe the impact of the operational conditions on the system performance. For example, as

suggested in the literature [78], the total solvent flow rate and the CO_2 degree of separation in the regenerator can be replaced by the CO_2 recovery in the absorber and the CO_2 concentration in the lean solvent.

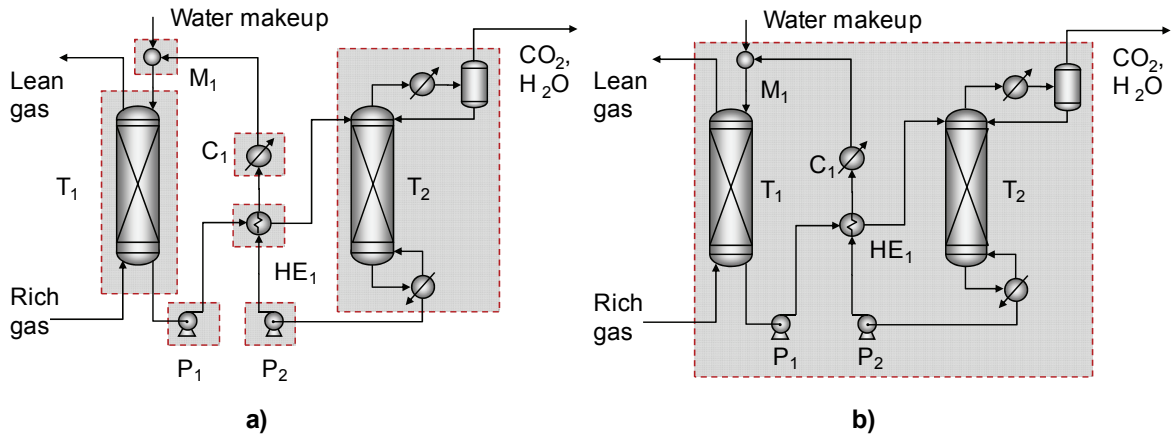


Figure 46: Amine based CO_2 capture unit. **a)** Use of surrogate models at the unit operation level (|IS|+|DS|=99). **b)** Use of a surrogate for the whole system (|IS|+|DS|=33).

7.4 Integration of multiplatform models

Another interesting aspect of the proposed surrogate framework is that it enables the formulation of optimization models that combine unit models developed using different tools. This is possible since models developed in different commercial simulators, programming languages and modeling environments can be used as black boxes to generate data that is later used to fit surrogate mappings. In addition, process data can also come from experiments (e.g. chemical reaction experiments which can be used to generate a reaction rate surrogate mapping). This capability, makes the surrogate modeling framework suitable to address the optimization of processes based on new technologies or, in general, unit operations that cannot be readily modeled using commercial tools.

As an example, consider the production of methanol from water and CO_2 represented in Figure 47. Here, CO_2 is split into CO and O_2 using a thermo-chemical splitter via concentrated solar power. The resulting CO is mixed with water in a Water Gas Shift (WGS) reactor to produce a mixture of CO, CO_2 and H_2 that is sent to a Methanol Synthesis (MS) reactor to generate the product. Scientists from Sandia National Labs are currently exploring this pathway to methanol using a recently developed thermo-chemical splitter, called the CR5 reactor [79],[80]. Clearly, to optimize the process as a whole, we need to account simultaneously for the CR5 splitter and all remaining units. While the WGS and MS reactors and separation units can be modeled using any commercial process simulator, this is not possible for the CR5. This reactor features a configuration where multiple parallel discs supporting a catalytic material rotate inside the devise, passing from and oxidation section where CO_2 is stripped of one oxygen atom, to a reduction section where the catalyst is regenerated and O_2 is released.

However, a FORTRAN model was recently developed that accounts for the effects of ring rotation, reaction and diffusion in the active ring material in both the oxidation and reduction sections of the ring, and energy recuperation via heat transfer with a counter-rotating ring outside of the oxidation and reduction sections. A shrinking core particle model is used to account for the reaction and diffusion processes occurring on the scale of individual particles, and reacting particles are assumed to be embedded within an inert solid matrix through which reacting species and heat diffuse[79]. Using this FORTRAN model it is possible to generate the realistic process data required to develop a surrogate model for the CR5 (see Figure 48) which can be used along with surrogates for the remaining units (developed using ASPEN) for plant-wide optimization.

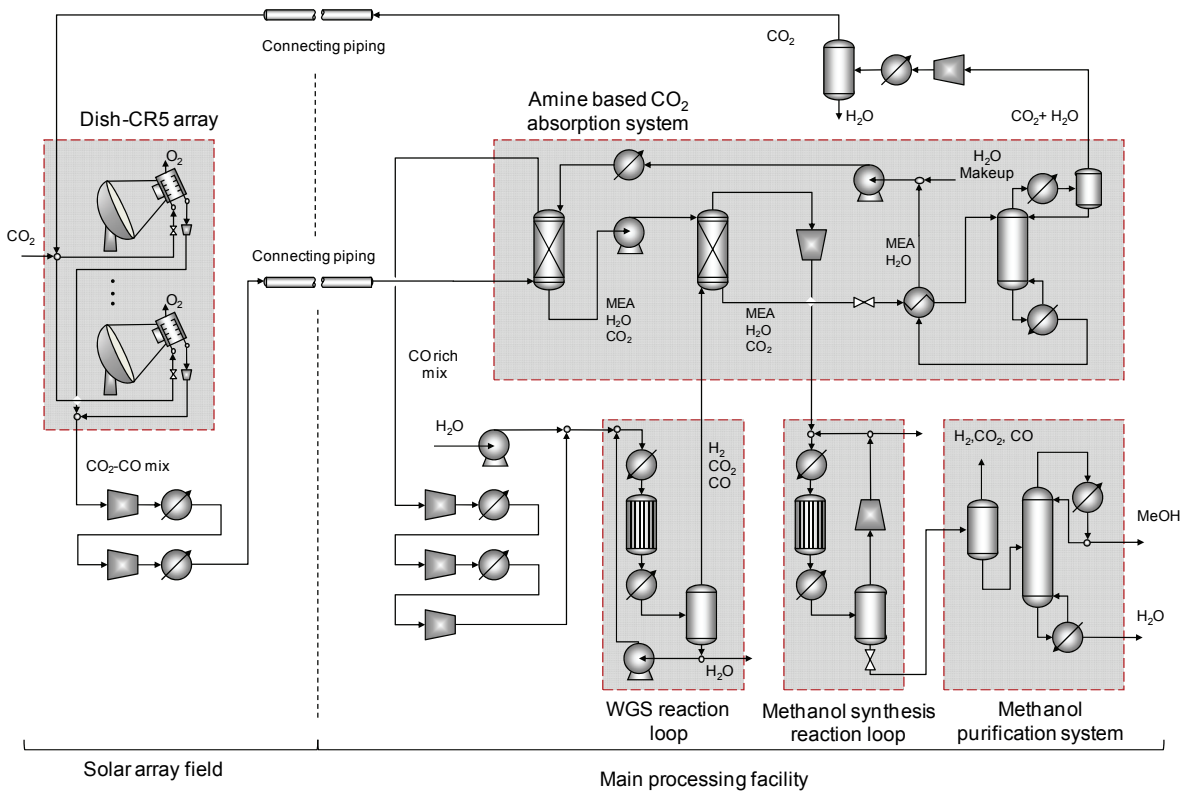


Figure 47: Production of methanol from CO_2 and water. The system couples a thermo-chemical splitter (Dish-CR5 array) and traditional processing subsystems (Amine based CO_2 absorption, water gas shift reaction loop, methanol synthesis loop, methanol purification.)

7.5 Outline of solution strategies

As mentioned in section 5.3, the use of surrogate mappings results in SO models with a unique structure, where all the different complex nonlinearities commonly used in detailed unit operation models are replaced by the few nonlinearities present in the surrogate mappings. This is why the mathematical form of the surrogate mappings affects the tractability of the whole SO model. In particular, surrogate mappings should be selected to have a mathematical structure which could be

exploited by current numerical solvers or tailored solution algorithms. In our case, the use of ANNs was decided based on their simplicity, fitting capabilities, and the fact that that they bring a unique non-linearity to the models, namely the $\tanh(\cdot)$ function, which can be exploited as indicated next.

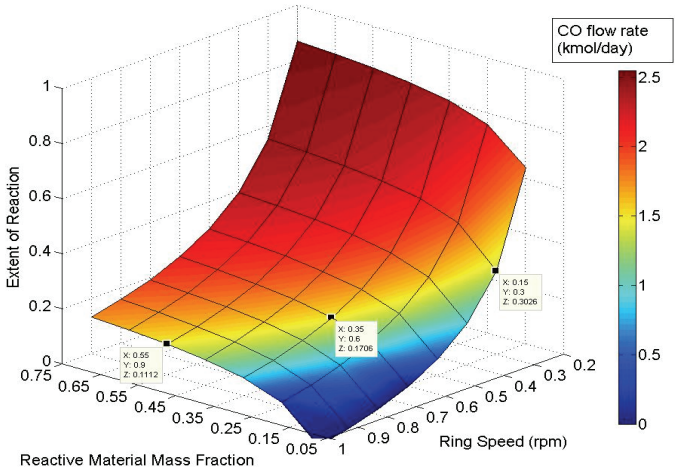


Figure 48: CO production and extent of reaction in the CR5 sub-system model as a function of rotation ring speed and fraction of reactive material.

Current modeling environments such as GAMS offer different numerical solvers with different capabilities. Depending on the accuracy requirements, the size and the type of problem, our surrogate approach can be used in different ways:

Large scale preliminary studies: In order to gain insights about a complex system, sometimes it is convenient to use simplified unit models based on key principles (e.g. conservation principles) while having a looser characterization of other aspects (e.g. reaction kinetics, thermodynamics, etc.) As an illustration, consider a high level study of competing process technologies in the design of an entire industrial complex where a number of chemical plants are interlinked. Here, it is important to capture the main economic tradeoffs of different subsystems. Hence, sharp separators models can

used instead of detailed separation models, and reactors with fixed conversions can be used Instead of detailed reactor models. To address these modeling requirements, the superstructure element models can be easily modified by replacing the nonlinear surrogate mappings with simple linear approximation. As a result, the extremely large SO model becomes a large MIP which could be solved to optimality using a state of the art numerical solve such as CPLEX.

Medium scale detailed studies: In those cases where accurate modeling is required to obtain significant results, but model complexity is an issue, our surrogate approach is recommended. Here the SO model is a nonconvex MINLP due to the presence of tanh(.) in the surrogate mappings. To find local solutions to these formulations, MINLP solvers such as DICOPT can be used. However, a tailored global optimization algorithm must be implemented to find the global solution. Such algorithms could be based on piecewise linear approximations or under/over estimators replacing function tanh(.) in the formulation. Similar ideas are reported in the literature to deal with the global optimization of non-convex models [65],[66].

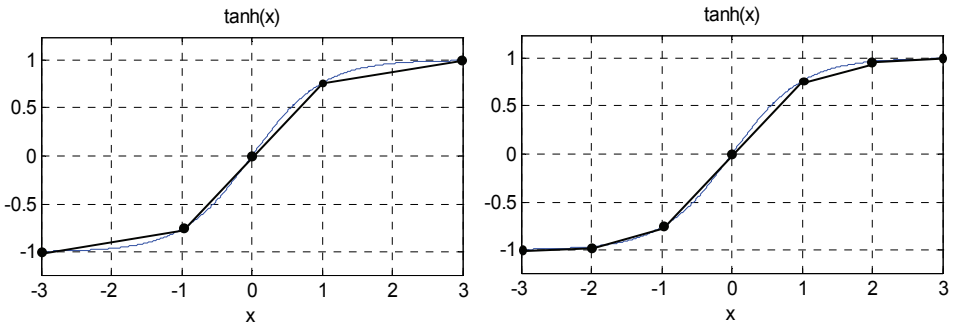


Figure 49: Updating piecewise linear approximation of tanh(.)

Replacing tanh(.) with a piecewise linear approximation transforms the SO model into an MIP which can be solved using CPLEX. Replacing tanh(.) with a set of over/under estimators transforms the SO model into a convex MINLP (an under estimator of the original problem) which can be solved to

optimality using DICOPT. Obviously, when the set o points defining the approximations is increased (see Figure 49 and Figure 50), the solution of the resulting model (MIP or MINLP) gets closer to the global solution of the original surrogate formulation. For an algorithm based on one of these replacements, the set of points defining the approximations is updated and the resulting model is solved until no updating is required.

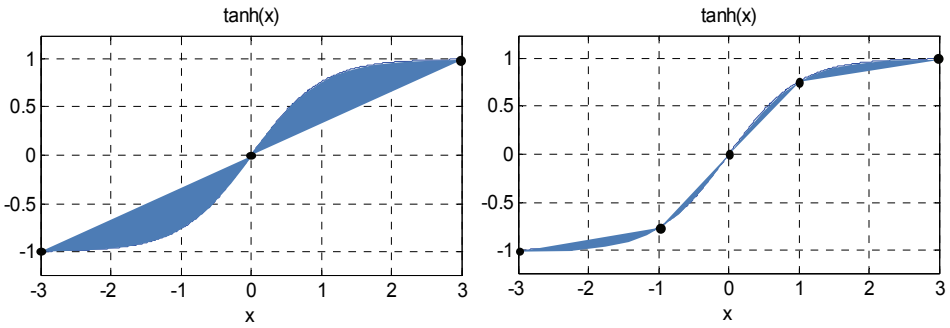


Figure 50: Updating over/under estimators of tanh(.)

Detailed studies of medium to small scale: For cases where accurate modeling is required, global solutions can be obtained using the general techniques implemented in state of the art global optimization solvers such as BARON. However, this is only possible for problems of small to medium size due to the limitations of such solver [81].

Parametric optimization: For cases where the focus is the optimization of process operating conditions, the element models included in section 4.4 and appendix B, D can be used instead of their MINLP versions. Depending on their size, the resulting non-convex NLPs can be solved using local solvers such as CONOPT, global solvers such as BARON or the tailored optimization algorithm previously mentioned, where tanh(.) is replaced with a piecewise linear approximation or proper over/under estimators.

Chapter 8

CASE STUDIES

8.1 Amine regeneration surrogate

Consider the amine based CO_2 capture systems presented in Figure 51.

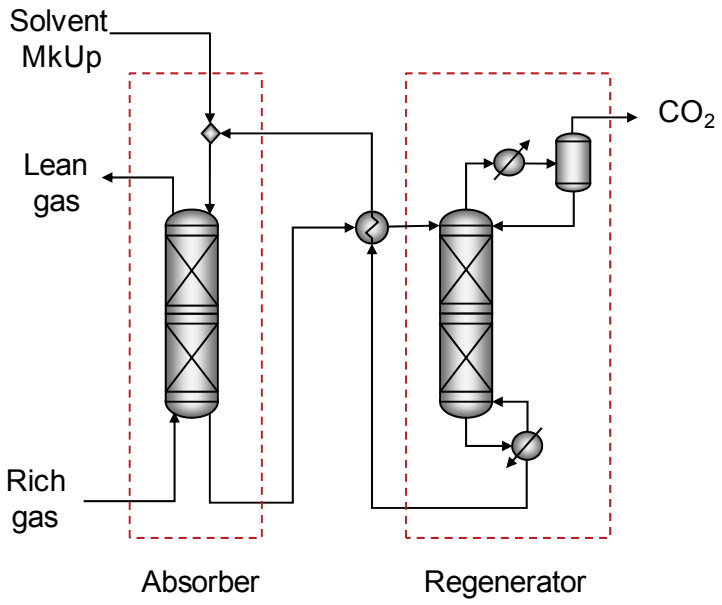


Figure 51: Simple amine based CO_2 capture system.

The system uses an absorber fed with a proper solvent (e.g. an aqueous solution of MonoEthanolAmine) in order to recover the CO_2 present in the "rich gas" coming from a stack. The used solvent is regenerated by stripping its CO_2 content in a distillation column. The lean solvent leaving the bottom of the regenerator is recycled back to the absorber. In order to obtain realistic and significant results in the optimization of this kind of systems, realistic models for both the absorber and the regenerator are required. In particular, it is important for these models to include a detailed characterization of the highly non-linear thermodynamics of CO_2-MEA-H_2O mixtures. Our surrogate approach is especially suitable in this case, where there is need for realistic modeling without all the burden of detailed, first principle, tray by tray column models (e.g. MESH models).

For the regenerator, model (D-47)-(D-56) included in section D.1.3 can be used. The number NC of components for the system is five, namely O_2, N_2, CO_2, H_2O and MEA. The column has one feed, one partial condenser with total reflux, one partial reboiler, and five equilibrium stages (N = 5). A detailed MESH model for a unit like this one has N(3NC+5)+2 equations (including the summation of mol fractions for the feed stream as well as the thermo equations to calculate K-values and stream molar enthalpies) and N(3NC+5)+(NC+4)+3 variables (including the K-values and molar enthalpies for every stage and every stream), leading to NC+5 degrees of freedom. For our regenerator, this implies a total of 102 equations and 112 variables, 10 of which are independent (e.g. 5 feed component flows, feed temperature, feed pressure, column pressure drop, condenser temperature and boil up ratio).

Simplified models are often employed to generate computationally tractable optimization problems for systems containing complex units such as separation columns. For example, the use of ideal thermodynamics brings Liquid-Vapor component distribution coefficients (i.e. K-values) which are independent of composition, and enthalpy calculations which do not account for mixing effects.

While these models can be useful in addressing simple *ideal* mixtures, their capabilities can be rather poor in more general cases like the one considered here. Our surrogate approach does not suffer from this limitation because regardless of the mixture thermodynamics complexity, the form of the surrogate and its size remains approximately the same. More importantly, surrogate models can be substantially less complex and more accurate than first principle unit models using simplified thermodynamics.

To illustrate this point, we compare a MESH model with ideal thermodynamics against a distillation surrogate model. Considering a fixed number of trays, the surrogate mappings in model (D-47)-(D-56) includes 9 independent variables (i.e. 5 feed component flows, feed temperature, column pressure, condenser temperature and boil up ratio) and 7 dependent variables (distillate flows, condenser duty and reboiler duty).

A surrogate mapping was built using the domain bounds presented in Table 4. The mapping is an ANN with 2 hidden layers including 15 sigmoid neurons per layer, and an output layer with 7 linear neurons. The samples where generated from a MESH model in ASPEN Plus using a thermodynamic method tailored for systems containing amines. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU 920 @ 2.67 GHz. The generation of the 1024 sample points took 140 seconds while the training of the network took 144 seconds. Figure 52 represents the post-regression analysis for two important dependent variables: reboiler duty and bottoms CO_2 flow. These two variables where selected since they are related to the column regeneration capabilities and its operational cost. As shown, the agreement between the detailed MESH model and the surrogate is very good, with a maximum deviation of about 2% of the variable range.

Table 4: Bounds and typical values for the domain of a MEA regeneration column.

Independent variables		Units	Lower Bound (x^{Lo})	Upper Bound (x^{Up})	Typical values (x^{*})
Flows	O_2	kmol/h	0.01	0.1	0.1
	N_2	kmol/h	0.01	0.1	0.1
	CO_2	kmol/h	5000	10000	7000
	H_2O	kmol/h	50000	100000	70000
	MEA	kmol/h	5000	15000	10000
Temperature		K	300	360	330
Pressure		kPa	150	200	170

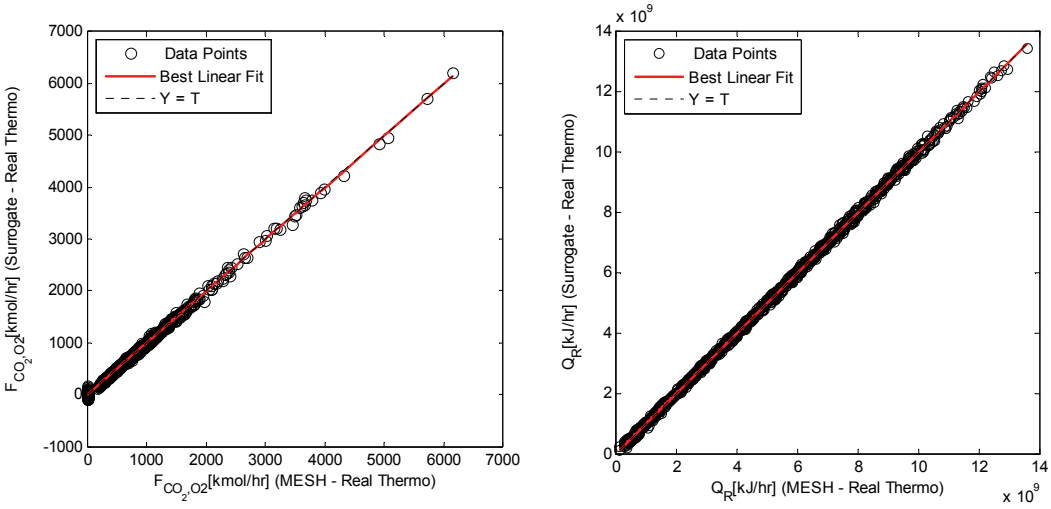


Figure 52: Post regression analysis for the CO_2 content in the regenerated solvent and the reboiler heating duty. This compares data and surrogate predictions.

A similar analysis is presented in Figure 53 where the values predicted by the detailed model and the simplified MESH model (i.e. using ideal thermodynamics) are compared. As shown, the simplified model greatly underestimates the CO_2 content in the regenerated solvent, although it does not show significant deviations in the case of the reboiler duty. This is expected since the major contribution of a non-ideal amine thermodynamic model is in the correct estimation of the CO_2-MEA phase distribution. The enthalpy calculation of aqueous amine solutions does not seem as challenging in cases like the one presented here, where water is abundant.

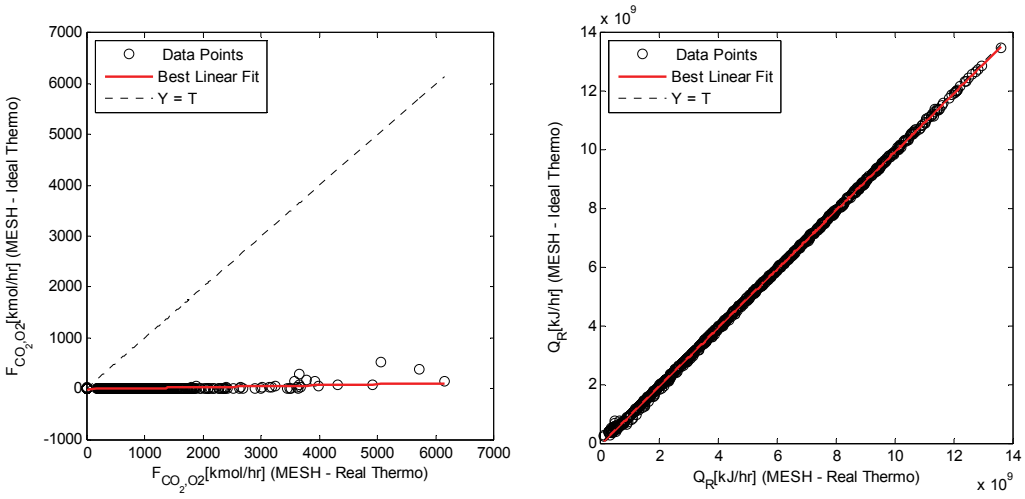


Figure 53: Comparison of values predicted using the simplified MESH model for the regenerator (i.e. using ideal thermodynamics), and the values predicted by the detailed model

8.2 Production of Maleic Anhydride: reactor optimization

Consider the production of Maleic Anhydride (MA) from Benzene (B) in a reactor fed with 966 kmol/h of air and 34 kmol/h of benzene at 300 K, 1013 kPa. Under this reaction conditions the predominant stoichiometry is as follows:

$$C_6H_6 + (9/2)\cdot O_2 \Rightarrow C_4H_2O_3 + 2\cdot CO_2 + 2\cdot H_2O$$
$$C_4H_2O_3 + 3\cdot O_2 \Rightarrow 4\cdot CO_2 + H_2O$$
$$C_6H_6 + (15/2)\cdot O_2 \Rightarrow 6\cdot CO_2 + 3\cdot H_2O$$

Assuming no pressure drop, the goal is to find the operating temperature and reaction volume maximizing the annualized profit. Reaction kinetics is reported in the literature[82]. Annualized profit includes revenue from Maleic Anhydride at current price (www.icis.com), capital cost [72], and utility costs.

The underlying optimization problem has the same structure (5-11)-(5-13) presented in section 5.5, and according to the analysis presented there, a detailed reactor model (5-14)-(5-20) can be replaced by a mapping relating the reactor volume and operating temperature to the flow of main product leaving the reactor and the required heating. (i.e. $\mathbf{I}_S = \{V_R, T_O\}, \mathbf{D_S} = \{F_{c*,O}, Q_R\}$).

Considering a total of 6 components (including nitrogen from air), the detailed reactor model (5-14)-(5-20) includes 43 variables, 9 of which are fixed. All these equations and variables can be replaced by a single surrogate relating only 4 variables.

The surrogate reactor model here was generated as an ANN with a linear output layer, 2 hidden layers with 5 neurons per layer, using tanh(.) as the transfer function. The maximum deviation obtained for the dependent variables was 0.4% of the mapping range. Figure 54 presents the fitting capabilities of the surrogate model.

The optimization problem was posed as and NLP with 32 equations and 35 variables, and solved using CONOPT in GAMS v22.7 to obtain an optimal reactor temperature of 670 K and an optimal reactor volume of 40m^3. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU

920 @ 2.67 GHz. The time required for the generation of the surrogate was 30.4 s and the time required for the solution of the simplified optimization model was 0.13s

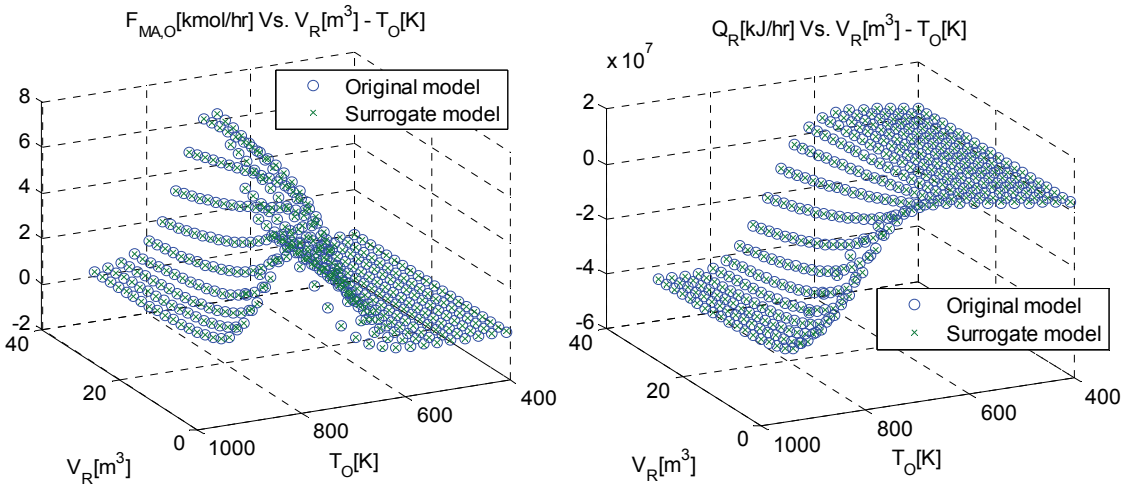


Figure 54: Data fitting for a CSRT surrogate model in the production of Maleic Anhydride from Benzene.

8.3 Production of Maleic Anhydride: simple superstructure optimization

Consider the simple process superstructure presented in Figure 54, for the production on Maleic Anhydride from Benzene. It consists of eight units: one stirred tank reactor (SR1), one plug flow reactor (PR1), one flash tank (FV1), one mixer-splitter (MX1) two raw material sources (SC1-SC2) and two product sinks (SK1-SK2). The goal is to find the best reactor configuration (CSTR or PFR), reactor temperature and volume, flash temperature and recycle fraction, maximizing the annualized profit accounting for revenue from Maleic Anhydride at current price (www.icis.com), capital costs [72], and utility costs.

The system feed, kinetics and annualized profit calculations are the same used previously for the CSTR optimization problem. The thermodynamic surrogate includes 2 hidden layers with 12 neurons

per layer, the CSTR and PFR surrogates include 2 hidden with 7 neurons per layer. The surrogate for the flash vessel includes 2 hidden layers and 15 neurons in each layer. The formulation involves 1434 variables and 1770 constraints (both equalities and inequalities). The constraints using the tanh(.) function add up to 330, and total number of constraints with bilinear terms is 64. The optimization was performed using GAMS 22.7 –DICOPT. The optimal solution involves a 53 m^3 PFR at 653K, a flash unit at 340 K, and recycle stream that is equal to 52% of the flash vapor stream. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU 920 @ 2.67 GHz. The times required for the generation of each surrogate mapping varies between 30 seconds (i.e. thermodynamics) and 240 seconds (i.e. flash unit). The time required to solve the optimization model was 17.56 s.

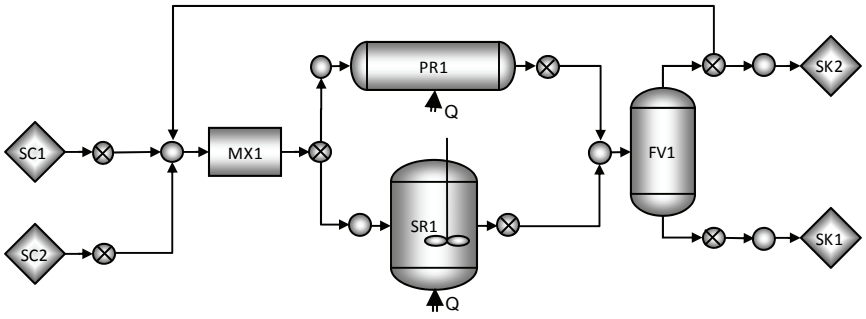


Figure 55: Data fitting for a CSRT surrogate model in the production of Maleic Anhydride from Benzene.

8.4 Linear superstructure model optimization

Consider the process superstructure presented in Figure 56. Having raw materials "A", "B" the purpose is to produce as much component "D" as possible while minimizing the generation of byproduct "E". The following stoichiometry accounts for the chemical transformations:

$$2A + B \rightarrow C + D \quad (rxn1)$$
$$C + 3B \rightarrow 2D \quad (rxn2)$$
$$A + D \rightarrow E \quad (rxn3)$$

It is assumed that components can be separated using LV operations (distillation columns and flash vessels) based on volatilities which follow the order $\alpha_A > \alpha_B > \alpha_C > \alpha_D > \alpha_E$. In general, the reaction effluent can carry all components. It is convenient to recover and recycle components A,B,C, purify component D and separate-dispose component E.

Linear capital cost functions were included in the element model linear formulations. Capital cost coefficients calculated by fitting cost data are presented in Table 5. Raw material source unit SC1 provides 100 kmol/h of A, 10 kmol/h of B, 3 kmol/h of C, 1 kmol/h of D and 2 kmol/h of E at 200 K and 100 kPa. Raw material source unit SC2 provides 7 kmol/h of A, 100 kmol/h of B, 5 kmol/h of C, 2 kmol/h of D and 3 kmol/h of E at 180K and 150kPa. The objective is to maximize the net annual profit considering an annual evaluation interest rate of 10% and an equipment lifespan of 20 years. The prices of components are π_A=5 USD/kmol , π_B=5 USD/kmol π_C = 10 USD/kmol and π_D =15 USD/kmol. The average heat transfer utility price is 3.6E-6 USD/kJ, and the average electricity price is 1.39E-5 USD/kJ

Outlet port models are completely linear (bilinear terms replaced with the mixed integer formulation presented in chapter 4), using a list of 21 possible stream split values. The complete SO model is an MINLP including 4002 constraints and 3584 variables. The optimization was performed using GAMS 22.7 – CPLEX. The optimal structure is presented in Figure 56. All calculations were performed in a PC with an Intel(R) Core(TM)i7 CPU 920 @ 2.67 GHz. The time required to solve the model was 51. 6 seconds.\

Table 5: Capital cost coefficients for different units.

Unit	$Cc\__o$ [USD]	$Cc\__M$ [USD/unit cap.]	Capacity [Unit cap.]
Plug Reactor	16504.13	1596.556	Unit Volume[m^3]
Stirred Reactor	18037.77	60.6866	Unit Volume[m^3]
Flash Vessel	3937.227	608.3078	Unit Volume[m^3]
Distillation Column	3877.626	715.2378	Unit Volume[m^3]
$Cc_u = Cc\_o_u + Cc\_{Mu}(Unit\ Capacity)$			

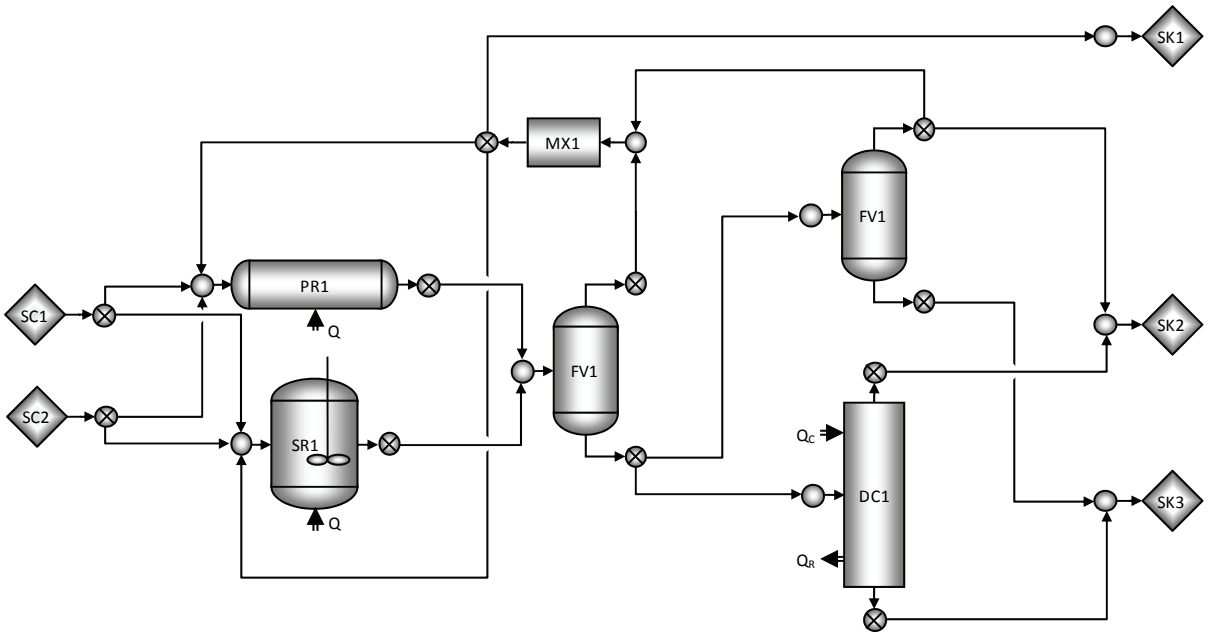


Figure 56: Process superstructure for the production of D from A and B.

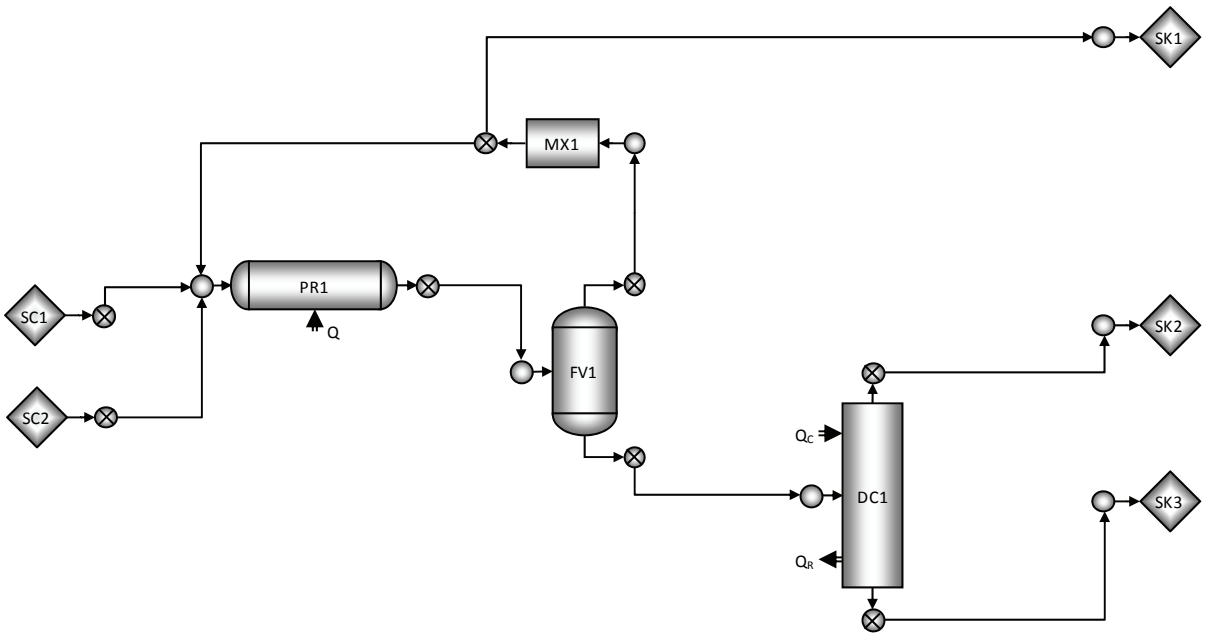


Figure 57: Process superstructure for the production of D from A and B.

Chapter 9

CONCLUSIONS

This chapter highlights the main contributions presented in this thesis and points out some directions to continue the outlined research.

9.1 Contributions

Superstructure representation: Although some authors have used units and unit ports in the representation of superstructures (e.g. R-graphs [24],[25]) the use of general conditioning streams is novel. As mentioned in chapter 3 this representation discriminates between main processing units (i.e. reactors and separators) and conditioning units (i.e. heat exchangers, compressors, turbines, pumps, expansion valves), providing a general approach to deal with the latter while emphasizing the importance of the former. This distinction facilitates the superstructure generation, since the designer can focus on the processing units assuming all required conditioning for their proper operation can be provided. In addition, this approach facilitates the extension of the superstructure model trough the use of aggregated models for heat and power integration (e.g. pinch analysis based on transshipment models [83])

Superstructure generation: For a given set of units, the use of simplification rules generates a superstructure connectivity which is simpler than the full connectivity used in general SEN representations, and usually richer than the connectivity defined in STN representations. To our knowledge, the use of these rules based on minimal and feasible component lists for each unit port is novel. In addition, it allows the automatic generation of superstructures using, for example, the set manipulation capabilities of a modeling language such as GAMS. Finally, it is worth noting that this approach can be used to incorporate engineering judgment in the form of additional connectivity rules.

Superstructure modeling: In contrast to ad hoc models found very frequently in the literature, this work introduces a general modeling framework with a well defined structure and a complete set of models for superstructure elements, including the most common processing and conditioning units used in the synthesis of chemical processes. Of particular importance is the modular nature of the framework, allowing a fast formulation and implementation complete SO models from scratch, as well as the efficient modification of existing models. In addition, as a result of this modularity, the formulation and implementation of new unit models is very simple.

Surrogate models: In the area of process systems engineering, there are several publications advocating the use of surrogate models as a way to reduce the complexity and simplify the solution of process optimization problems. However, the systematic design of surrogate models (i.e. selection of independent and dependent variables) presented here is new. In addition, several authors use surrogates as black box models relating the required set of dependent and independent variables [51],[52],[53]. In contrast, we formulate our unit models by mixing as many linear equations from the detailed model as possible with a small set of surrogate mappings replacing the complex equation blocks. This approach improves the behavior of the models since at least the key

conservation principles (i.e. material and energy balances), are explicitly accounted for, contributing also to the use of more compact surrogate mappings (since they do not relate as many model variables). A good illustration of this idea is the use of a linear energy balances in every unit, and a single thermodynamic surrogate relating port state variables with port thermodynamic properties. This linear equation eliminates the need to calculate all outlet port temperatures for every unit using dedicated surrogate models. Instead, the energy balance can be used to calculate one of the port temperatures.

9.2 Future work

Connectivity simplification rules: As mentioned, connectivity rules can be a way to formalize and incorporate engineering judgment in the definition of a superstructure. The rules presented in section 3.3 are simple and fundamental, but there is still room for improvement, and new rules could be devised trough the analysis of specific process design cases.

Automatic updating of surrogate domains: As discussed in section 7.2, the implementation of a subroutine to update the surrogate mapping domain is a key element in the practical use of the surrogate approach presented here. This is particularly true in superstructures with high connectivity and complicated units such as separation columns. In these cases, the surrogate domains have to be updated several times before finding an optimal solution, due to the complicated unit iteration and the tight surrogate domains required to obtain accurate surrogate mappings.

Use of other surrogate mappings : In his work, ANNs where used exclusively; however, the use of other surrogate forms should be considered and properly evaluated. As in the case of ANNs, the

selection of new types of surrogate mappings has to be based on the existence of techniques to deal with the surrogate nonlinearity (i.e. proper MINLP reformulations, convexification techniques, etc.)

Implementation and refinement of tailored global solution algorithms: As discussed in section 7.5, finding global solutions even for medium size SO models can be challenging due to the current limitations of state of the art numerical solvers. As a result, development of tailored convexification schemes and global solution algorithms to deal with the specific surrogate nonlinearities could result in great contributions to the overall capabilities of the methodology presented in this thesis.

APPENDICES

Appendix A

Superstructure connectivity evaluation for the HDA process

Table 6: Evaluation of all possible port connections in the HDA process superstructure.

Connection		Rules satisfied			Valid ?	Connection		Rules satisfied			Valid ?
Outlet port	Inlet port	1	3	4		Outlet port	Inlet port	1	3	4	
SC1-O1	SK1-I1	Y		N	N	FV2-O2	SK1-I1	N		N	N
	SK2-I1	N		N	N		SK2-I1	Y		Y	Y
	SK3-I1	N		N	N		SK3-I1	N		Y	N
	PR1-I1	Y	Y		Y		PR1-I1	N	Y		N
	SR1-I1	Y	Y		Y		SR1-I1	N	Y		N
	FV1-I1	Y		N	N		FV1-I1	Y		N	N
	FV2-I1	N		N	N		FV2-I1	Y		N	N
	FV3-I1	N		N	N		FV3-I1	N		Y	N
	AC1-I1	Y		N	N		AC1-I1	Y		N	N
	AC1-I2	N		N	N		AC1-I2	Y		Y	Y
	DC1-I1	N		N	N		DC1-I1	Y		Y	Y

Table 7: Evaluation of all possible port connections in the HDA process superstructure (cont.)

Connection		Rules satisfied			Valid ?	Connection		Rules satisfied			Valid ?
Outlet port	Inlet port	1	3	4		Outlet port	Inlet port	1	3	4	
SC2-O2	SK1-I1	N		N	N	FV3-O1	SK1-I1	N		N	N
	SK2-I1	N		N	N		SK2-I1	N		N	N
	SK3-I1	N		N	N		SK3-I1	N		Y	N
	PR1-I1	Y	Y		Y		PR1-I1	Y	Y		Y
	SR1-I1	Y	Y		Y		SR1-I1	Y	Y		Y
	FV1-I1	Y		N	N		FV1-I1	Y		N	N
	FV2-I1	Y		N	N		FV2-I1	Y		N	N
	FV3-I1	Y		N	N		FV3-I1	Y		Y	Y
	AC1-I1	Y		N	N		AC1-I1	Y		N	N
	AC1-I2	Y		Y	Y		AC1-I2	Y		Y	Y
	DC1-I1	Y		N	N		DC1-I1	Y		N	N
PR1-O1	SK1-I1	N		Y	N	FV3-O2	SK1-I1	N		N	N
	SK2-I1	N		Y	N		SK2-I1	N		N	N
	SK3-I1	N		Y	N		SK3-I1	Y		Y	Y
	PR1-I1	N	Y		N		PR1-I1	Y	Y		Y
	SR1-I1	N	Y		N		SR1-I1	Y	Y		Y
	FV1-I1	Y		Y	Y		FV1-I1	Y		N	N
	FV2-I1	N		Y	N		FV2-I1	Y		N	N
	FV3-I1	N		Y	N		FV3-I1	Y		N	N
	AC1-I1	Y		Y	Y		AC1-I1	Y		N	N
	AC1-I2	N		Y	N		AC1-I2	Y		N	N
	DC1-I1	N		Y	N		DC1-I1	Y		N	N
SR1-O1	SK1-I1	N		Y	N	AC1-O1	SK1-I1	Y		N	N
	SK2-I1	N		Y	N		SK2-I1	N		N	N
	SK3-I1	N		Y	N		SK3-I1	N		N	N
	PR1-I1	N	Y		N		PR1-I1	Y	Y		Y
	SR1-I1	N	Y		N		SR1-I1	Y	Y		Y
	FV1-I1	Y		Y	Y		FV1-I1	Y		N	N
	FV2-I1	N		Y	N		FV2-I1	N		N	N
	FV3-I1	N		Y	N		FV3-I1	N		N	N
	AC1-I1	Y		Y	Y		AC1-I1	Y		N	N
	AC1-I2	N		Y	N		AC1-I2	N		N	N
	DC1-I1	N		Y	N		DC1-I1	N		N	N

Table 8: Evaluation of all possible port connections in the HDA process superstructure (cont.)

Connection		Rules satisfied			Valid ?	Connection		Rules satisfied			Valid ?
Outlet port	Inlet port	1	3	4		Outlet port	Inlet port	1	3	4	
FV1-O1	SK1-I1	Y		Y	Y	AC1-O2	SK1-I1	N		Y	N
	SK2-I1	N		N	N		SK2-I1	N		Y	N
	SK3-I1	N		N	N		SK3-I1	N		Y	N
	PR1-I1	Y		Y	Y		PR1-I1	Y	Y		Y
	SR1-I1	Y		Y	Y		SR1-I1	Y	Y		Y
	FV1-I1	Y		N	N		FV1-I1	Y		Y	Y
	FV2-I1	Y		N	N		FV2-I1	Y		Y	Y
	FV3-I1	N		N	N		FV3-I1	N		Y	N
	AC1-I1	Y		Y	Y		AC1-I1	Y		N	N
	AC1-I2	N		N	N		AC1-I2	N		Y	N
	DC1-I1	N		N	N		DC1-I1	N		Y	N
FV1-O2	SK1-I1	N		Y	N	DC1-O1	SK1-I1	N		N	N
	SK2-I1	N		Y	N		SK2-I1	Y		Y	Y
	SK3-I1	N		Y	N		SK3-I1	N		N	N
	PR1-I1	N		Y	N		PR1-I1	N	N		N
	SR1-I1	N		Y	N		SR1-I1	N	N		N
	FV1-I1	Y		Y	Y		FV1-I1	Y		N	N
	FV2-I1	Y		Y	Y		FV2-I1	Y		N	N
	FV3-I1	N		Y	N		FV3-I1	N		N	N
	AC1-I1	Y		N	N		AC1-I1	Y		N	N
	AC1-I2	N		Y	N		AC1-I2	Y		N	N
	DC1-I1	N		Y	N		DC1-I1	Y		N	N
FV2-O1	SK1-I1	N		Y	N	DC1-O2	SK1-I1	N		N	N
	SK2-I1	N		Y	N		SK2-I1	N		N	N
	SK3-I1	N		Y	N		SK3-I1	N		Y	N
	PR1-I1	N	Y		N		PR1-I1	Y		Y	Y
	SR1-I1	N	Y		N		SR1-I1	Y		Y	Y
	FV1-I1	Y		Y	Y		FV1-I1	Y		N	N
	FV2-I1	Y		Y	Y		FV2-I1	Y		N	N
	FV3-I1	N		N	N		FV3-I1	Y		Y	Y
	AC1-I1	Y		N	N		AC1-I1	Y		N	N
	AC1-I2	N		N	N		AC1-I2	Y		Y	Y
	DC1-I1	N		N	N		DC1-I1	Y		N	N

Appendix B

Mol fraction based formulation for superstructure element models

One of the disadvantages of the formulations in section 4.4 is the implicit disregard for the bilinear nature of variable relationships. For example, in the calculation of enthalpies two different formulations using two different nonlinear mappings can be used:

$$\left[H = \mathrm{f}^{H}\left(\left[F_c\right]_{c\in\mathbf{C}}, T, P\right)\right] \tag{B-1}$$

$$\left[\begin{array}{l} F = \sum_{c\in\mathbf{C}} F_c \quad , \quad F_c = F \cdot x_c \quad \forall c \in \mathbf{C} \\ H = F \cdot h \quad , \quad h = \mathrm{f}^{h}\left(\left[x_c\right]_{c\in\mathbf{C}^*}, T, P\right) \quad , \quad \mathbf{C}^* = \left\{c \in \mathbf{C} \,/\, ord\left(c\right) < \left|\mathbf{C}\right|\right\} \end{array}\right] \tag{B-2}$$

(B-1) is the F_c-formulation we have used so far. (B-2) is an $F_c - x_c$-formulation which establishes the same relationship between the component flows, temperature and pressure and enthalpy flows, but involves the use of a nonlinear mapping in terms of mol fractions, temperature and pressure. That is, a function of intensive variable exclusively. This formulation presents the enthalpy flow as the product of a molar enthalpy and a total molar flow. Formulation (B-2) seems to be more

complex than formulation (B-1) since it includes an additional linear constraint and several ($|\mathbf{C}|+1$) bilinear ones. However, the surrogate mapping domain is more compact and easier to bound since, as opposed to molar flows F_c, the number of independent mol fractions x_c is one less than the number of components (as they are related by $\Sigma_{c\in\mathbf{C}} x_c = 1$), and their bounds are universal (i.e. $\Sigma_{c\in\mathbf{C}} x_c = 1$, $x_c \geq 0 \; \forall c \in \mathbf{C}$) instead of problem specific (i.e. $F^{Lo} \leq F_c \leq F^{Up} \; \forall c \in \mathbf{C}$). These conditions simplify the data generation and fitting required to build proper nonlinear surrogate mappings.

By allowing bilinear equations as those in (B-2), we have developed an alternative version for the superstructure element models included in section 4.4.

B.1 Unit ports

Inlet ports $\left(p \in \mathbf{P}^{\mathbf{I}}\right)$*:* Following the guidelines in model (4-7)-(4-10), the following is an alternative formulation:

$$F_{c,p} = \sum_{s\in\mathbf{S}_p^{\mathbf{I}}} F_{c,s} \quad , \quad \forall c \in \mathbf{C} \tag{B-3}$$

$$H_p = \sum_{s\in\mathbf{S}_p^{\mathbf{I}}} H_s^{S3} \tag{B-4}$$

$$P_p = P_s^{S3} \quad , \quad \forall s \in \mathbf{S}_p^{\mathbf{I}} \tag{B-5}$$

$$F_p = \sum_{c\in\mathbf{C}} F_{c,p} \quad , \quad F_p \cdot x_{c,p} = F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{B-6}$$

$$H_p = F_p \cdot h_p \quad , \quad S_p = F_p \cdot s_p \tag{B-7}$$

$$h_p = \mathrm{f}^h\left(\left[x_{c,p}\right]_{c\in\mathbf{C}^*}, T_p, P_p\right) \quad , \quad s_p = \mathrm{f}^s\left(\left[x_{c,p}\right]_{c\in\mathbf{C}^*}, T_p, P_p\right) \tag{B-8}$$

Equations (B-3) and (B-4) are component and energy balances. Equation (B-5) is the pressure balance condition between inlet streams. Equations (B-6) relate port component flows, total flow and molar fractions. Finally, equations (B-7), (B-8) are thermodynamic relationships.

Unit outlet ports $\left(p'\in \mathbf{P}^{\mathbf{O}}\right)$*:* Following some the guidelines in model (4-11)-(4-17), the following is an alternative formulation:

$$\sum_{s\in\mathbf{S}^{\mathbf{O}}_{p'}} \xi_s = 1 \tag{B-9}$$

$$F_{c,s} = \xi_s \cdot F_{c,p'} \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}^{\mathbf{O}}_{p'} \tag{B-10}$$

$$T_s^{S1} = T_{p'} \quad , \quad \forall s \in \mathbf{S}^{\mathbf{O}}_{p'} \tag{B-11}$$

$$P_s^{S1} = P_{p'} \quad , \quad \forall s \in \mathbf{S}^{\mathbf{O}}_{p'} \tag{B-12}$$

$$x_{c,s} = x_{c,p'} \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}^{\mathbf{O}}_{p'} \tag{B-13}$$

$$h_s^{S1} = h_{p'} \quad , \quad \forall s \in \mathbf{S}^{\mathbf{O}}_{p'} \tag{B-14}$$

$$s_s^{S1} = s_{p'} \quad , \quad \forall s \in \mathbf{S}^{\mathbf{O}}_{p'} \tag{B-15}$$

$$F_{p'} = \sum_{c\in\mathbf{C}} F_{c,p'} \quad , \quad F_{p'} \cdot x_{c,p'} = F_{c,p'} \quad , \quad \forall c \in \mathbf{C} \tag{B-16}$$

$$H_{p'} = F_{p'} \cdot h_{p'} \quad , \quad S_{p'} = F_{p'} \cdot s_{p'} \tag{B-17}$$

$$h_{p'} = \mathrm{f}^h\left(\left[x_{c,p'}\right]_{c\in\mathbf{C}^*}, T_{p'}, P_{p'}\right) \quad , \quad s_{p'} = \mathrm{f}^s\left(\left[x_{c,p'}\right]_{c\in\mathbf{C}^*}, T_{p'}, P_{p'}\right) \tag{B-18}$$

Equations (B-9) and (B-10) are the basic relationship between stream split fractions and component flows in a splitter. Equations (B-11)-(B-15) make the intensive properties of port p' equal to those in state 1 for streams leaving p'. They are a consequence of the port adiabatic-isobaric operation and the way the components are distributed among the outlet streams. Equations (B-16) relate component flows and total port flow to port mol fractions. Finally, equations (B-17) and (B-18) are thermodynamic relationships.

B.2 Streams

Conditioning streams $\left(s \in \mathbf{S}^{\mathbf{C}}\right)$*:* Following the guidelines in model (4-29)-(4-35), the following is an alternative formulation:

$$0 \le \xi_s \le 1 \tag{B-19}$$

$$H_s^{S2} = H_s^{S1} + Q_s \quad , \quad H_s^{S3} = H_s^{S2} + W_s \tag{B-20}$$

$$P_s^{S2} = P_s^{S1} \quad , \quad P_s^{S3} = P_s^{S2} + \Delta P_s \tag{B-21}$$

$$s_s^{S3} = s_s^{S2} \tag{B-22}$$

$$F_s = \sum_{c \in \mathbf{C}} F_{c,s} \tag{B-23}$$

$$H_s^{S1} = F_s \cdot h_s^{S1} \quad , \quad H_s^{S2} = F_s \cdot h_s^{S2} \quad , \quad H_s^{S3} = F_s \cdot h_s^{S3} \tag{B-24}$$

$$S_s^{S1} = F_s \cdot s_s^{S1} \quad , \quad S_s^{S2} = F_s \cdot s_s^{S2} \quad , \quad S_s^{S3} = S_s^{S2} \tag{B-25}$$

$$\begin{aligned} h_s^{S2} &= \mathrm{f}^h\left(\left[x_{c,s}\right]_{c \in \mathbf{C}^*}, T_s^{S2}, P_s^{S2}\right) \quad , \quad h_s^{S3} = \mathrm{f}^h\left(\left[x_{c,s}\right]_{c \in \mathbf{C}^*}, T_s^{S3}, P_s^{S3}\right) \\ s_s^{S2} &= \mathrm{f}^S\left(\left[x_{c,s}\right]_{c \in \mathbf{C}^*}, T_s^{S2}, P_s^{S2}\right) \quad , \quad s_s^{S3} = \mathrm{f}^s\left(\left[x_{c,s}\right]_{c \in \mathbf{C}^*}, T_s^{S3}, P_s^{S3}\right) \end{aligned} \tag{B-26}$$

$$Cc_s = \mathrm{f}^{Cc,HC}\left(Q_s, T_s^{S1}, P_s^{S1}, T_s^{S2}\right) + \mathrm{f}^{Cc,CE}\left(W_s, T_s^{S3}, P_s^{S3}\right) \quad \text{(B-27)}$$

$$Co_s = \mathrm{f}^{Co,HC}\left(Q_s, T_s^{S1}, T_s^{S2}\right) + \mathrm{f}^{Co,CE}\left(W_s\right) \quad \text{(B-28)}$$

Constraints (B-19)-(B-21) are the same as (4-29)-(4-31). Equation (B-22) is the isentropic condition in the compression-expansion operation. Equation (B-23) relates component flows. Notice that the relationship between stream component flows and compositions is not included since the latter are calculated at the origin port using (B-13). This reduces the total number of bilinear terms in the model. Equations (B-24), (B-25), (B-26) are thermodynamic relationships. Equations (B-27), (B-28) are cost relationships.

Non conditioning streams $\left(s \in \mathbf{S^{NC}}\right)$*:* The model is again very simple, and composed entirely by simple equalities as in (4-36) to (4-42):

$$0 \le \xi_s \le 1 \quad \text{(B-29)}$$

$$H_s^{S3} = H_s^{S1} \quad \text{(B-30)}$$

$$S_s^{S3} = S_s^{S1} \quad \text{(B-31)}$$

$$h_s^{S3} = h_s^{S1} \quad \text{(B-32)}$$

$$s_s^{S3} = s_s^{S1} \quad \text{(B-33)}$$

$$P_s^{S3} = P_s^{S1} \quad \text{(B-34)}$$

$$T_s^{S3} = T_s^{S1} \quad \text{(B-35)}$$

$$Cc_s = 0 \tag{B-36}$$

$$Co_s = 0 \tag{B-37}$$

Once more, the intensive and extensive properties of states S1 and S3 are made equal, while bilinear terms are completely avoided.

B.3 General unit operation

For a general unit $\left(u \in \mathbf{U}^{\mathrm{G}}\right)$, using the guidelines in (4-43)-(4-52), the following is an alternative formulation:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{B-38}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + \sum_{l \in \{1,\ldots,L\}} Q_{l,u} + \sum_{o \in \{1,\ldots,O\}} W_{o,u} \tag{B-39}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{B-40}$$

$$P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{B-41}$$

$$fr_p \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p = F_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{B-42}$$

$$F_{c,p'} = \phi_{c,p'} \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord\left(p'\right) \neq 1, \forall c \in \mathbf{C} \tag{B-43}$$

$$\phi_{c,p'} = \mathrm{f}_{c,p'}^{\phi,\mathrm{G}}\left(\left[x_{c,p}, T_p\right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta} P_u, \left[fr_p\right]_{\forall p \in \mathbf{P}_u^{\mathbf{I}^*}}\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord\left(p'\right) \neq 1, \forall c \in \mathbf{C} \tag{B-44}$$

$$T_{p'} = \mathrm{f}_{p'}^{T,\mathrm{G}}\left(\left[x_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}^* \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta P}_u, \left[fr_p\right]_{\forall p\in\mathbf{P}_u^{\mathbf{I}*}}\right) \quad , \quad \forall p'\in \mathbf{P}_u^{\mathbf{O}} / ord(p') \neq 1 \tag{B-45}$$

$$Q_{l,u} = q_{l,u} \cdot \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_p \quad , \quad \forall l \in \{1,...,L\} \tag{B-46}$$

$$q_{l,u} = \mathrm{f}_{l,u}^{q,G}\left(\left[x_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}^* \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta P}_u, \left[fr_p\right]_{\forall p\in\mathbf{P}_u^{\mathbf{I}*}}\right) \quad , \quad \forall l \in \{1,..., L\} \tag{B-47}$$

$$W_{o,u} = w_{o,u} \cdot \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_p \quad , \quad \forall l \in \{1,...,L\} \tag{B-48}$$

$$w_{o,u} = \mathrm{f}_{o,u}^{w,G}\left(\left[x_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}^* \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta P}_u, \left[fr_p\right]_{\forall p\in\mathbf{P}_u^{\mathbf{I}*}}\right) \quad , \quad \forall o \in \{1,..., O\} \tag{B-49}$$

$$Cc_u = \mathrm{f}^{Cc,\mathrm{G}}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C} \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, Pu, \boldsymbol{\psi}_u, \underline{\Delta P}_u\right) \tag{B-50}$$

$$Co_u = \mathrm{f}^{Co,G}\left(\left[Q_{l,u}\right]_{l\in\{1,...,L\}}, \left[W_{o,u}\right]_{o\in\{1,...,O\}}\right) \tag{B-51}$$

Equations (B-38) and (B-39) are component and energy balances. Equation (B-40) represents the unit hydraulics. Equation (B-41) is an internal pressure balance condition. Equation (B-42) defines the inlet port flow ratios fr_p, intensive variables indicating the proportions in which materials enter the unit trough different ports. Equations (B-43) and (B-44) calculate component flows leaving the unit trough all but one of outlet ports. Equation (B-45) estimates all but one of the outlet port temperatures. Equations (B-46)-(B-49) calculate heating utility and power consumption. Finally, equations (B-50) and (B-51) are capital and operative costs relationships.

It is important to notice that the surrogates to calculate the intensive variables $\phi_{c,p'}$, $T_{p'}$, $q_{l,u}$ and $w_{o,u}$ include only intensive independent variables. The capital cost calculation cannot take this form;

that is, it cannot be expressed as the product of some "intensive capital cost" and a proper extensive variable if economy of scale is to be accounted for. In this way, even in these formulations, capital cost surrogates in terms of molar flows instead of molar fractions are still required.

To illustrate the logic behind the model, consider the degrees of freedom and the following calculation sequence. The model degrees of freedom are the same as in model (4-43)-(4-52): component flows and temperatures of all inlet streams $\left\lfloor F_{c,p}, T_p \right\rfloor_{c\in\mathbf{C}, p\in\mathbf{P}_u^{\mathrm{I}}}$, the unit internal pressure P_u, the unit pressure drop $\underline{\Delta}P_u$, and other unit internal variables $\boldsymbol{\psi}_u$. Using (B-6) inlet port total flows and mol fractions are calculated. Using (B-42), all inlet port flow ratios are calculated. Using (B-40) and (B-41), all port pressures are calculated. Using (B-44) all component split fractions are calculated for all outlet ports except one. Using (B-43) and (B-38) all outlet port component flows are calculated. Using (B-46)-(B-49) all heat and work additions to the units are calculated. Using the enthalpy equations in (B-7)-(B-8) all inlet port enthalpy flows are calculated. Using (B-39), (B-45) and the enthalpy equations in (B-17)-(B-18), an equation system can be posed to find all outlet port enthalpies and temperatures. Finally capital and operating costs can be calculated with (B-50) and (B-51).

Appendix C

MINLP reformulations of superstructure element models

C.1 Component flow based formulation

In this section the general reformulation form (4-58) is used to propose MINLP reformulations to the models included in section 4.4.

C.1.1 Unit ports

Inlet ports $\left(p \in \mathbf{P^I}\right)$: Using the unit binaries to activate/deactivate the port, the reformulation of (4-7)-(4-10) is as follows:

$$F_{c,p} = \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} F_{c,s} \quad , \quad \forall c \in \mathbf{C} \tag{C-1}$$

$$H_p = \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} H_s^{S3} \tag{C-2}$$

$$P_p = P_s^{S3} + \Delta P_p \quad , \quad 0 \le \Delta P_p \le P^{Up}.\left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{S}_p^{\mathbf{I}} \tag{C-3}$$

$$H_p = \bar{\mathrm{f}}^{H}\left(\left[F_{c,p}\right]_{c\in\mathbf{C}}, T_p, P_p, y_u\right) \quad , \quad S_p = \bar{\mathrm{f}}^{S}\left(\left[F_{c,p}\right]_{c\in\mathbf{C}}, T_p, P_p, y_u\right) \quad , \quad \forall u \in \mathbf{U}_p \tag{C-4}$$

$$F_{c,\mathbf{P}}^{Lo} \cdot y_u \le F_{c,p} \le F_{c,\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_p, \forall c \in \mathbf{C} \tag{C-5}$$

$$T_{\mathbf{P}}^{Lo} \cdot y_u \le T_p \le T_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_p \tag{C-6}$$

$$P_{\mathbf{P}}^{Lo} \cdot y_u \le P_p \le P_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_p \tag{C-7}$$

Here, $F_{c,\mathbf{P}}^{Lo}, F_{c,\mathbf{P}}^{Up}, T_{\mathbf{P}}^{Lo}, T_{\mathbf{P}}^{Up}, P_{\mathbf{P}}^{Lo}, P_{\mathbf{P}}^{Up}$ are lower and upper bounds defining the model domain via (C-5)-(C-7), which has to be interior to the domain of the nonlinear mappings $\bar{\mathrm{f}}^{H}(\cdot)$ and $\bar{\mathrm{f}}^{S}(\cdot)$. Notice how an additional $\underline{\Delta}P_{p'}$ variable had to be introduced and properly bounded in (C-3) to avoid the pressure of an active port going to zero during the deactivation of any of the inlet port streams. Also, port p is activated/deactivated using the binary y_u of the unit it belongs to. That is, we use the implications of (4-54) to reduce the number of binaries in our superstructure model.

Unit outlet ports $\left(p' \in \mathbf{P}^{\mathbf{O}}\right)$*:* The mixed integer reformulation of the bilinear model version (4-11)-(4-17) is as follows:

$$\sum_{s\in\mathbf{S}_{p'}^{\mathbf{O}}} \xi_s = y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-8}$$

$$F_{c,s} = \xi_s \cdot F_{c,p'} \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-9}$$

$$H_s^{S1} = \xi_s \cdot H_{p'} \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-10}$$

$$S_s^{S1} = \xi_s \cdot S_{p'} \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-11}$$

$$T_s^{S1} = T_{p'} - \underline{\Delta}T_{p'} \quad , \quad 0 \le \underline{\Delta}T_{p'} \le T^{Up}.(1 - y_s) \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-12}$$

$$P_s^{S1} = P_{p'} - \underline{\Delta}P_{p'} \quad , \quad 0 \le \underline{\Delta}P_{p'} \le P^{Up}.(1 - y_s) \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-13}$$

$$H_{p'} = \bar{\mathrm{f}}^{H}\left(\left[F_{c,p'}\right]_{c\in\mathbf{C}}, T_{p'}, P_{p'}, y_u\right) \quad , \quad S_{p'} = \bar{\mathrm{f}}^{S}\left(\left[F_{c,p'}\right]_{c\in\mathbf{C}}, T_{p'}, P_{p'}, y_u\right) \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-14}$$

$$F_{c,\mathbf{P}}^{Lo} \cdot y_u \le F_{c,p'} \le F_{c,\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_{p'}, \forall c \in \mathbf{C} \tag{C-15}$$

$$T_{\mathbf{P}}^{Lo} \cdot y_u \le T_{p'} \le T_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-16}$$

$$P_{\mathbf{P}}^{Lo} \cdot y_u \le P_{p'} \le P_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-17}$$

Notice how (C-8) is the reformulated version of the only non-homogeneous linear equation in the original port model, and how additional $\underline{\Delta}T_{p'}$, $\underline{\Delta}P_{p'}$ variables had to be introduced and properly bounded in (C-12) and (C-13) to avoid the temperature and pressure of an active port going to zero during the deactivation of any of the outlet port streams. Here, as in the inlet port model, outlet port p' is activated/deactivated using the binary y_u of the unit it belongs to.

If a purely linear formulation is required, the following reformulated constraints can be used, instead of equations (C-8)-(C-10):

$$\sum_{k\in\mathbf{K}} y_s^k = y_u \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall u \in \mathbf{U}_{p'} \tag{C-18}$$

$$\sum_{s\in\mathbf{S}_p^{\mathbf{O}}}\sum_{k\in\mathbf{K}} y_s^k \cdot \xi^k = y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-19}$$

$$F_{c,p'} = \sum_{k\in\mathbf{K}} F_{c,p',s}^k \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-20}$$

$$F^{Lo} \cdot y_s^k \le F_{c,p',s}^k \le F^{Up} \cdot y_s^k \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall k \in \mathbf{K} \tag{C-21}$$

$$F_{c,s} = \sum_{k \in \mathbf{K}} \xi^k \cdot F_{c,p',s}^k \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-22}$$

$$H_{p'} = \sum_{k \in \mathbf{K}} H_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-23}$$

$$H^{Lo} \cdot y_s^k \le H_{p',s}^k \le H^{Up} \cdot y_s^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall k \in \mathbf{K} \tag{C-24}$$

$$H_s^{S1} = \sum_{k \in \mathbf{K}} \xi^k \cdot H_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-25}$$

$$S_{p'} = \sum_{k \in \mathbf{K}} S_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-26}$$

$$S^{Lo} \cdot y_s^k \le S_{p',s}^k \le S^{Up} \cdot y_s^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}}, \forall k \in \mathbf{K} \tag{C-27}$$

$$S_s^{S1} = \sum_{k \in \mathbf{K}} \xi^k \cdot S_{p',s}^k \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-28}$$

Here, $F^{Lo}, F^{Up}, H^{Lo}, H^{Up}$ and S^{Lo}, S^{Up} continue to be absolute bounds on component flows, enthalpy flows and entropy flows. Obviously, F^{Lo}, F^{Up} could be replaced by $F_{c,\mathbf{P}}^{Lo}$ and $F_{c,\mathbf{P}}^{Up}$ in order to have a tighter and a more consistent formulation.

C.1.2 Streams

Conditioning streams $\left(s \in \mathbf{S}^{\mathbf{C}}\right)$*:* Reformulating (4-29)-(4-35) results in:

$$0 \le \xi_s \le y_s \tag{C-29}$$

$$H_s^{S2} = H_s^{S1} + Q_s \quad , \quad H_s^{S3} = H_s^{S2} + W_s \tag{C-30}$$

$$P_s^{S2} = P_s^{S1} \quad , \quad P_s^{S3} = P_s^{S2} + \Delta P_s \tag{C-31}$$

$$S_s^{S3} = S_s^{S2} \tag{C-32}$$

$$\begin{aligned} H_s^{S2} &= \bar{\mathrm{f}}^{H}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S2}, P_s^{S2}, y_s\right) \quad , \quad H_s^{S3} = \bar{\mathrm{f}}^{H}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S3}, P_s^{S3}, y_s\right) \\ S_s^{S2} &= \bar{\mathrm{f}}^{S}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S2}, P_s^{S2}, y_s\right) \quad , \quad S_s^{S3} = \bar{\mathrm{f}}^{S}\left(\left[F_{c,s}\right]_{c\in\mathbf{C}}, T_s^{S3}, P_s^{S3}, y_s\right) \end{aligned} \tag{C-33}$$

$$Cc_s = \bar{\mathrm{f}}^{Cc,HC}\left(Q_s, T_s^{S1}, P_s^{S1}, T_s^{S2}, y_s\right) + \bar{\mathrm{f}}^{Cc,CE}\left(W_s, T^{S3}{}_s, P^{S3}{}_s, y_s\right) \tag{C-34}$$

$$Co_s = \bar{\mathrm{f}}^{Co,HC}\left(Q_s, T_s^{S1}, T_s^{S2}, y_s\right) + \bar{\mathrm{f}}^{Co,CE}\left(W_s, y_s\right) \tag{C-35}$$

$$F_{c,\mathbf{S}}^{Lo} \cdot y_s \le F_{c,s} \le F_{c,\mathbf{S}}^{Up} \cdot y_s \quad , \quad \forall c \in \mathbf{C} \tag{C-36}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S1} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-37}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S2} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-38}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S3} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-39}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S1} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-40}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S2} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-41}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S3} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-42}$$

Here, $F_{c,\mathbf{S}}^{Lo}, F_{c,\mathbf{S}}^{Up}, T_{\mathbf{S}}^{Lo}, T_{\mathbf{S}}^{Up}, P_{\mathbf{S}}^{Lo}, P_{\mathbf{S}}^{Up}$ define the model domain via (C-36)-(C-42), which has to be interior to the domain of all nonlinear mappings $\bar{\mathrm{f}}^{H}(\cdot)$ and $\bar{\mathrm{f}}^{S}(\cdot)$.

Non conditioning streams $\left(s \in \mathbf{S}^{\mathbf{NC}}\right)$*:* Reformulating (4-36) to (4-42) results in:

$$0 \le \xi_s \le y_s \tag{C-43}$$

$$H_s^{S3} = H_s^{S1} \tag{C-44}$$

$$S_s^{S3} = S_s^{S1} \tag{C-45}$$

$$P_s^{S3} = P_s^{S1} \tag{C-46}$$

$$T_s^{S3} = T_s^{S1} \tag{C-47}$$

$$Cc_s = 0 \tag{C-48}$$

$$Co_s = 0 \tag{C-49}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S1} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-50}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S1} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-51}$$

C.1.3 General unit operation model

Taking the model presented in section 4.4.3, for a general unit $\left(u \in \mathbf{U}^{\mathrm{G}}\right)$, the reformulation results in:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{C-52}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + \sum_{l \in \{1,\ldots,L\}} Q_{l,u} + \sum_{o \in \{1,\ldots,O\}} W_{o,u} \tag{C-53}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{C-54}$$

$$P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{C-55}$$

$$F_{c,p'} = \bar{\mathrm{f}}_{c,p'}^{F,\mathrm{G}}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta}P_u, y_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord(p') \neq 1, \forall c \in \mathbf{C} \tag{C-56}$$

$$T_{p'} = \bar{\mathrm{f}}_{p'}^{T,\mathrm{G}}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta}P_u, y_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord(p') \neq 1 \tag{C-57}$$

$$Q_{l,u} = \bar{\mathrm{f}}_{l,u}^{Q,G}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta}P_u, y_u\right) \quad , \quad \forall l \in \{1,..., L\} \tag{C-58}$$

$$W_{o,u} = \bar{\mathrm{f}}_{o,u}^{W,G}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta}P_u, y_u\right) \quad , \quad \forall o \in \{1,..., O\} \tag{C-59}$$

$$Cc_u = \bar{\mathrm{f}}^{Cc,\mathrm{G}}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta}P_u, y_u\right) \tag{C-60}$$

$$Co_u = \bar{\mathrm{f}}^{Co,G}\left(\left[Q_{l,u}\right]_{l\in\{1,...,L\}}, \left[W_{o,u}\right]_{o\in\{1,...,O\}}, y_u\right) \tag{C-61}$$

$$F_{c,\mathbf{U}}^{Lo} \cdot y_u \leq F_{c,p} \leq F_{c,\mathbf{U}}^{Up} \cdot y_u \quad , \quad c \in \mathbf{C}, p \in \mathbf{P}_u^{\mathbf{I}} \tag{C-62}$$

$$T_{\mathbf{U}}^{Lo} \cdot y_u \leq T_p \leq T_{\mathbf{U}}^{Up} \cdot y_u \quad , \quad p \in \mathbf{P}_u^{\mathbf{I}} \tag{C-63}$$

$$\boldsymbol{\psi}_{\mathbf{U}}^{Lo} \cdot y_u \leq \boldsymbol{\psi}_u \leq \boldsymbol{\psi}_{\mathbf{U}}^{Up} \cdot y_u \tag{C-64}$$

$$P_{\mathbf{U}}^{Lo} \cdot y_u \leq P_u \leq P_{\mathbf{U}}^{Up} \cdot y_u \quad , \quad p \in \mathbf{P}_u^{\mathbf{I}} \tag{C-65}$$

$$\underline{\Delta}P_{\mathbf{U}}^{Lo} \cdot y_u \leq \underline{\Delta}P_u \leq \underline{\Delta}P_{\mathbf{U}}^{Up} \cdot y_u \tag{C-66}$$

Here, $F_{c,\mathbf{U}}^{Lo}, F_{c,\mathbf{U}}^{Up}$, $T_{\mathbf{U}}^{Lo}, T_{\mathbf{U}}^{Up}$, $P_{\mathbf{U}}^{Lo}, P_{\mathbf{U}}^{Up}$, $\boldsymbol{\psi}_{\mathbf{U}}^{Lo}, \boldsymbol{\psi}_{\mathbf{U}}^{Up}$ and $\underline{\Delta}P_{\mathbf{U}}^{Lo}, \underline{\Delta}_{\mathbf{U}}^{Up}$ define the model domain enforced in (C-62)-(C-66), which is interior to the domain of all nonlinear mappings. A certain degree of redundancy is inevitable as domain constraints for port state variables are included in the unit model as well as in the port models. However, this helps in accounting for different bounds given to the same independent variables present in different surrogates. For example the temperature range for which the surrogate thermodynamic is valid may be different from the inlet port temperature range for which the unit surrogates are valid.

C.2 Mol fraction based formulation

In this section, the general reformulation form (4-58) is used to propose MINLP reformulations to the models included in appendix B.

C.2.1 Unit ports

Inlet ports $\left(p \in \mathbf{P}^{\mathbf{I}}\right)$*:* Reformulating model (B-3)-(B-8) results in:

$$F_{c,p} = \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} F_{c,s} \quad , \quad \forall c \in \mathbf{C} \tag{C-67}$$

$$H_p = \sum_{s \in \mathbf{S}_p^{\mathbf{I}}} H_s^{S3} \tag{C-68}$$

$$P_p = P_s^{S3} + \Delta P_p \quad , \quad 0 \le \Delta P_p \le P^{Up}.\left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{S}_p^{\mathbf{I}} \tag{C-69}$$

$$F_p = \sum_{c \in \mathbf{C}} F_{c,p} \quad , \quad F_p \cdot x_{c,p} = F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{C-70}$$

$$H_p = F_p \cdot h_p \quad , \quad S_p = F_p \cdot s_p \tag{C-71}$$

$$h_p = \bar{\mathrm{f}}^{h}\left(\left[x_{c,p}\right]_{c\in\mathbf{C}^*}, T_p, P_p, y_u\right) \quad , \quad s_p = \bar{\mathrm{f}}^{s}\left(\left[x_{c,p}\right]_{c\in\mathbf{C}^*}, T_p, P_p, y_u\right) \quad , \quad \forall u \in \mathbf{U}_p \tag{C-72}$$

$$F^{Lo} \cdot y_u \le F_{c,p} \le F^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_p, \forall c \in \mathbf{C} \tag{C-73}$$

$$0 \le x_{c,p} \le y_u \quad , \quad \forall u \in \mathbf{U}_p, \forall c \in \mathbf{C} \tag{C-74}$$

$$T_{\mathbf{P}}^{Lo} \cdot y_u \le T_p \le T_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_p \tag{C-75}$$

$$P_{\mathbf{P}}^{Lo} \cdot y_u \le P_p \le P_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_p \tag{C-76}$$

Here constraint (C-74) is required in order to set mol fractions to zero in case the port is deactivated.

Unit outlet ports $\left(p' \in \mathbf{P}^{\mathbf{O}}\right)$*:* Reformulating model (B-9)-(B-18) results in:

$$\sum_{s\in\mathbf{S}_{p'}^{\mathbf{O}}} \xi_s = y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-77}$$

$$F_{c,s} = \xi_s \cdot F_{c,p'} \quad , \quad \forall c \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-78}$$

$$T_s^{S1} = T_{p'} - \underline{\Delta} T_{p'} \quad , \quad 0 \le \underline{\Delta} T_{p'} \le T^{Up} \cdot \left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-79}$$

$$P_s^{S1} = P_{p'} - \underline{\Delta} P_{p'} \quad , \quad 0 \le \underline{\Delta} P_{p'} \le P^{Up} \cdot \left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-80}$$

$$x_{c,s} = x_{c,p'} - \underline{\Delta} x_{c,p'} \quad , \quad 0 \le \underline{\Delta} x_{c,p'} \le \cdot\left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{C}, \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-81}$$

$$h_s^{S1} = h_{p'} - \underline{\Delta} h_{p'} \quad , \quad 0 \le \underline{\Delta} h_{p'} \le h^{Up} \cdot \left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-82}$$

$$s_s^{S1} = s_{p'} - \underline{\Delta} s_{p'} \quad , \quad 0 \le \underline{\Delta} s_{p'} \le s^{Up} \cdot \left(1 - y_s\right) \quad , \quad \forall s \in \mathbf{S}_{p'}^{\mathbf{O}} \tag{C-83}$$

$$F_{p'} = \sum_{c \in \mathbf{C}} F_{c,p'} \quad , \quad F_{p'} \cdot x_{c,p'} = F_{c,p'} \quad , \quad \forall c \in \mathbf{C} \tag{C-84}$$

$$H_{p'} = F_{p'} \cdot h_{p'} \quad , \quad S_{p'} = F_{p'} \cdot s_{p'} \tag{C-85}$$

$$h_{p'} = \bar{f}^{h}\left(\left[x_{c,p'}\right]_{c \in \mathbf{C}^*}, T_{p'}, P_{p'}, y_u\right) \quad , \quad s_{p'} = \bar{f}^{s}\left(\left[x_{c,p'}\right]_{c \in \mathbf{C}^*}, T_{p'}, P_{p'}, y_u\right) \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-86}$$

$$F_{c,\mathbf{P}}^{Lo} \cdot y_u \le F_{c,p'} \le F_{c,\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_{p'}, \forall c \in \mathbf{C} \tag{C-87}$$

$$0 \le x_{c,p'} \le y_u \quad , \quad \forall u \in \mathbf{U}_{p'}, \forall c \in \mathbf{C} \tag{C-88}$$

$$T_{\mathbf{P}}^{Lo} \cdot y_u \le T_{p'} \le T_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-89}$$

$$P_{\mathbf{P}}^{Lo} \cdot y_u \le P_{p'} \le P_{\mathbf{P}}^{Up} \cdot y_u \quad , \quad \forall u \in \mathbf{U}_{p'} \tag{C-90}$$

C.2.2 Streams

<u>*Conditioning streams*</u> $\left(s \in \mathbf{S}^{\mathbf{C}}\right)$*:* Reformulation of (B-19)-(B-28) results in:

$$0 \le \xi_s \le y_s \tag{C-91}$$

$$H_s^{S2} = H_s^{S1} + Q_s \quad , \quad H_s^{S3} = H_s^{S2} + W_s \tag{C-92}$$

$$P_s^{S2} = P_s^{S1} \quad , \quad P_s^{S3} = P_s^{S2} + \Delta P_s \tag{C-93}$$

$$s_s^{S3} = s_s^{S2} \tag{C-94}$$

$$F_s = \sum_{c \in \mathbf{C}} F_{c,s} \tag{C-95}$$

$$H_s^{S1} = F_s \cdot h_s^{S1} \quad , \quad H_s^{S2} = F_s \cdot h_s^{S2} \quad , \quad H_s^{S3} = F_s \cdot h_s^{S3} \tag{C-96}$$

$$S_s^{S1} = F_s \cdot s_s^{S1} \quad , \quad S_s^{S2} = F_s \cdot s_s^{S2} \quad , \quad S_s^{S3} = S_s^{S2} \tag{C-97}$$

$$\begin{aligned} h_s^{S2} &= \bar{\mathrm{f}}^{h}\left(\left[x_{c,s}\right]_{c\in\mathbf{C}^*}, T_s^{S2}, P_s^{S2}, y_s\right) \quad , \quad h_s^{S3} = \bar{\mathrm{f}}^{h}\left(\left[x_{c,s}\right]_{c\in\mathbf{C}^*}, T_s^{S3}, P_s^{S3}, y_s\right) \\ s_s^{S2} &= \bar{\mathrm{f}}^{S}\left(\left[x_{c,s}\right]_{c\in\mathbf{C}^*}, T_s^{S2}, P_s^{S2}, y_s\right) \quad , \quad s_s^{S3} = \bar{\mathrm{f}}^{s}\left(\left[x_{c,s}\right]_{c\in\mathbf{C}^*}, T_s^{S3}, P_s^{S3}, y_s\right) \end{aligned} \tag{C-98}$$

$$Cc_s = \bar{\mathrm{f}}^{Cc,HC}\left(Q_s, T_s^{S1}, P_s^{S1}, T_s^{S2}, y_s\right) + \mathrm{f}^{Cc,CE}\left(W_s, T_s^{S3}, P_s^{S3}, y_s\right) \tag{C-99}$$

$$Co_s = \bar{\mathrm{f}}^{Co,HC}\left(Q_s, T_s^{S1}, T_s^{S2}, y_s\right) + \bar{\mathrm{f}}^{Co,CE}\left(W_s, y_s\right) \tag{C-100}$$

$$0 \le x_{c,s} \le y_s \quad , \quad \forall c \in \mathbf{C} \tag{C-101}$$

$$h^{Lo} \cdot y_s \le h_s^{S1} \le h^{Up} \cdot y_s \tag{C-102}$$

$$s^{Lo} \cdot y_s \le s_s^{S1} \le s^{Up} \cdot y_s \tag{C-103}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S1} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-104}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S2} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-105}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S3} \le T_{\mathbf{S}}^{Up} \cdot y_s \tag{C-106}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S1} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-107}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S2} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-108}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S3} \le P_{\mathbf{S}}^{Up} \cdot y_s \tag{C-109}$$

<u>*Non conditioning streams*</u> $\left(s \in \mathbf{S}^{\mathbf{NC}}\right)$*:* Reformulation of (B-29)-(B-37) results in:

$$0 \le \xi_s \le y_s \tag{C-110}$$

$$H_s^{S3} = H_s^{S1} \quad \text{(C-111)}$$

$$S_s^{S3} = S_s^{S1} \quad \text{(C-112)}$$

$$h_s^{S3} = h_s^{S1} \quad \text{(C-113)}$$

$$s_s^{S3} = s_s^{S1} \quad \text{(C-114)}$$

$$P_s^{S3} = P_s^{S1} \quad \text{(C-115)}$$

$$T_s^{S3} = T_s^{S1} \quad \text{(C-116)}$$

$$Cc_s = 0 \quad \text{(C-117)}$$

$$Co_s = 0 \quad \text{(C-118)}$$

$$0 \le x_{c,s} \le y_s \quad , \quad \forall c \in \mathbf{C} \quad \text{(C-119)}$$

$$h^{Lo} \cdot y_s \le h_s^{S1} \le h^{Up} \cdot y_s \quad \text{(C-120)}$$

$$s^{Lo} \cdot y_s \le s_s^{S1} \le s^{Up} \cdot y_s \quad \text{(C-121)}$$

$$H^{Lo} \cdot y_s \le H_s^{S1} \le H^{Up} \cdot y_s \quad \text{(C-122)}$$

$$S^{Lo} \cdot y_s \le S_s^{S1} \le S^{Up} \cdot y_s \quad \text{(C-123)}$$

$$T_{\mathbf{S}}^{Lo} \cdot y_s \le T_s^{S1} \le T_{\mathbf{S}}^{Up} \cdot y_s \quad \text{(C-124)}$$

$$P_{\mathbf{S}}^{Lo} \cdot y_s \le P_s^{S1} \le P_{\mathbf{S}}^{Up} \cdot y_s \quad \text{(C-125)}$$

C.2.3 General unit operation model

Taking the model presented in section B.3, for a general unit $\left(u \in \mathbf{U}^{\mathrm{G}}\right)$, the reformulation results in:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{C-126}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + \sum_{l \in \{1,\ldots,L\}} Q_{l,u} + \sum_{o \in \{1,\ldots,O\}} W_{o,u} \tag{C-127}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{C-128}$$

$$P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{C-129}$$

$$fr_p \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p = F_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{C-130}$$

$$F_{c,p'} = \phi_{c,p'} \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') \neq 1, \forall c \in \mathbf{C} \tag{C-131}$$

$$\phi_{c,p'} = \mathrm{f}_{c,p'}^{\phi,\mathrm{G}}\left(\left[x_{c,p}, T_p\right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta} P_u, \left[fr_p\right]_{\forall p \in \mathbf{P}_u^{\mathbf{I}^*}}\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') \neq 1, \forall c \in \mathbf{C} \tag{C-132}$$

$$T_{p'} = \bar{\mathrm{f}}_{p'}^{T,\mathrm{G}}\left(\left[x_{c,p}, T_p\right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta} P_u, \left[fr_p\right]_{\forall p \in \mathbf{P}_u^{\mathbf{I}^*}}, y_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') \neq 1 \tag{C-133}$$

$$Q_{l,u} = q_{l,u} \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p \quad , \quad \forall l \in \{1,\ldots,L\} \tag{C-134}$$

$$q_{l,u} = \bar{\mathrm{f}}_{l,u}^{q,G}\left(\left[x_{c,p}, T_p\right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, \boldsymbol{\psi}_u, P_u, \underline{\Delta} P_u, \left[fr_p\right]_{\forall p \in \mathbf{P}_u^{\mathbf{I}^*}}, y_u\right) \quad , \quad \forall l \in \{1,\ldots,L\} \tag{C-135}$$

$$W_{o,u} = w_{o,u} \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p \quad , \quad \forall l \in \{1,\ldots,L\} \tag{C-136}$$

$$w_{o,u} = \bar{\mathrm{f}}_{o,u}^{w,G}\left(\left[x_{c,p},T_p\right]_{\substack{c\in\mathbf{C}^*\\ p\in\mathbf{P}_u^{\mathbf{I}}}},\boldsymbol{\psi}_u,P_u,\underline{\Delta}P_u,\left[fr_p\right]_{\forall p\in\mathbf{P}_u^{\mathbf{I}^*}},y_u\right) \quad , \quad \forall o\in\{1,\ldots,O\} \tag{C-137}$$

$$Cc_u = \bar{\mathrm{f}}^{Cc,\mathrm{G}}\left(\left[F_{c,p},T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}},\boldsymbol{\psi}_u,P_u,\underline{\Delta}P_u,y_u\right) \tag{C-138}$$

$$Co_u = \bar{\mathrm{f}}^{Co,G}\left(\left[Q_{l,u}\right]_{l\in\{1,\ldots,L\}},\left[W_{o,u}\right]_{o\in\{1,\ldots,O\}},y_u\right) \tag{C-139}$$

$$F_{c,\mathbf{U}}^{Lo}\cdot y_u \le F_{c,p} \le F_{c,\mathbf{U}}^{Up}\cdot y_u \quad , \quad c\in\mathbf{C}, p\in\mathbf{P}_u^{\mathbf{I}} \tag{C-140}$$

$$T_{\mathbf{U}}^{Lo}\cdot y_u \le T_p \le T_{\mathbf{U}}^{Up}\cdot y_u \quad , \quad p\in\mathbf{P}_u^{\mathbf{I}} \tag{C-141}$$

$$\boldsymbol{\psi}_{\mathbf{U}}^{Lo}\cdot y_u \le \boldsymbol{\psi}_u \le \boldsymbol{\psi}_{\mathbf{U}}^{Up}\cdot y_u \tag{C-142}$$

$$P_{\mathbf{U}}^{Lo}\cdot y_u \le P_u \le P_{\mathbf{U}}^{Up}\cdot y_u \quad , \quad p\in\mathbf{P}_u^{\mathbf{I}} \tag{C-143}$$

$$\underline{\Delta}P_{\mathbf{U}}^{Lo}\cdot y_u \le \underline{\Delta}P_u \le \underline{\Delta}P_{\mathbf{U}}^{Up}\cdot y_u \tag{C-144}$$

$$0 \le fr_p \le y_u \quad , \quad \forall p\in\mathbf{P}_u^{\mathbf{I}} \tag{C-145}$$

Appendix D

DETAILED UNIT OPERATION MODELS

This appendix presents models for common type of operations used in chemical engineering. In this work, process units are divided in 4 groups: inactive units (i.e. raw material source units, final product sink units and mixer-splitters) which have no conditioning or processing function, conditioning units (i.e. heater-coolers, compressor-expanders and expansion valves), separators (i.e. membrane separators, flash vessels, and absorption columns, extraction columns and distillation columns) and reactors (i.e. stirred tank reactors and plug flow reactors). Both, component flow based models and mol fraction based models are presented.

D.1 Component flow based formulation

D.1.1 Inactive units

Figure 58 includes the graphic representation of inactive units.

Raw material source and final product sink units $\left(u \in \mathbf{U}^{\mathbf{SC}}, u \in \mathbf{U}^{\mathbf{SK}}\right)$*:* No real models are required for these units, since only their port variables are used in the superstructure model. These variables

establish all intensive and extensive characteristics (including the total amount) for every raw material used and every final product generated by the process.

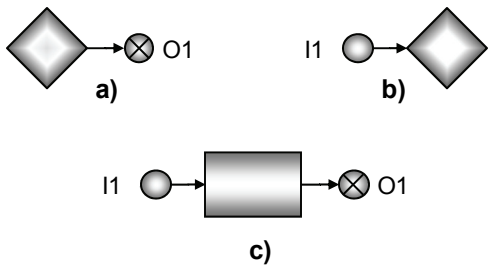


Figure 58: Inactive units. **a)** Source unit. **b)** Sink unit. **c)** Mixer-splitter.

General mixing-splitting units $\left(u \in \mathbf{U}^{\mathbf{MX}}\right)$*:* The highly connected superstructures generated by our methodology can be hectic. Having an operation that combines and split streams can simplify the graphical representation and the analysis of the numerical results. A general adiabatic mixing splitting operation is included here:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-1}$$

$$T_{p'} = T_p \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}}, \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-2}$$

$$P_{p'} = P_p \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}}, \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-3}$$

$$Cc_u = 0 \tag{D-4}$$

$$Co_u = 0 \tag{D-5}$$

Equations (D-1) are component balances. Equation (D-2) is a consequence of the adiabatic and isobaric operation. Equation (D-3) is the isobaric operation condition. Equations (D-4) and (D-5) indicate there are no costs associated to these units.

D.1.2 Conditioning units

Figure 59 includes the graphic representation conditioning units

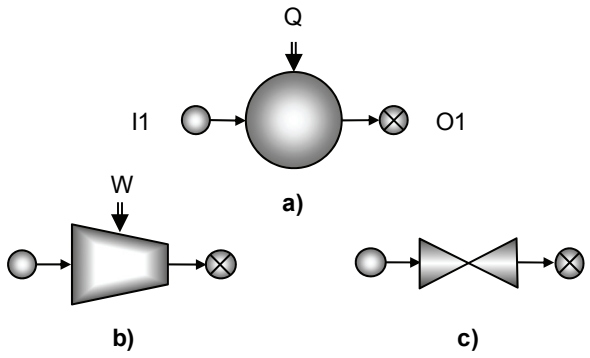


Figure 59: Conditioning units. **a)** Heater-cooler. **b)** Compressor-expander. **c)** Expansion valve.

Heater-Coolers $\left(u \in \mathbf{U}^{\mathbf{HC}}\right)$: These units perform an isobaric temperature change in the materials going through:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-6}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + Q_u \tag{D-7}$$

$$P_u = P_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-8}$$

$$T_u = T_{p'} \quad , \quad P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-9}$$

$$Cc_u = \mathrm{f}^{Cc,HC}\left(Q_u, T_u, P_u\right) \tag{D-10}$$

$$Co_u = \mathrm{f}^{Co,HC}\left(Q_u, T_u\right) \tag{D-11}$$

Equations (D-6) and (D-7) are component and energy balances. Equation (D-8) represents the unit hydraulics (no pressure drop). Equations (D-9) define the unit conditions as those in the outlet port. Equation (D-10) and (D-11) are capital and operative costs models.

Compressor-Expanders $\left(u \in \mathbf{U}^{\mathbf{CE}}\right)$: These units perform an isentropic pressure change in the materials going through, disregarding the phase regime. The following model approximates the behavior of compressors, turbines and pumps.

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-12}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + W_u \tag{D-13}$$

$$P_u = P_p + \Delta P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-14}$$

$$T_u = T_{p'} \quad , \quad P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-15}$$

$$S_{p'} = S_p \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}}, \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-16}$$

$$Cc_u = \mathrm{f}^{Cc,\mathrm{CE}}\left(W_u, T_u, P_u\right) \tag{D-17}$$

$$Co_u = \mathrm{f}^{Co,\mathrm{CE}}\left(W_u\right) \tag{D-18}$$

Equations (D-12) and (D-13) are component and energy balances. Equation (D-14) represents the unit hydraulics. Equations (D-15) define the unit conditions as those in the outlet port. Equation (D-16) is the isentropic operating condition. Equation (D-17) and (D-18) are capital and operative costs models.

Expansion valves $\left(u \in \mathbf{U}^{\mathbf{EV}}\right)$: These units perform an isenthalpic pressure change in the materials going through:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-19}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p \tag{D-20}$$

$$P_u = P_p + \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-21}$$

$$T_u = T_{p'} \quad , \quad P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-22}$$

$$Cc_u = 0 \tag{D-23}$$

$$Co_u = 0 \tag{D-24}$$

Equations (D-19) and (D-20) are component and energy balances. Equation (D-21) represents the unit hydraulics. Equations (D-22) define the unit conditions as those in the outlet port. Equations (D-23) and (D-24) indicate the capital and operative costs for these units are negligible.

D.1.3 Processing units: Separators

Figure 60 shows the graphic representation of common separators.

Membrane separators $\left(u \in \mathbf{U}^{\mathbf{MS}}\right)$: These units perform an adiabatic separation based on the difference in component permeabilities trough a membrane. The component distribution is calculated by a single non-linear mapping.

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-25}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p \qquad \text{(D-26)}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \qquad \text{(D-27)}$$

$$P_u = P_{p'} \quad , \quad T_u = T_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \, , \forall c \in \mathbf{C} \qquad \text{(D-28)}$$

$$F_{c,p'} = \mathrm{f}_c^{F,MS}\left(\left[F_{c,p} \right]_{\substack{c \in \mathbf{C} \\ p' \in \mathbf{P}_u^{\mathbf{I}}}}, T_u, P_u, \underline{\Delta} P_u, Am_u \right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \, / \, ord(p') = 1, \forall c \in \mathbf{C} \qquad \text{(D-29)}$$

$$Cc_u = \mathrm{f}^{Cc,MS}\left(T_u, P_u, Am_u \right) \qquad \text{(D-30)}$$

$$Co_u = 0 \qquad \text{(D-31)}$$

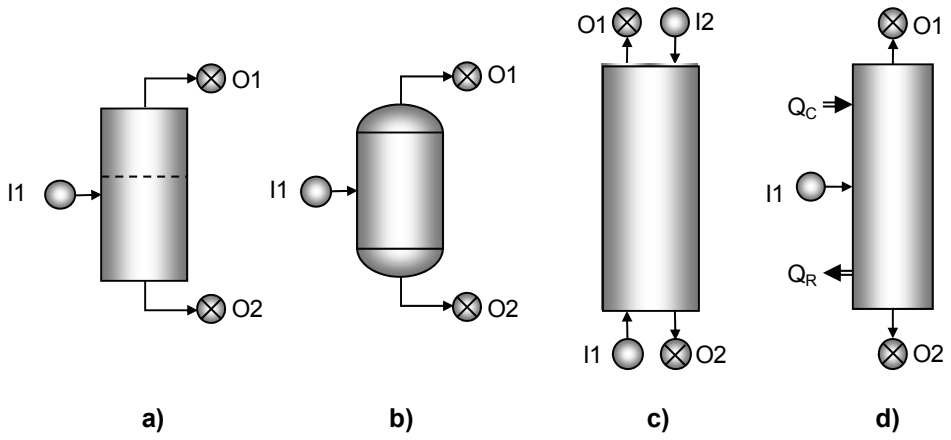


Figure 60: Separation units. **a)** Membrane separator. **b)** Flash vessel. **c)** Absorption/ extraction column. **d)** Distillation column.

To simplify the model, it is assumed that both the *permeate* (the fraction of material crossing the membrane) and the *retentate* (the fraction of material not crossing the membrane) leave the equipment at the same temperature and pressure.

Equations (D-25) and (D-26) are component and energy balances respectively. Equation (D-27) represents the unit hydraulics. Equations (D-28) are the pressure and temperature equilibrium conditions. Equation (D-29) includes a nonlinear mapping to calculate permeate component flows, where $\underline{\Delta}P_u$ is the pressure drop through the membrane and Am_u is the membrane area. It is important to notice these mappings are built for a certain set of components, and a certain membrane material and thickness (i.e. fixed component permeabilities). Equations (D-30) and (D-31) are capital and operative cost models.

Flash vessel units $\left(u \in \mathbf{U}^{\mathbf{FV}}\right)$: These units perform an adiabatic and a near isobaric LV phase separation. The component distribution is calculated using a single non-linear mapping.

$$\sum_{\forall p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{\forall p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-32}$$

$$\sum_{\forall p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{\forall p \in \mathbf{P}_u^{\mathbf{I}}} H_p \tag{D-33}$$

$$P_u = P_p - \underline{\Delta}P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-34}$$

$$P_u = P_{p'} \quad , \quad T_u = T_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-35}$$

$$F_{c,p'} = \mathrm{f}_c^{F,FV}\left(\left[F_{c,p}\right]_{\substack{c \in \mathbf{C} \\ p' \in \mathbf{P}_u^{\mathbf{I}}}}, T_u, P_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord\left(p'\right) = 1, \forall c \in \mathbf{C} \tag{D-36}$$

$$Cc_u = \mathrm{f}^{Cc,FV}\left(\left[F_{c,p}\right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, T_u, P_u\right) \tag{D-37}$$

$$Co_u = 0 \tag{D-38}$$

Equations (D-32) and (D-33) are component and energy balances. Equation (D-34) represents the unit hydraulics. Equations (D-35) are the thermal and mechanical equilibrium conditions for the mixtures leaving the unit. Equation (D-36) includes a nonlinear mapping to calculate vapor product component flows. Equations (D-37) and (D-38) are capital and operative cost models.

Absorption columns $\left(u \in \mathbf{U}^{\mathbf{AC}}\right)$*:* These units are composed by a stack of vapor-liquid equilibrium (VLE) stages. They perform adiabatic mass transfer between two process streams. For simplicity, we assume isobaric operation. Such absorber can be modeled as a stack of flash vessels, but this approach increases significantly the total number of nonlinear mappings, and requires fixing the number of stages or use activation-deactivation binaries, which increases even further the model complexity. Alternatively, a model can be built for the entire unit resulting in the following:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-39}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p \tag{D-40}$$

$$P_u = P_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-41}$$

$$P_u = P_{p'} \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{O}} \tag{D-42}$$

$$F_{c,p'} = \mathrm{f}_{c,p',u}^{F,AC}\left(\left[F_{c,p}, T_p\right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, N_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord\left(p'\right) = 1, \forall c \in \mathbf{C} \tag{D-43}$$

$$T_{p'} = \mathrm{f}_{p',u}^{T,AC}\left(\left[F_{c,p}, T_p\right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, N_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord\left(p'\right) = 1 \tag{D-44}$$

$$Cc_u = \mathrm{f}^{Cc,AC}\left(\left[F_{c,p}, T_p\right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, N_u\right) \tag{D-45}$$

$$Co_u = 0 \qquad \text{(D-46)}$$

Equations (D-39) and (D-40) are component and energy balances respectively. Equation (D-41) represents the unit hydraulics (no pressure drop). Equation (D-42) is the internal pressure balance condition. Equations (D-43) and (D-44) include nonlinear mappings to calculate the vapor product component flows and temperature, where N_u is the number of separation stages. Equations (D-45) and (D-46) are capital and operative cost models.

The main difficulty in this model comes from the high dimension of its domain (i.e. the number of independent variables) which is almost twice that of models for units with only one inlet port. In order to maintain the accuracy of the mappings with such high dimensionality, it is necessary to build specific nonlinear mapping for each unit instance (i.e. $\mathrm{f}_{c,p',u}^{F,AC}(\cdot)$, $\mathrm{f}_{p',u}^{T,AC}(\cdot)$ instead of $\mathrm{f}_{c,p'}^{F,AC}(\cdot)$, $\mathrm{f}_{p'}^{T,AC}(\cdot)$), each covering a small domain around the current iterate produced by the optimization algorithm. As suggested by other authors [52],[53], this requires the use of a calculation loop where the optimization algorithm is run in along with proper subroutines which automatically update the unit model domains so they can be re-sampled and their mappings rebuilt before the next set of optimization calculations.

Extraction columns $\left(u \in \mathbf{U}^{\mathbf{EC}}\right)$*:* These units are composed by a stack of Liquid-Liquid-Equilibrium (LLE) stages. They perform an adiabatic mass transfer between two process streams. Isobaric operation is also assumed for simplicity. The model is identical to that of the absorption columns, but the phase regime is different and accounted for by the non-linear mappings.

Distillation columns $\left(u \in \mathbf{U}^{\mathbf{DC}}\right)$*:* These units are composed by a number of VLE stages, coupled with two heating-cooling devises providing recycle streams to the main mass transfer sections. In fact,

these units can be conceived as arrangements of interconnected flash and heating-cooling units; or ensembles of a feed stage (flash unit), striping and rectifying sections (absorbers), a total condenser (cooler) and a partial reboiler (heater + flash vessel). To model distillation columns, a possible approach is to combine the models for simpler units, but this comes with the same problems highlighted for the case of absorption columns. Another alternative is to build a model for the entire unit and include the required nonlinear mappings. Again, the dimensionality of the domain is high, and the generation of the nonlinear mapping functions could require an iterative sampling-fitting process limited to small domains for every unit instance. The proposed model is as follows:

$$\sum_{p'\in\mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-47}$$

$$\sum_{p'\in\mathbf{P}_u^{\mathbf{O}}} H_{p'} + Qc_u = \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} H_p + Qr_u \tag{D-48}$$

$$P_u = P_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-49}$$

$$P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-50}$$

$$T_{p'} = Tcond_u \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') = 1 \tag{D-51}$$

$$F_{c,p'} = \mathrm{f}_{c,p',u}^{F,DC}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C} \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \quad , \quad \forall c \in \mathbf{C}, \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') = 1 \tag{8-52}$$

$$Qc_u = \mathrm{f}_u^{Qc,DC}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C} \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \tag{D-53}$$

$$Qr_u = \mathrm{f}_u^{Qr,DC}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C} \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \tag{D-54}$$

$$Cc_u = \mathrm{f}^{Cc,DC}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u\right) \quad \text{(D-55)}$$

$$Co_u = \mathrm{f}^{Co,DC}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}\\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u\right) \quad \text{(D-56)}$$

Equations (D-47) and (D-48) are component and energy balances, where Qc_u and Qr_u are the unit condenser and reboilers duties. Equation (D-49) represents the unit hydraulics. Equation (D-50) is the internal pressure balance condition. Equation (D-51) defines the condenser temperature $Tcond_u$, an important internal operating variable. Equation (8-52) includes a mapping to calculate the component flows for the distillate, where Nr_u and Ns_u, are the number of rectifying and stripping stages, and Br_u is the boil-up ratio. Equations (D-53) and (D-54) include mappings to calculate reboiler and condenser duties. Equations (D-55) and (D-56) are capital and operative cost models.

D.1.4 Processing units: Reactors

Figure 61 includes the graphic representation of common reactors.

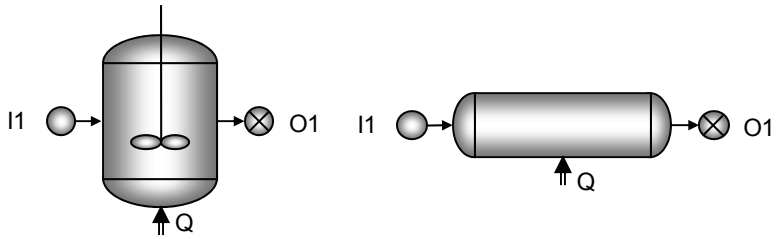


Figure 61: Reactors. **a)** Stirred tank reactor. **b)** Plug flow reactor.

Stirred tank reactors $\left(u \in \mathbf{U}^{\mathbf{SR}}\right)$: These units perform an isothermal and near isobaric reaction operation.

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} + \sum_{r \in \mathbf{R}_u} \nu_{c,r} \cdot \chi_{r,u} \quad , \quad \forall c \in \mathbf{C} \tag{D-57}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + Q_u \tag{D-58}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-59}$$

$$T_u = T_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-60}$$

$$T_u = T_{p'} \quad , \quad P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-61}$$

$$\chi_{r,u} = \mathrm{f}_{r,u}^{\chi,SR}\left(\left[F_{c,p} \right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, T_u, P_u, \underline{\Delta} P_u, V_u \right) \quad , \quad \forall r \in \mathbf{R}_u \tag{D-62}$$

$$Cc_u = \mathrm{f}^{Cc,SR}\left(V_u, T_u, P_u \right) \tag{D-63}$$

$$Co_u = \mathrm{f}^{Co,SR}\left(Q_u, T_u \right) \tag{D-64}$$

Equations (D-57) and (D-58) are component and energy balances respectively, where $\nu_{c,r}$ are stoichiometric coefficients (parameter) and $\chi_{r,u}$ are reaction extent rates. Equation (D-59) is the unit hydraulics. Equations (D-60) and (D-61) enforce isothermal operation and defines the unit conditions as those in the outlet port. Equation (D-62) includes a nonlinear mapping to calculate reaction extent rates. Equations (D-63) and (D-64) are capital and operative cost models.

Plug flow reactors $\left(u \in \mathbf{U}^{\mathbf{PR}} \right)$: The basic characteristics of these units are identical to those in stirred tank reactors, except for the mixing pattern. The model for plug flow reactors has the same form used for stirred tank reactors; however, differences caused by a different mixing pattern are accounted for in the reaction extent mapping.

D.2 Mol fraction based formulation

D.2.1 Inactive units

Raw material source and final product sink units $\left(u \in \mathbf{U}^{\mathbf{SC}}, u \in \mathbf{U}^{\mathbf{SK}}\right)$*:* As in the previous case, no models are required here. Only the port variables included in the port models are useful

General mixing-splitting units $\left(u \in \mathbf{U}^{\mathbf{MX}}\right)$*:* Since no mappings are used in this unit model, the mol fraction based formulation is identical to the molar flow based formulation (D-1)- (D-5).

D.2.2 Conditioning units

Heater-Coolers, Compressor-Expanders and Expansion valves $\left(u \in \mathbf{U}^{\mathbf{HC}}, u \in \mathbf{U}^{\mathbf{CE}}, u \in \mathbf{U}^{\mathbf{EV}}\right)$: Since no mappings in terms of mol fractions are used here, the mol fraction based formulations are identical to the mol flow based formulations (D-6)-(D-11), (D-12)-(D-18), and (D-19)-(D-24).

D.2.3 Processing units: Separators

Membrane separators $\left(u \in \mathbf{U}^{\mathbf{MS}}\right)$: An alternative formulation for (D-25)- (D-31) is:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-65}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p \tag{D-66}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-67}$$

$$P_u = P_{p'} \quad , \quad T_u = T_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}}, \forall c \in \mathbf{C} \tag{D-68}$$

$$F_{c,p'} = N_{c,u} \cdot Am_u \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord\left(p'\right) = 1, \forall c \in \mathbf{C} \tag{D-69}$$

$$AFr_u \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p = Am_u \tag{D-70}$$

$$N_{c,u} = \mathrm{f}_c^{F,MS}\left(\left[x_{c,p} \right]_{p' \in \mathbf{P}_u^{\mathbf{I}}}^{c \in \mathbf{C}^*}, AFr_u, T_u, P_u, \underline{\Delta} P_u \right) \quad , \quad \forall c \in \mathbf{C} \tag{D-71}$$

$$Cc_u = \mathrm{f}^{Cc,MS}\left(Am_u, T_u, P_u \right) \tag{D-72}$$

$$Co_u = 0 \tag{D-73}$$

Equations (D-65)-(D-68) and (D-72)-(D-73) are the same (D-25)-(D-28) and (D-30)-(D-31). Equation (D-69) calculates the permeate flows as the product of component fluxes $N_{c,p'}$ and membrane area. Equation (D-70) defines the total area flow ratio for the equipment. Equation (D-71) uses a mapping to calculate the permeate fluxes, and equations (D-72)-(D-73) are cost models.

Flash vessel units $\left(u \in \mathbf{U}^{\mathbf{FV}} \right)$: An alternative formulation for (D-32)-(D-38) is:

$$\sum_{\forall p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{\forall p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-74}$$

$$\sum_{\forall p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{\forall p \in \mathbf{P}_u^{\mathbf{I}}} H_p \tag{D-75}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-76}$$

$$P_u = P_{p'} \quad , \quad T_u = T_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-77}$$

$$F_{c,p'} = \phi_{c,u} \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord\left(p' \right) = 1, \forall c \in \mathbf{C} \tag{D-78}$$

$$\phi_{c,u} = \mathrm{f}_c^{\phi,FV}\left(\left[x_{c,p}\right]_{p'\in\mathbf{P}_u^{\mathbf{I}}}^{c\in\mathbf{C}^*}, T_u, P_u\right) \quad , \quad \forall c \in \mathbf{C} \tag{D-79}$$

$$Cc_u = \mathrm{f}^{Cc,FV}\left(\left[F_{c,p}\right]_{p\in\mathbf{P}_u^{\mathbf{I}}}^{c\in\mathbf{C}}, T_u, P_u\right) \tag{D-80}$$

$$Co_u = 0 \tag{D-81}$$

Equations (D-74)-(D-77) and (D-80)-(D-81) are the same as (D-32)-(D-35)and (D-37)-(D-38). Equation (D-78) calculates the vapor component flows based on the split fractions calculated using a non-linear mapping in equation (D-79).

Simple absorption columns $\left(u \in \mathbf{U}^{\mathbf{AC}}\right)$*:* An alternative formulation for (D-39)-(D-46) is:

$$\sum_{p'\in\mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-82}$$

$$\sum_{p'\in\mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} H_p \tag{D-83}$$

$$P_u = P_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-84}$$

$$P_u = P_{p'} \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{O}} \tag{D-85}$$

$$F_{c,p'} = \phi_{c,u} \cdot \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord(p') = 1, \forall c \in \mathbf{C} \tag{D-86}$$

$$fr_u \cdot \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_p = \sum_{p\in\mathbf{P}_u^{\mathbf{I}} \,/\, ord(p)=1} F_p \tag{D-87}$$

$$\phi_{c,u} = \mathrm{f}_{c,u}^{\phi,AC}\left(\left[x_{c,p}, T_p\right]_{p\in\mathbf{P}_u^{\mathbf{I}}}^{c\in\mathbf{C}^*}, P_u, N_u, fr_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \,/\, ord(p') = 1, \forall c \in \mathbf{C} \tag{D-88}$$

$$T_{p'} = \mathrm{f}_{p',u}^{T,AC}\left(\left[x_{c,p}, T_p\right]_{\substack{c\in\mathbf{C}^* \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, N_u, fr_u\right) \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \;/\; ord(p') = 1 \tag{D-89}$$

$$Cc_u = \mathrm{f}^{Cc,AC}\left(\left[F_{c,p}, T_p\right]_{\substack{c\in\mathbf{C} \\ p\in\mathbf{P}_u^{\mathbf{I}}}}, P_u, N_u\right) \tag{D-90}$$

$$Co_u = 0 \tag{D-91}$$

Equations (D-82)-(D-85) and (D-90)-(D-91) are the same as (D-39)-(D-42) and equations (D-45)-(D-46). Equation (D-86) calculates the vapor product component flows based on component split fractions estimated by a nonlinear mapping in (D-88). Equation (D-87) defines the fraction of material entering though inlet port 1. Equation (D-89) includes a nonlinear mapping to calculate the vapor product temperature. Notice that the dimension of the domain for mappings (D-88), (D-89) is equal to the one in the component flow based formulation. The only remaining advantage of this formulation regards how simple it is to bound variables $x_{c,p}, fr_u$.

Simple extraction columns $\left(u \in \mathbf{U}^{\mathbf{EC}}\right)$*:* The model has the same form used previously for absorption columns.

Simple distillation columns $\left(u \in \mathbf{U}^{\mathbf{DC}}\right)$*:* An alternative formulation for (D-47)-(D-56) is:

$$\sum_{p'\in\mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall c \in \mathbf{C} \tag{D-92}$$

$$\sum_{p'\in\mathbf{P}_u^{\mathbf{O}}} H_{p'} + Qc_u = \sum_{p\in\mathbf{P}_u^{\mathbf{I}}} H_p + Qr_u \tag{D-93}$$

$$P_u = P_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-94}$$

$$P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-95}$$

$$T_{p'} = Tcond_u \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') = 1 \tag{D-96}$$

$$F_{c,p'} = \phi_{c,u} \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} / ord(p') = 1, \forall c \in \mathbf{C} \tag{D-97}$$

$$Qc_u = qc_u \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p \tag{D-98}$$

$$Qr_u = qr_u \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p \tag{D-99}$$

$$\phi_{c,u} = \mathrm{f}_{c,u}^{\phi,DC}\left(\left[x_{c,p}, T_p \right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \quad , \quad \forall c \in \mathbf{C} \tag{D-100}$$

$$qc_u = \mathrm{f}_u^{qc,DC}\left(\left[x_{c,p}, T_p \right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \tag{D-101}$$

$$qc_u = \mathrm{f}_u^{qr,DC}\left(\left[x_{c,p}, T_p \right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \tag{D-102}$$

$$Cc_u = \mathrm{f}^{Cc,DC}\left(\left[F_{c,p}, T_p \right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \tag{D-103}$$

$$Co_u = \mathrm{f}^{Co,DC}\left(\left[F_{c,p}, T_p \right]_{\substack{c \in \mathbf{C} \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, P_u, Nr_u, Ns_u, Tcond_u, Br_u \right) \tag{D-104}$$

Equations (D-92)-(D-96) and (D-103)-(D-104) are the same (D-47)-(D-51) and (D-55)-(D-56). Equations (D-97), (D-98) and (D-99) calculate distillate component flows, reboiler duty and condenser duty using split fractions and specific duties estimated using the nonlinear mappings in equations (D-100), (D-101) and (D-102).

D.2.4 Processing units: Reactors

Stirred tank reactors $\left(u \in \mathbf{U}^{\mathbf{SR}}\right)$: An alternative formulation for (D-57)-(D-64) is:

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} F_{c,p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_{c,p} + \sum_{r \in \mathbf{R}_u} \nu_{c,r} \cdot \chi_{r,u} \quad , \quad \forall c \in \mathbf{C} \tag{D-105}$$

$$\sum_{p' \in \mathbf{P}_u^{\mathbf{O}}} H_{p'} = \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} H_p + Q_u \tag{D-106}$$

$$P_u = P_p - \underline{\Delta} P_u \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-107}$$

$$T_u = T_p \quad , \quad \forall p \in \mathbf{P}_u^{\mathbf{I}} \tag{D-108}$$

$$T_u = T_{p'} \quad , \quad P_u = P_{p'} \quad , \quad \forall p' \in \mathbf{P}_u^{\mathbf{O}} \tag{D-109}$$

$$\chi_{r,u} = \bar{r}_{r,u} \cdot V_u \quad , \quad \forall r \in \mathbf{R}_u \tag{D-110}$$

$$VFr_u \cdot \sum_{p \in \mathbf{P}_u^{\mathbf{I}}} F_p = V_u \tag{D-111}$$

$$\bar{r}_{r,u} = \mathrm{f}_{r,u}^{\bar{r},SR}\left(\left[x_{c,p} \right]_{\substack{c \in \mathbf{C}^* \\ p \in \mathbf{P}_u^{\mathbf{I}}}}, VFr_u, T_u, P_u, \underline{\Delta} P_u \right) \quad , \quad \forall r \in \mathbf{R}_u \tag{D-112}$$

$$Cc_u = \mathrm{f}^{Cc,SR}\left(V_u, T_u, P_u\right) \tag{D-113}$$

$$Co_u = \mathrm{f}^{Co,SR}\left(Q_u, T_u\right) \tag{D-114}$$

Equations (D-105)-(D-109) and (D-113)-(D-114) are the same (D-57)-(D-61) and (D-63)-(D-64). Equation (D-110) calculates the reaction extent rates based on average rates of reaction estimated using nonlinear mappings in (D-112). Finally (D-111) defines the volume flow ratio VFr_u.

Plug flow reactors $\left(u \in \mathbf{U}^{\mathbf{PR}}\right)$: As in the molar flow based formulation, the form of the model for plug flow reactors is the same used for stirred tank reactors.

REFERENCES

[1] Douglas JM (1985) A Hierarchical Decision Procedure for Process Synthesis. AICHE J. 31(3):353-362.

[2] Biegler LT, Grossmann IE, & Westerberg AW (1997) Systematic methods of chemical process design (Prentice Hall PTR, Upper Saddle River, N.J.) pp xviii, 796 p.

[3] Kokossis AC & Floudas CA (1994) Optimization of Complex Reactor Networks .2. Nonisothermal Operation. Chem Eng Sci 49(7):1037-1051.

[4] Lakshmanan A & Biegler LT (1996) Synthesis of optimal chemical reactor networks. Ind Eng Chem Res 35(4):1344-1353.

[5] Aggarwal A & Floudas CA (1990) Synthesis of General Distillation Sequences - Nonsharp Separations. Comput Chem Eng 14(6):631-653.

[6] Bauer MH & Stichlmair J (1996) Superstructures for the mixed integer optimization of nonideal and azeotropic distillation processes. Comput Chem Eng 20:S25-S30.

[7] Heckl I, Kovacs Z, Friedler F, Fan LT, & Liu J (2007) Algorithmic synthesis of an optimal separation network comprising separators of different classes. Chem. Eng. Process. 46(7):656-665.

[8] Floudas CA & Grossmann IE (1987) Automatic-Generation of Multiperiod Heat-Exchanger Network Configurations. Comput Chem Eng 11(2):123-142.

[9] Chvátal V (1983) Linear programming (W.H. Freeman, New York) pp xiii, 478 p.

[10] Wolsey LA (1998) Integer programming (Wiley, New York) pp xviii, 264 p.

[11] Nocedal J & Wright SJ (2006) Numerical optimization (Springer, New York) 2nd Ed pp xxii, 664 p.

[12] Gupta OK & Ravindran V (1985) Branch-and-Bound Experiments In Convex Nonlinear Integer Programming. Mgmt Sc 31(12):1533-1546.

[13] Borchers B & Mitchell JE (1994) An Improved Branch-and-Bound Algorithm for Mixed-Integer Nonlinear Programs. Comput Oper Res 21(4):359-367.

[14] Geoffrion AM (1972) Generalized Benders Decomposition. J Optim Theory Appl 10(4):237-260.

[15] Duran MA & Grossmann IE (1986) An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. Math Program 36(3):307-339.

[16] Quesada I & Grossmann IE (1992) An Lp/Nlp Based Branch and Bound Algorithm for Convex Minlp Optimization Problems. Comput Chem Eng 16(10-11):937-947.

[17] Tawarmalani M & Sahinidis NV (2002) Convexification and global optimization in continuous and mixed-integer nonlinear programming : theory, algorithms, software, and applications (Kluwer Academic Publishers, Dordrecht Boston) pp xxv, 475 p.

[18] Kocis GR & Grossmann IE (1989) A Modeling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets. Comput Chem Eng 13(7):797-819.

[19] Yeomans H & Grossmann IE (1999) A systematic modeling framework of superstructure optimization in process synthesis. Comput Chem Eng 23(6):709-731.

[20] Kondili E, Pantelides CC, & Sargent RWH (1993) A General Algorithm for Short-Term Scheduling of Batch-Operations .1. Milp Formulation. Comput Chem Eng, 17(2):211-227.

[21] Smith EMB & Pantelides CC (1995) Design of Reaction Separation Networks Using Detailed Models. Comput. Chem. Eng. 19:S83-S88.

[22] Friedler F, Tarjan K, Huang YW, & Fan LT (1992) Graph-Theoretic Approach to Process Synthesis - Axioms and Theorems. Chem. Eng. Sci. 47(8):1973-1988.

[23] Friedler F, Tarjan K, Huang YW, & Fan LT (1993) Graph-Theoretic Approach to Process Synthesis - Polynomial Algorithm for Maximal Structure Generation. Comput. Chem. Eng. 17(9):929-942.

[24] Farkas T, Rev E, & Lelkes Z (2005) Process flowsheet superstructures: Structural multiplicity and redundancy part I: Basic GDP and MINLP representations. Comput. Chem. Eng. 29(10):2180-2197.

[25] Farkas T, Rev E, & Lelkes Z (2005) Process flowsheet superstructures: Structural multiplicity and redundancy Part II: Ideal and binarily minimal MINLP representations. Comput. Chem. Eng. 29(10):2198-2214.

[26] Achenie LKE & Biegler LT (1990) A Superstructure Based Approach to Chemical Reactor Network Synthesis. Comput. Chem. Eng. 14(1):23-40.

[27] Kokossis AC & Floudas CA (1994) Optimization of Complex Reactor Networks .2. Nonisothermal Operation. Chem Eng Sci 49(7):1037-1051.

[28] Lakshmanan A & Biegler LT (1996) Synthesis of optimal chemical reactor networks. Ind Eng Chem Res 35(4):1344-1353.

[29] Schweiger CA & Floudas CA (1999) Optimization framework for the synthesis of chemical reactor networks. Ind Eng Chem Res 38(3):744-766.

[30] Yee TF & Grossmann IE (1990) Simultaneous-Optimization Models for Heat Integration .2. Heat-Exchanger Network Synthesis. Comput Chem Eng 14(10):1165-1184.

[31] Floudas CA, Ciric AR, & Grossmann IE (1986) Automatic Synthesis of Optimum Heat-Exchanger Network Configurations. Aiche J 32(2):276-290.

[32] Floudas CA & Paules GE (1988) A Mixed-Integer Nonlinear-Programming Formulation for the Synthesis of Heat-Integrated Distillation Sequences. Comput Chem Eng 12(6):531-546.

[33] Novak Z, Kravanja Z, & Grossmann IE (1996) Simultaneous synthesis of distillation sequences in overall process schemes using an improved MINLP approach. Comput Chem Eng 20(12):1425-1440.

[34] Ostrovskii GM, Ziyatdinov NN, & Borisevich TV (1997) Synthesis of chemical process flowsheets by a modified structural parameter method. Theor. Found. Chem. Eng. 31(1):78-86.

[35] Lee S & Grossmann IE (2003) Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks. Comput Chem Eng 27(11):1557-1575.

[36] Raman R & Grossmann IE (1994) Modeling and Computational Techniques for Logic-Based Integer Programming. Comput Chem Eng 18(7):563-578.

[37] Turkay M & Grossmann IE (1996) Logic-based MINLP algorithms for the optimal synthesis of process networks. Comput Chem Eng 20(8):959-978.

[38] Lee S & Grossmann IE (2000) New algorithms for nonlinear generalized disjunctive programming. Comput Chem Eng 24(9-10):2125-2141.

[39] Lee S & Grossmann IE (2001) A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. Comput Chem Eng 25(11-12):1675-1697.

[40] Vecchietti A & Grossmann IE (1999) LOGMIP: a disjunctive 0-1 non-linear optimizer for process system models. Comput Chem Eng 23(4-5):555-565.

[41] Aggarwal A & Floudas CA (1990) Synthesis of General Distillation Sequences - Nonsharp Separations. Comput. Chem. Eng. 14(6):631-653.

[42] Caballero JA & Grossmann IE (2001) Generalized disjunctive programming model for the optimal synthesis of thermally linked distillation columns. Ind Eng Chem Res 40(10):2260-2274.

[43] Yeomans H & Grossmann IE (2000) Disjunctive programming models for the optimal design of distillation columns and separation sequences. Ind Eng Chem Res 39(6):1637-1648.

[44] Xiong Y, Chen W, Apley D, & Ding XR (2007) A non-stationary covariance-based Kriging method for metamodelling in engineering design. Int J Numer Meth Eng 71(6):733-756.

[45] Shao T, Krishnamurty S, & Wilmes GC (2007) Preference-based surrogate Modeling in engineering design. Aiaa J 45(11):2688-2701.

[46] Won KS & Ray T (2005) A framework for design optimization using surrogates. Eng Optimiz 37(7):685-703.

[47] Palmer K & Realff M (2002) Metamodeling approach to optimization of steady-state flowsheet simulations - Model generation. Chem Eng Res Des 80(A7):760-772.

[48] Mujtaba IM, Aziz N, & Hussain MA (2006) Neural network based modeling and control in batch reactor. Chem Eng Res Des 84(A8):635-644.

[49] Fernandez FAN (2006) Optimization of Fischer-Tropsch synthesis using neural networks. Chem Eng Technol 29(4):449-453.

[50] Serra JM, Corma A, Argente E, Valero S, & Botti V (2003) Neural networks for modeling of kinetic reaction data applicable to catalyst scale up and process control and optimization in the frame of combinatorial catalysis. Appl Catal a-Gen 254(1):133-145 (in English).

[51] Caballero JA, Odjo A, & Grossmann IE (2007) Flowsheet optimization with complex cost and size functions using process simulators. AICHE J. 53(9):2351-2366.

[52] Caballero JA & Grossmann IE (2008) An algorithm for the use of surrogate models in modular flowsheet optimization. AICHE J. 54(10):2633-2650.

[53] Davis E & Ierapetritou M (2008) A kriging-based approach to MINLP containing black-box models and noise. Ind Eng Chem Res 47(16):6101-6125.

[54] Wan XT, Pekny JF, & Reklaitis GV (2005) Simulation-based optimization with surrogate models - Application to supply chain management. Comput Chem Eng 29(6):1317-1328.

[55] Douglas JM (1988) Conceptual design of chemical processes (Mc-Graw Hill, New York, N.Y.) pp xvi, 601 p.

[56] Balas E (1974) Disjunctive Programming. Annals of Discrete Mathematics 5(1):3-20.

[57] Balas E (1985) Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. Siam J Algebra Discr 6(3):466-486.

[58] Raman R & Grossmann IE (1992) Integration of Logic and Heuristic Knowledge in MINLP Optimization for Process Synthesis. Comput. Chem. Eng. 16(3):155-171.

[59] Raman R & Grossmann IE (1993) Symbolic-Integration of Logic in Mixed-Integer Linear-Programming Techniques for Process Synthesis. Comput Chem Eng 17(9):909-927.

[60] Lee S & Grossmann IE (2001) New algorithms for nonlinear generalized disjunctive programming (vol 24, pg 2125, 2000). Comput Chem Eng 25(7-8):1153-1153.

[61] Grossmann IE & Lee S (2003) Generalized convex disjunctive programming: Nonlinear convex hull relaxation. Comput Optim Appl 26(1):83-100.

[62] Sawaya NW & Grossmann IE (2007) Computational implementation of non-linear convex hull reformulation. Comput Chem Eng 31(7):856-866.

[63] Bunus P & Fritzson P (2004) Automated static analysis of equation-based components. Simul-T Soc Mod Sim 80(7-8):321-345.

[64] Haykin, S. Artificial Neural Networks: A comprehensive foundation. Upper Saddle River: Prentice Hall, 1999.

[65] Porn R, Harjunkoski I, & Westerlund T (1999) Convexification of different classes of non-convex MINLP problems. Comput Chem Eng 23(3):439-448.

[66] Bjork KM, Lindberg PO, & Westerlund T (2003) Some convexifications in global optimization of problems containing signomial terms. Comput Chem Eng 27(5):669-679.

[67] Mckay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics. 2000;42:55-61.

[68] Pukelsheim F. Optimal design of experiments. Classic ed. Philadelphia: SIAM/Society for Industrial and Applied Mathematics, 2006

[69] Tian GL, Fang HB, Tan M, Qin H, & Tang ML (2009) Uniform distributions in a class of convex polyhedrons with applications to drug combination studies. J Multivariate Anal 100(8):1854-1865.

[70] Demuth H, Beale M, Hagan M. MATLAB Neural Network Toolbox™ 6. Natick: The MathWorks, Inc, 2009.

[71] Nguyen D, Widrow B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. Proceedings of the International Joint Conference on Neural Networks, 3:21–26, 1990.

[72] Guthrie KM (1969) Data and Techniques for Preliminary Capital Cost Estimating. Chem Eng-New York 76(6):114-&.

[73] Smith R (2005) Chemical process design and integration (Wiley, Chichester, West Sussex, England ; Hoboken, NJ) pp xxiii, 687 p.

[74] Turton R (2009) Analysis, synthesis, and design of chemical processes (Prentice Hall, Upper Saddle River, N.J.) 3rd Ed pp xxxiii, 1068 p.

[75] Biegler LT, Grossmann IE, & Westerberg AW (1997) Systematic methods of chemical process design (Prentice Hall PTR, Upper Saddle River, N.J.) pp xviii, 796 p.

[76] Rosenthal R, Brooke, Kendrick, Meeraus A, & Raman (2012) GAMS - A User's Guide (GAMS Development Corporation, Washington, DC, USA) pp xii, 297 p.

[77] Biegler LT, Grossmann IE, & Westerberg AW (1985) A Note on Approximation Techniques Used for Process Optimization. Comput Chem Eng 9(2):201-206.

[78] Alie C, Backham L, Croiset E, & Douglas PL (2005) Simulation of CO2 capture using MEA scrubbing: a flowsheet decomposition method. Energ Convers Manage 46(3):475-487.

[79] Diver RB, Miller JE, Allendorf MD, Siegel NP, & Hogan RE (2008) Solar thermochemical water-splitting ferrite-cycle heat engines. J Sol Energ-T Asme 130(4).

[80] Kim K, Henao CA, Johnson TA, Dedrick DE, Miller JE, Stecheld EB, Maravelias CT (2011) Methanol production from CO2 using solar-thermal energy: process development and techno-economic analysis. Energy & Env Sc. 4(9): 3122-3132.

[81] Neumaier A, Shcherbina O, Huyer W, & Vinko T (2005) A comparison of complete global optimization solvers. Math Program 103(2):335-356.

[82] Westerink EJ & Westerterp KR (1988) Safe Design of Cooled Tubular Reactors for Exothermic Multiple Reactions - Multiple-Reaction Networks. Chem Eng Sci 43(5):1051-1069.

[83] Papoulias SA & Grossmann IE (1983) A Structural Optimization Approach in Process Synthesis .2. Heat-Recovery Networks. Comput Chem Eng 7(6):707-721.