UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA QUÍMICA Y BIOTECNOLOGÍA

DESIGN OF A REAL BIOTECHNOLOGICAL MULTIPRODUCT BATCH PLANT WITH AN OPTIMIZATION BASED APPROACH

TESIS PARA OPTAR AL GRADO DE DOCTORA EN CIENCIAS DE LA INGENIERÍA MENCIÓN QUÍMICA

GABRIELA DANIELA SANDOVAL HEVIA

PROFESORES GUÍA:
JUAN ASENJO DE LEUZE
NICOLÁS FIGUEROA GONZÁLEZ

COMISIÓN:
DANIEL ESPINOZA GONZÁLEZ
MARÍA ELENA LIENQUEO CONTRERAS
LUIS CISTERNAS ARAPIO

Este trabajo es financiado por una beca CONICYT para estudios de doctorado en Chile, los proyectos Fondecyt regular 1110024 y 1150046; y por el Centro Basal financiado por CONICYT CeBiB FB0001

SANTIAGO DE CHILE
MARZO 2016

RESUMEN DE LA TESIS
PARA OPTAR AL GRADO DE DOCTORA EN CIENCIAS DE LA INGENIERÍA, MENCIÓN QUÍMICA
POR: GABRIELA DANIELA SANDOVAL HEVIA
FECHA: MARZO 2016
PROF. GUÍA: SR. JUAN ASENJO DE LEUZE

Productos biotecnológicos, como los biofarmacéuticos entre otros, son productos cuyas tecnologías de producción están en constante desarrollo. Adicionalmente, sus escalas de producción son pequeñas haciendo de las plantas *batch* las más apropiadas para su producción. En particular, plantas *batch* multi-producto permiten la producción de una variedad de productos biotecnológicos con varias etapas en común.

Una forma de modelar el diseño de plantas *batch* multi-producto es mediante el enfoque basado en optimización que fue estudiado por primera vez para este tipo de plantas por Robinson y Lonkar, quienes estudiaron el diseño de este tipo de plantas dimensionando los equipos que la conforman. Pese a los múltiples avances en el área, los que incluyen decisiones como la duplicación de unidades, la disposición de tanques de almacenamiento intermedio, programación de la producción y consideración ambientales, entre otras mejores, aún existe una falta de trabajos donde este tipo de enfoques es aplicado en plantas reales.

En este trabajo se estudia una reformulación Entera-Mixta Lineal (MILP) del problema Entero-Mixto No-Lineal (MINLP) que resulta al plantear el modelo para el diseño de una planta biotecnológica *batch* multi-producto. En un primer paso se estudia una reformulación MILP que permite modelar el diseño de una planta utilizando tamaños de equipos en un conjunto continuo y una selección de *hosts* en un conjunto discreto de opciones. Esta reformulación hace uso de técnicas avanzadas de reformulación, probando ser escalable y confiable para su aplicación en casos reales. En un segundo paso, la reformulación MILP original fue modificada para la inclusión de una selección de equipos, tanto en un conjunto discreto, como en uno continuo, dando un enfoque más realista para poder modelar una planta biotecnológica *batch* multi-producto; donde unidades como los reactores pueden ser construidos de acuerdo con las necesidades del cliente, sin embargo, unidades como las columnas cromatográficas sólo están disponibles en tamaños discretos dados por el proveedor.

Información de procesos reales que formaban parte de una planta *batch* multi-producto real permitieron la determinación de los parámetros del modelo y una comparación entre las distintas lineas de producción versus la planta real mostraron que este tipo de modelos puede permitir grandes ahorros en los costos de los principales equipos de la planta.

Finalmente, como el enfoque estudiado utiliza *software* de modelación y optimización, el modelo es más amigable para quienes puedan utilizarlo en la práctica. Sin embargo niveles más bajos de implementación podrían mejorar los tiempos de resolución permitiendo la inclusión de formulaciones más complejas, como por ejemplo, la inclusión de costos u objetivos de producción variables.

A mi marido por ser mi fuerza,
a mis hijos por ser mi alegría

Agradecimientos

Quiero agradecer a mi profesor guía, Juan Asenjo, por confiar en mí, por su generosidad al permitirme trabajar además con Daniel y Nicolás, y por su apoyo en los momentos menos gratos que me tocó vivir en estos años de doctorado.

A Daniel y Nicolás, por su tiempo de trabajo, ideas y correcciones. Daniel, en particular te agradezco los tiempos que te diste para preguntar un poco más y por esos consejos tanto personales, como para mi futuro académico. No siempre fueron fáciles de escuchar, pero siempre fueron valiosos y me ayudaron a reflexionar.

Agradezco a las lindas personas que conocí en mi paso por el laboratorio. Entre ellos a Pablo que siempre estuvo disponible para ayudarme a entender el lenguaje matemático a veces demasiado inentendible para mí. A quienes me acompañaron en los primeros pasos y estuvieron ahí en los grandes hitos de mi historia en estos 6 años. Cami, Fran, Dani S., Vida, Gianni, Paty y Alicia. Fueron mis amigas y compañeras en parte de este recorrido y guardo bellos recuerdos.

Gracias a mis padres por observar a la distancia y estar siempre prestos a ayudar. A mi suegra por portarse un 10 conmigo. Por darme la tranquilidad para poder trabajar sabiendo que mi Cati estaba siendo bien cuidada y regaloneada.

Finalmente, quiero agradecer a mi marido. Por permitirme trabajar aún a costa de sus propios tiempos. Por la compañía y el apoyo incondicional. Por la confianza y el amor infinito.

Hoy termina esta etapa, pero sé que queda mucho camino por recorrer. Espero seguir encontrando en mi camino personas como ustedes, que me regalaron lindos momentos y un lindo espacio para trabajar.

Gabriela Sandoval Hevia.
Marzo, 2016

Abstract

Biotechnological products such as biopharmaceuticals among others are products which production technologies are in constant development. In addition, their production scales are small making batch plants the most suitable type for their production. In particular, multi-product batch plants allows the production of a variety of biotechnological products with many common steps.

One way to model the design of such plants is the optimization based approach that was first studied in 1972 by Robinson and Lonkar who addressed the equipment sizing of a multi-product batch plant. Despite of the advances in the area, including decisions as duplication of units, allocation of intermediate storage vessels, scheduling and environmental considerations, among other improvements, there is a lack of reported work where this type of approach is applied to real plants.

In this work a Mixed-Integer Linear Programming (MILP) reformulation of the resulted Mixed-Integer Non-Linear problem (MINLP) for the design of a biotechnological multi-product batch plant is studied. In a first step a MILP reformulation that addresses the desing of a plant using continuous equipment sizes and discrete host selection is studied. This reformulation made use of advanced reformulation techniques and proved to be scalable and reliable for its application in real cases. In a second step the former MILP reformulation was modified for the inclusion of the selection of equipment sizes in both, continuous and discrete sizes giving a more realistic approach to model a real biotechnological multi-product batch plant. Items such as reactors may be build according to customer needs, but units such as chromatographic columns are only available in discrete sets of sizes given by manufacturers.

Information from real processes that where part of an actual multi-product batch plant allowed the computation of the model parameters; and a comparison of the optimized facilities versus the actual plant showed that this type of models may achieve great savings in the cost of the main equipment of the plant.

As the studied approach relies on "off the shelve" optimization and modelling software the model is more amiable to practioners. Nevertheless lower implementation levels could improve resolution times allowing for the inclusion of more complex formulations such as the inclusion of variable costs and production target parameters, among others.

Contents

List of Figures

List of Tables

Nomenclature

Indices

h	host
i	product
j	stage
m	number of duplicated units
UP, LO	upper and lower bounds

Sets

$\mathcal{E}^1$	set of batch stages, $\subset \mathcal{E}$
$\mathcal{E}^2$	set of semi-continuous stages, $\subset \mathcal{E}$
$\mathcal{E}^3$	set of chromatographic stages, $\subset \mathcal{E}$
$\mathcal{E}$	set of all stages
$\mathcal{H}$	set of hosts
$\mathcal{I}$	set of products
$\mathcal{M}$	set of available units operating in parallel in-phase or out-of-phase
$\mathcal{R}$	set of routes: stages j needed to process the product i synthesized by host h
$\mathcal{U}$	set of available hosts h for product i synthesis

Parameters

δ	time horizon
c_j^n, γ_j^n	cost coefficients related to Y_j^n with $n \in \{1, 2, 3\}$
d_i	production target for product i
S_{ij}^n, s_{ij}^n	size factor for product i in stage j related to Y_j^n. $s_{ij}^n = \ln\left(S_{ij}^n\right)$ $n \in \{1, 2, 3\}$
T_{ij}^0, t_{ij}^0	time factor for product i in the batch or chromatographic stage j. $t_{ij}^0 = \ln\left(T_{ij}^0\right)$
T_{ij}^1, t_{ij}^1	time factor for product i in the semi-continuous or chromatographic stage j. $t_{ij}^1 = \ln\left(T_{ij}^1\right)$

Variables

v	slack variable
X_j^n, x_j^n	number of units operating in parallel in-phase (X_j^1) and out-of-phase (X_j^2) in stage j. $x_j^n = \ln\left(X_j^n\right)$
Y_j^1, y_j^1	volumetric capacity for batch units and retentate or feed tank for semi-continuous or chromatographic stages. $y_j^1 = \ln\left(Y_j^1\right)$
Y_j^2, y_j^2	volumetric capacity for permeate of product tanks for semi-continuous or chromatographic stages. $y_j^2 = \ln\left(Y_j^2\right)$
Y_j^3, y_j^3	capacity of semi-continuous items. $y_j^3 = \ln\left(Y_j^3\right)$
Y_i^4, y_i^4	batch size for final product i. $y_i^4 = \ln\left(Y_i^4\right)$
Y_i^5, y_i^5	cycle time for product i. $y_i^5 = \ln\left(Y_i^5\right)$
z_{ih}^1	binary variable: 1 if protein i is synthesized by host h; 0 otherwise
z_j^2	binary variable: 1 if stage j is used to process at least one of the products
z_{jm}^3	binary variable: 1 if m units are operating in parallel in-phase in stage j; 0 otherwise
z_{jm}^4	binary variable: 1 if m units are operating in parallel out-of-phase in stage j; 0 otherwise

Chapter 1

Introduction

1.1 Biotechnological industry and batch processes

According to Moreno & Montagna (2007b):

> The batch mode of operation in food and biotechnological industries has received a renewed interest particularly because of the market wich has become more uncertain, complex and competitive.

Batch plants can be easily reorganized to allow for production modifications within the same plant (Barbosa-Póvoa, 2007) being the most common and studied structures the multi-product and the multipurpose batch plants. In multi-product or flowshop plants all products follow the same production path (see Figure 1.1). On the other hand, in multipurpose or jobshop plants different products can be produced by sharing available equipment, raw materials, utilities and production time resources. Major difference with multi-product plants is that different products may be produced in arbitrary sequences and locations (see Figure 1.2).

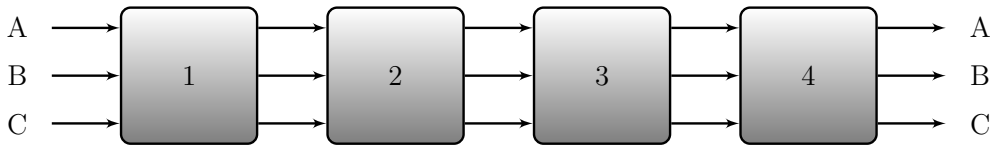


Figure 1.1 – Multi-product/flowshop plant (taken from Biegler et al. (1997)).

This work is focused on multi-product batch plants since the main objective is the application of an optimization based approach to the design of a biotechnological multi-product batch plant using information of real processes that were in fact part of a real multi-product batch plant (Imperatore & Asenjo, 2001).

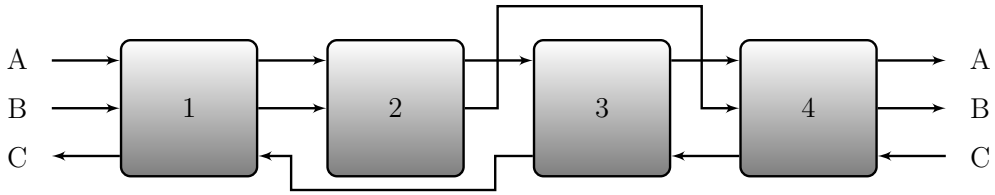


Figure 1.2 – Multipurpose/jobshop plant (taken from Biegler et al. (1997)).

1.2 Multi-product batch plant design with an optimization based approach

The multi-product batch plant design may include a variety of problems such as synthesis, design, production planning, and scheduling (Voudouris & Grossmann, 1993). The main objective is to minimize the production requirements over a defined time horizon (Barbosa-Póvoa, 2007). Voudouris & Grossmann (1993) classified such decisions as follows:

1. Synthesis decisions
   (a) allocation of tasks to equipment
   (b) parallel units operating either in-phase or out-of-phase
   (c) location of intermediate storage
2. Design decisions
   (a) selection of equipment of standard sizes
   (b) sizing of intermediate storage vessels with standard sizes
3. Production planning decisions
   (a) optimal length of production cycle during which the optimal schedule is executed
   (b) levels of inventory of final products
4. Scheduling decisions
   (a) sequencing of products

The optimization based approach for the design of batch plants began with Robinson & Loonkar (1972) who studied the design decision of selection of equipment sizes and since then great progress has been made. In recent years synthesis decisions such as duplication of units in series has been incorporated (Moreno et al., 2009a,b). Corsano et al. (2005) studied synthesis, design and operation decisions simultaneously; Fumero et al. (2011, 2012b) studied the simultaneous design and scheduling of a multi-product batch plant; and authors such as Pinto-Varela et al. (2009) have included demand uncertainty, environmental issues (Wang et al., 2010) and transportation concerns (Yi & Reklaitis, 2011).

According to Iribarren et al. (2004) towards the middle of the past decade despite of the existance of a great amount of work based on expert systems for the synthesis of bioprocesses just few papers that used an optimization based approach had been published; and those that had been published dealt with small portions of the global problem (Montagna et al., 2004). That is the case of Vásquez-Alvarez et al. (2001) that developed a strategy for the synthesis of a purification process based on a variety of chromatographic stages; the MILP model they studied used physico-chemical data of a mixture of proteins. On the other hand, a few years earlier Samsatli & Shah (1996a) studied the design problem for the entire production of a unique product that included a fermentation stage followed by primary separation steps and high resolution stages for a final purification. After this work a second part dealt with scheduling decisions for a more accurate sequencing and timing determination for each operation unit in the plant (Samsatli & Shah, 1996b).

In the year 2000 Montagna et al. began a series of collaborations that studied the design of a biotechnological multi-product batch plant (Asenjo et al., 2000; Pinto et al., 2001) including later synthesis decisions (Iribarren et al., 2004). Former work studied the production of 4 recombinant proteins where 6 steps of separation and purification followed the fermentation process; this process was used a few years later by other authors as an example process to test their own formulations (Dietz et al., 2005; Moreno et al., 2009a). On the other hand, the work of Iribarren et al. (2004) was described by Moreno-Benito et al. (2014) as one of the most relevant contributions to the area at that time because their MINLP formulation addressed the combination of process synthesis decisions -selection of the microorganism responsible of the biological process; and selection of separation and purifucation techniques- plant allocation decisions -operation mode- and plant design decisions such as equipment sizing.

After these papers, advances in biotechnological multi-product batch plants literature have not been as much as those that can be found for chemical plants (see Barbosa-Póvoa (2007) for an extended review); nevertheless most of the advances applied for chemical plants can be also applied for bioprocesses. Among the papers in biotechnology are the work of Srinivasan et al. (2003) who included the uncertainty of the demand in the design of a bioreactor that produces penicillin; Dietz et al. (2005) included environmental considerations to the design of a multi-product batch plant and Moreno & Montagna (2007b) developed a model for the design and scheduling of a plant of 5 stages for a vegetable extraction.

1.3 Elements for the design and synthesis of a biotechnological multi-product batch plant

In this work the design of a biotechnological multi-product batch plant with an optimization based approach took into account design and synthesis decisions which will be explained below.

1.3.1 Synthesis decisions

Synthesis decisions make reference to the configuration of the plant taking into account 3 topics:

Allocation of tasks to equipment: This makes reference to the selection of one technique among two or more that can perform the same downstream processing step.

Parallel units operating either in-phase or out-of-phase: Duplication of units in-phase permits the elimination of bottlenecks due to equipment capacities as different units process the incoming stream simultaneously (Voudouris & Grossmann, 1993). This decision is used when the maximum capacity available is used allowing the processing of batches of bigger sizes. The duplication of units out-of-phase, on the other hand, eliminates the bottlenecks due to cycle times as different units process the incoming stream with different initial times (Voudouris & Grossmann, 1993).

Location of intermediate storage: This decision is highly related to the used storage policy and the objective is to reduce the idle times during production (Galiano & Montagna, 1993). According to Barbosa-Póvoa (2007) 5 policies can be identified:

- **Zero-wait:** the material is unstable and has to be processed immediately.
- **Unlimited intermediate storage:** the material is stable and can be arranged in one or more storage vessels with an unlimited capacity.
- **Finite intermediate storage:** the material is stable and can be arranged in one or more storage vessels with a finite capacity.
- **Shared intermediate storage:** the material is stable and can be arranged in one or more storage vessels that can be shared with other material but not simultaneously.
- **No intermediate storage:** the material is stable but no storage vessels are available. However, it may reside temporarily in the processing equipment where it has been produced.

1.3.2 Design decisions

Design decisions correspond to the sizing of different process equipment which can be selected among a continuous range of sizes or among a discrete set of available sizes. According to Salomone et al. (1994) process equipment can be clasified into 3 types:

Batch intensive: these are the traditional batch stages in which the process material remains in the unit for a time that depends on the process kinetics, that in turn depends on concentration and temperature. Examples of these units are fermenters and reactors.

Semi-continuous: these are the traditional semi-continuous stages that operate between two batch stages. These units operate continuously but intermittently to transfer the material among different batch stages. Examples of these units are heat exchangers.

Batch extensive: these equipments involve both items, batch and semi-continuous. Examples of these units are filters and centrifuges that need feed and product tanks, togheter to the semi-continuous item.

In this work batch and semi-continuous items are sized in stages that are batch intensive and batch extensive.

1.4 Mixed Integer Non-Linear Programming (MINLP) versus Mixed Integer Linear Programming (MILP)

As stated by Grossmann et al. (2000) design decision problems can be written as a Mixed-Integer Non-Linear Problem (MINLP) of the form:

$$\begin{array}{rl} \min Z = & f(x,y) \\ s.t. & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in X, y \in \{0,1\} \end{array} \tag{1.1}$$

where $f(x,y)$ is the objective function (e.g. cost), $h(x,y) = 0$ are equations that describe the performance of the system, such as mass and/or energy balances, $g(x,y) \leq 0$ are inequalities that define specific restrictions to feasible options and at least one of these functions is non-linear. x variables model equipment sizes and production times and volume and the y variables are restricted to be 0 or 1 modelling action selection.

This problem has a unique global optimum if all of the functions involved are strictly convex (Grossmann et al., 2000) otherwise finding a global optimum is not guaranteed. One way to deal with non-convexities arising in the standard model of batch facilities is the use of the logarithmic change of variables proposed by Kocis & Grossmann (1988) which linearizes most of the functions and leads to a convex problem, approach that has been used among others by Rippin (1993), Montagna et al. (2000) and Moreno et al. (2009b). Another approach is the use of heuristic procedures (Grossmann et al., 2000). This was the option selected by Pinto et al. (2001) among others.

Some other authors as Voudouris & Grossmann (1993), Moreno & Montagna (2007b), Moreno & Montagna (2011) and Fumero et al. (2011) have modeled the design problem as a Mixed-Integer Linear Problem (MILP or MIP) selecting equipment sizes among a discrete set of available sizes. This alternative guarantees a global optimality in the solution of the batch design problem (Voudouris & Grossmann, 1993). Moreno & Montagna (2011) made a comparison between both MINLP and MILP approaches and conclude that although the precision of the model is reduced in a MILP approach, a superior performance is achieved.

A key feature in these design problems is the use of *Big-M* constraints to account for the selection decisions despite of being problematic (Bosch & Trick, 2005). Some authors that have included this type of constraints in their formulations are Gupta & Karimi (2003), Corsano et al. (2009), Moreno et al. (2009a) and Moreno & Montagna (2012). Obviously, these authors have found that the value of the *Big-M* parameters has a tremendous impact on the solution time. See for example Montagna et al. (2004) who made a comparison between the use of *Big-M* and *convex-hull* formulations; or Moreno & Montagna (2007b) who had to test different values for *Big-M* parameters.

1.5 Objectives

1.5.1 Main Objective

The main objective of this work is to study the use of an optimization based approach for the design of a biotechnological multiproduct batch plant with rigorous information of different processes from a real plant.

1.5.2 Specific Objectives

The specific objectives of this work are:

- To investigate a formulation for the desing of multi-product batch plants that is robust, scalable and reliable for its application in the design of real cases.
- To introduce a standard methodology, coming from the optimization field, to compare different formulations for the design of multi-product batch plants.
- To define an appropriate methodology to estimate the parameters of the defined optimization model based on rigorous information from real processes.
- To investigate the application of the developed methodology in the design of a biotechnological multi-product batch plant using real data of production processes that are actually part of a real multi-product batch plant.

1.6 Summary of methodology and principal results

A first approach to study the design of a real biotechnological multi-product batch plant was the use of the MINLP formulation proposed by Iribarren et al. (2004) using the *DICOPT* solver in *GAMS* language; nevertheless it was found that their formulation was only reliable for small plants with no more than 15 stages in total. A discussion of this fact can be found in Sandoval et al. (2016) in Chapter 2.

To avoid the problems that can be found in MINLP formulations a MILP reformulation is proposed that in a first stage permits the selection of equipment sizes over a range of continuous sizes (see Chapter 2); and in a second stage permits both, discrete and continuous sizes (see Chapter 3).

Selection of techniques to perform a defined step is addressed with the introduction of a route formulation that makes use of advanced reformulation techniques coming from the mixed-integer-programming literature: *clique* constraints. This formulation avoids the use of *Big-M* constraints.

The combination of the MILP reformulation with the *clique* constraints permits the definition of a methodology that meets all the desired requirements: is scalabe, robust and realiable for its application in the design of real multi-product batch plants.

As a final step of this work the proposed approach is used to study the design of the real biotechnological multi-product batch plant. Mass balances allow the computation of the parameters needed by the formulation and the obtained results illustrate the reliability of the optimization based approach in the design of real multi-product batch plants. The use of these type of models may achieve big savings in the cost of the main equipment of the plant.

Finally, it is important to highlight that the proposed approach takes at most a few minutes to find an optimum solution leaving plenty of space for continuing the addition of new and more complex constraints or objetive functions. In addition, lower level implementations in C or C^{++} could improve timing performance and the complexity of the model as well.

Chapter 2

MILP reformulations for the design of biotechnological multi-product batch plants using continuous equipment sizes and discrete host selection

Published on Computers and Chemical Engineering at January 2016 (Sandoval, G., Espinoza, D., Figueroa, N. y Asenjo, J.A. / Computers and Chemical Engineering 84 (2016) 1-11. See appendix A)

Abstract

In this article we present a new approach, relying on mixed-integer linear programming (MILP) formulations, for the design of multi-product batch plants with continuous sizes for processing units and host selection. The main advantage of the proposed approach is its scalability, that allows us to solve, within *reasonable* precision requirements, realistic instances. Furthermore, we show that many other alternatives are either numerically unstable (for the problem sizes that we are interested in), unable to solve large instances, or much slower than the proposed method. We present extensive computational experiments, which show that we are able to solve almost all tested instances, and, in average, we are ten times faster than alternative approaches. As we use a high level implementation language (AMPL) we should get further time improvements if lower level implementations are used (C, C^{++}).

Reproducibility of our results can be tested using our models and data available on-line at BPLIB[1].

Keywords: multi-product batch plant, MINLP, MILP, production path.

[1] Available in `http://www.dii.uchile.cl/~daespino/`

2.1 Introduction

Conventional multi-product batch process literature using an optimization-based approach model the design and synthesis of such plants with Mixed-Integer Non-Linear Programming (MINLP) formulations (Floudas, 1995). The usual objective is to minimize the investment cost subject to the fulfillment of the production targets of a given set of products. Major drawbacks are given by the combinatorial nature of mixed-integer programming and possible nonconvexities due to non-linearities. In computational optimization numerical issues of these formulations given by rounding errors, numerical instabilities and approximation errors are well-documented (Goldberg, 1991; Koch, 2004; Margot, 2009; Vielma, 2013).

Since Robinson & Loonkar (1972) different procedures have been proposed to tackle these problems (Reklaitis, 1990; Rippin, 1993; Barbosa-Póvoa, 2007; Verderame et al., 2010; Nikolopoulou & Ierapetritou, 2012) but a method that is more efficient for a particular example is hardly predictable (Ponsich et al., 2007) and nowadays the development of effective solution approaches and algorithms remains very necessary (Grossmann & Guillén-Gosálbez, 2010).

The logarithmic change of variables proposed by Kocis & Grossmann (1988) linearizes most of the functions and leads to a convex MINLP problem, approach used by Ravemark & Rippin (1998) and Montagna et al. (2000) among others. Another approach chosen by Pinto et al. (2001) and Ponsich et al. (2007) among others is the use of specially designed solvers which can usually find good feasible solutions by the use of heuristic procedures (Grossmann et al., 2000). In practice the best *off the shelf* solvers for this kind of problems are the open source codes BONMIN and SCIP and the commercial solvers BARON and DICOPT that stand out in Mittelmann's benchmarks for optimization software (Mittelmann, 2013). Nevertheless none of them guarantee convergence to a global optimum, converging in some instances to local optima or not converging altogether. For the particular case of BARON and DICOPT performance failures are reported for non-convex models (Ponsich et al., 2007; Rebennack et al., 2011; Li et al., 2012); nevertheless even in cases where theoretically the algorithms work, we found that in practice, they do not converge to the global optimum. We have run precise experiments that demonstrate these failures in convex MINLP formulations (see Section 2.2).

It is a fact that there is a huge gap between Mixed-Integer Linear Programming (MILP or MIP) and MINLP solvers technology (Nowak, 2005). Nowadays mixed-integer linear techniques are fast, robust and able to provide solutions to problems with up to millions of variables (Geißler et al., 2012). Taking advantage of this Voudouris & Grossmann (1992) used reformulation schemes to develop MILP models for the preliminary design of multi-product batch plants, introducing binary variables for the selection of discrete available equipment sizes. From this point, to the design decisions other were included as synthesis, production planning and scheduling (Voudouris & Grossmann, 1993); design and planning in a multiperiod scenario (Moreno & Montagna, 2007a); design of multi-product batch plants considering duplication of units in series (Moreno et al., 2009b) and the design and planning of multi-product batch plants using mixed-product campaigns (Corsano et al., 2009). Most recently these MILP formulations have been used to account for the design and scheduling of this type of plants (Fumero et al., 2011, 2012b,a) and for the design under uncertainty

considering different types of decisions (Durand et al., 2012; Moreno & Montagna, 2012; Durand et al., 2014; Moreno-Benito et al., 2014).

A key feature in these design problems is the use of *Big-M* constraints to account for selection decisions despite being problematic (Bosch & Trick, 2005). Some authors that have included this type of constraints in their formulations are Gupta & Karimi (2003); Corsano et al. (2009); Moreno et al. (2009a); Moreno & Montagna (2012). Obviously, these authors have found that the value of the *Big-M* parameters has a tremendous impact on the solution time; see for example Moreno et al. (2007). In addition it has been proven experimentally that other methods, as the convex hull formulation presented by Montagna et al. (2004) are better to account for selection decisions.

In this paper we develop a robust methodology to solve the design problem of a biotechnological multi-product batch plant in situations where equipment can be manufactured according to customer needs, as fermentors or tanks in general. To do that, we develop a MILP formulation which does not rely on the use of *Big-M* constraints and does not use a discrete range of equipment sizes. To do that we use four basic techniques (see Figure 2.1): First, an extension of the non-linear (but convex) formulation proposed by Kocis & Grossmann (1988) is applied. Secondly, to deal with non-linear convex inequalities *a priori* we constructed linear outer (or inner) approximations of them which allow us to compute (*a posteriori*) true feasible solutions and lower (or upper) bounds. Thirdly, to deal with integer variables, we used advanced reformulation techniques coming from the mixed-integer-programming literature (*clique* constraints). Finally, once the initial problem is transformed into a standard mixed-integer programming problem, it is possible to take advantage of mature commercial MIP solvers.

This approach, at least in our experiments, is more stable numerically, scalable, and faster to solve than current alternatives and can deal with the more general problem of jointly selecting equipment sizes and alternative production paths for multiple products. Using our approach, it is possible to quickly and accurately compute solutions at any desired precision level. In our extensive computational experiments (see figure 2.11) we found that current non-linear solvers only solved 43% of the instances generated for this study, while our approach was able to solve over 95% of the studied instances in a running time that, on average, was more than ten times faster than MINLP solvers in equivalent and standard MINLP formulations. To make these comparisons we introduce the performance profiles; a methodology borrowed from the optimization literature.

The rest of this paper is organized as follows. In section 2.2 typical drawbacks found by a commonly used MINLP solver and the standard MINLP formulation is presented. In section 2.3 classic and novel formulations for the design problem are described. Relevant information about the methodology used to benchmark different formulations and to avoid numerical instabilities is given in Section 2.4 and computational results are presented and discussed in Section 2.5. Finally, the conclusions are presented in Section 2.6.

2.2 Current limitations

Our main objective is finding a robust and scalable methodology for the design of biotechnological multi-product batch plant considering equipment sizing (design decisions)

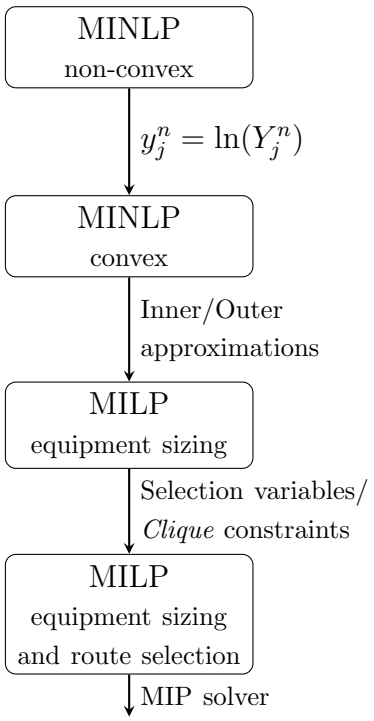


Figure 2.1 – Basic techniques used to model synthesis and design decisions considering continuous equipment sizes and discrete host selection.

and selecting the downstream processing stages (synthesis decisions). Given the complexities that to date have been added to the original design problem we decided to go back to the problem studied by Iribarren et al. (2004) where only design and synthesis decisions are modeled. In their paper they designed a biotechnological batch plant for the production of four recombinant proteins, i, where each can be synthesized by two different hosts, h, having four microorganisms in total. In addition to that three of the fifteen processing stages, j, may be performed by two different unit operations, d. In their formulation they used constant size (S_{ijdh}) and time (T_{ijdh}) factors to model each stage; considered duplication of units in parallel in-phase, G_{jd}, and out-of-phase, M_{jd}, in order to diminish either the equipment sizes V_j or cycle times TL_i, respectively, and used *Big-M* constraints to account for the selection of hosts and equipment.

As a correctness test, we took the example presented in Iribarren et al. (2004) and splitted into 16 different instances that only allow equipment selection. Then we tested two different yet equivalent formulations based on their model but removing host selection[2].

In the first (C1) the selection of hosts was eliminated by limiting the set of available hosts, H_i, to just one per protein, and in the second (C2), by setting the values of the selection binary variables to 1 for the selected hosts and 0 for those non-selected. If the solution is being found by the solvers, we should observe two things:

[2]To further isolate the results obtained from problems with non-standard local-settings, the GAMS modelling language was used and the experiments were run in the NEOS server (Gropp & Moré, 1997; Czyzyk et al., 1998; Dolan, 2001) available in `http://www.neos-server.org`

Table 2.1 – Comparison of the number of constraints and variables of some selected instances solved using models that account for selection with *Big-M* constraints -models (C1) and (C2)- and a classic formulation for design decisions only, (P1). All instances were solved using the DICOPT solver.

	Variables		Constraints		Status of solution
	Disc.	Cont.	Linear	Non-linear	
(C1)	480	185	303	10	Incorrect
(C2)	512	323	682	18	Incorrect
(P1)	360	71	179	9	Correct

(a) both model formulations (C1) and (C2) give the same solution, and

(b) the minimum of the separated instances is equivalent to the global minimum of the problem with host selection.

Contrary to what we expected differences in the objective function value for both (C1) and (C2) formulations went from 1% to 78% in the 16 instances studied (data not shown), and even more striking, the solver finds a local minimum which is worse than those found for most instances without host selection.

These numerical instabilities seem to be aggravated with size since it is known that DICOPT works fine for small instances. Situation in accordance to the results obtained by Ponsich et al. (2007). In order to show these differences in sizes we built Table 2.1 to compare the number of constraints and variables involved in the smaller instances of the cases (C1) and (C2), that were incorrectly solved according to the aforementioned results, with the size of a smaller instance that was correctly solved by a classic formulation (P1) that solves an equipment sizing problem similar to that presented by Iribarren et al. (2004), but with no selection of hosts or equipments, and using DICOPT solver. This last formulation is presented en Section 2.3.1.

2.3 Problem Formulation

Two major contributions are presented in this section. First, *clique* constraints are introduced to formulate the discrete part of the model allowing the selection of the production path without the use of *Big-M* constraints, in models (P2) and (P4). Second, a new approach, in Section 2.3.2, to handle non-linearities using standard reformulation techniques from the optimization field that permits the use of linear solvers leading to more reliable results and faster computing time. The relation among the four different models studied is shown in Figure 2.2.

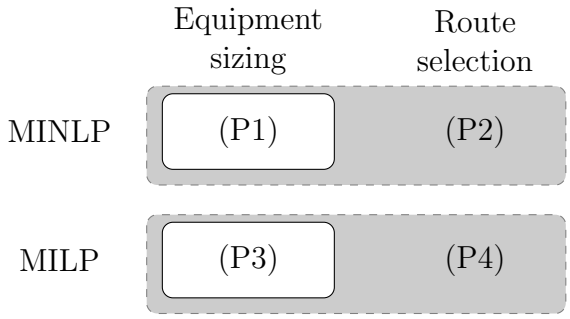


Figure 2.2 – Formulations compared in this article. Model (P1) is the most basic formulation that only includes design decisions. Model (P2) includes the selection of the downstream processes without the use of *Big-M* constraints. Models (P3) and (P4) are the transformed models of (P1) and (P2), respectively, using our proposed inner and outer approximations.

2.3.1 MINLP formulation

The equipment-sizing problem (P1)

In this section we present the most basic formulation for the design of biotechnological multi-product batch plants as only equipment sizing and duplication of units in parallel are considered.

The plant consists of a sequence of batch, semi-continuous and chromatographic stages used to manufacture different products i; where semi-continuous as well as chromatographic stages are composed by the semi-continuous items plus feed and product tanks. At each stage j there are X_j^2 groups of units operating in parallel out-of-phase and each group is conformed by X_j^1 units operating in-phase. For semi-continuous or chromatographic stages feed and product tanks can only be duplicated out-of-phase. Single production campaigns are considered and batches are transferred from one stage to the next without delay (zero wait policy).

The objective is to minimize the investment costs of main equipments of the plant (see equation (2.1)) given fixed production targets, d_i, over a time horizon δ.

$$\begin{aligned}\min \text{cost} = \sum_{j\in\mathcal{E}^1} X_j^1 X_j^2 \left(c_j^1 Y_j^{1\gamma_j^1}\right) \\ + \sum_{j\in\mathcal{E}^2\cup\mathcal{E}^3} \left[X_j^2\left(c_j^1 Y_j^{1\gamma_j^1}\right) + X_j^2\left(c_j^2 Y_j^{2\gamma_j^2}\right)\right. \\ \left. + X_j^1 X_j^2\left(c_j^3 Y_j^{3\gamma_j^3}\right)\right] + v\rho\delta \quad (2.1)\end{aligned}$$

Variables Y_j^i represent the different equipment sizes. Parameters c_j^i and γ_j^i are cost coefficients distinctive for each kind of equipment and v is a slack variable included to assure feasibility (Montagna et al., 2004).

Making the change of variables introduced by Kocis & Grossmann (1988) we get the new objective function (2.2).

$$\min \text{cost} = \sum_{j \in \mathcal{E}^1} c_j^1 \exp\left(x_j^1 + x_j^2 + y_j^1 \gamma_j^1\right) + \sum_{j \in \mathcal{E}^2 \cup \mathcal{E}^3} \left[c_j^1 \exp\left(x_j^2 + y_j^1 \gamma_j^1\right) + c_j^2 \exp\left(x_j^2 + y_j^2 \gamma_j^2\right) + c_j^3 \exp\left(x_j^1 + x_j^2 + y_j^3 \gamma_j^3\right)\right] + v\rho\delta \quad (2.2)$$

At each stage and for each product the size of the units must allow the processing of the incoming batch which can be splitted among X_j^1 units to not surpass the upper bound capacity of the equipment. In batch stages this constraint can be written as equation (2.3a); convexified in equation (2.3b).

$$Y_j^1 \geq \frac{S_{ij}^1 Y_i^4}{X_j^1} \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^1 \quad (2.3a)$$

$$y_j^1 + x_j^1 \geq s_{ij}^1 + y_i^4 \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^1 \quad (2.3b)$$

As in semi-continuous or chromatographic stages duplication is allowed just for semi-continuous items, feed and product tanks are sized using constraints (2.4) and (2.5).

$$y_j^1 \geq s_{ij}^1 + y_i^4 \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \cup \mathcal{E}^3 \quad (2.4)$$

$$y_j^2 \geq s_{ij}^2 + y_i^4 \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \cup \mathcal{E}^3 \quad (2.5)$$

Chromatographic columns have to process the incoming batch and both duplication in-phase and out-of-phase are allowed. Duplication in-phase is modeled in size constraint (2.6) since this permits smaller units and duplication out-of-phase is reflected in time constraints.

$$y_j^3 + x_j^1 \geq s_{ij}^3 + y_i^4 \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^3 \quad (2.6)$$

The cycle time for each product i, Y_i^5, is defined as the time elapsed between the production of two consecutive batches and is given by the larger operating time, T_{ij}, among the stages in the process. This time can be decreased if a duplication of units out-of-phase is used:

$$Y_i^5 \geq \frac{T_{ij}}{X_j^2} \qquad \forall i \in \mathcal{I}, j \in \mathcal{E} \quad (2.7)$$

As batch stages operate for a fixed time, T_{ij}^0, cycle time constraint in its convex form is given by equation (2.8):

$$y_i^5 + x_j^2 \geq t_{ij}^0 \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^1 \quad (2.8)$$

Semi-continuous stages, on the other hand, operate during a time that depends on the final batch size, Y_i^4. For those stages the cycle time is constrained as in equation (2.9).

$$Y_i^5 \geq \frac{T_{ij}^1 \frac{Y_i^4}{X_j^1 Y_j^3}}{X_j^2} \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \quad (2.9a)$$

$$y_i^5 + x_j^2 \geq t_{ij}^1 + y_i^4 - x_j^1 - y_j^3 \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \tag{2.9b}$$

Lastly, chromatographic stages are modeled considering both fixed and variable operation times leading to the highly non-linear constraint (2.10).

$$Y_i^5 \geq \frac{T_{ij}^0 + T_{ij}^1 \frac{Y_i^4}{X_j^1 Y_j^3}}{X_j^2} \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^3 \tag{2.10a}$$

$$y_i^5 + x_j^2 \geq \ln\left[\exp\left(t_{ij}^0\right) + \exp\left(t_{ij}^1 + y_i^4 - x_j^1 - y_j^3\right)\right] \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^3 \tag{2.10b}$$

Production targets for all products, d_i, must be satisfied within the time horizon δ.

$$\sum_{i \in \mathcal{I}} \frac{d_i Y_i^5}{Y_i^4} \leq \delta + v\delta \tag{2.11a}$$

$$\sum_{i \in \mathcal{I}} \frac{d_i}{\delta} \exp\left(y_i^5 - y_i^4\right) \leq 1 + v \tag{2.11b}$$

Finally, variables for duplication in-phase X_j^1 are restricted to integer values using constraints (2.12) and (2.13), where z_{jm}^3 are binary variables and $\mathcal{M}$ a set of available units to operate in parallel in-phase. The same is valid for variables for duplication out-of-phase, X_j^2.

$$x_j^1 = \sum_{m \in \mathcal{M}} z_{jm}^3 \ln(m) \qquad \forall j \in \mathcal{E} \tag{2.12}$$

$$\sum_{m \in \mathcal{M}} z_{jm}^3 = 1 \qquad \forall j \in \mathcal{E} \tag{2.13}$$

Appropriate upper and lower bounds are also considered for all of the variables.

The design problem with selection of routes and equipment sizing (P2)

More recent models (like Iribarren et al. (2004)) take into account the joint selection of the production processes including selection of hosts and equipment. Their formulation uses classical *Big-M* constraints. Since we know that these constraints are problematic (Bosch & Trick, 2005) in this work we propose a different way to formulate the integer part of the problem replacing the *Big-M* by *clique* constraints. With this formulation all constraints are ignored except the upper bound on the variables (Dietrich et al., 1993). Model (P2) includes the sizing of the equipment, the duplication of units in parallel working in-phase and out-of-phase and accounts for the selection of the global process selecting routes which are defined as the series of unit operations used to purify a protein given a certain host that synthesizes it. In this way once the pair product-host, (i, h), is selected the set of stages conforming the process is fixed.

The objective function becomes:

$$\begin{aligned}\min \ \text{cost} = & \sum_{j\in\mathcal{E}^1} z_j^2 c_j^1 \exp\left(x_j^1 + x_j^2 + y_j^1\gamma_j^1\right) \\ & + \sum_{j\in\mathcal{E}^2\cup\mathcal{E}^3} z_j^2 \left[c_j^1 \exp\left(x_j^2 + y_j^1\gamma_j^1\right) + c_j^2 \exp\left(x_j^2 + y_j^2\gamma_j^2\right)\right. \\ & \left. + c_j^3 \exp\left(x_j^1 + x_j^2 + y_j^3\gamma_j^3\right)\right] + v\rho\delta \end{aligned} \quad (2.14)$$

Since some stages can be unused and just one route per protein can be selected we introduced two binary variables: z_{ih}^1 and z_j^2. z_{ih}^1 is equal to 1 when for product i synthesis host h is selected and 0 otherwise and z_j^2 is 1 when stage j is used to process at least one of the products and 0 otherwise. Constraint (2.15) enforces to chose just one host h to produce the protein i and constraint (2.16) permits stage j to be used just in case at least one product needs it to be processed.

$$\sum_{(i,h)\in\mathcal{U}} z_{ih}^1 = 1 \quad (2.15)$$

$$z_j^2 \geq z_{ih}^1 \qquad \forall (i,h,j) \in \mathcal{R} | (i,h) \in \mathcal{U} \quad (2.16)$$

For chromatographic stages constraints take the form of equations (2.17)- (2.20) that are trivially satisfied if host h is not selected to produced protein i ($z_{ih}^1 = 0$). When the host h is selected for protein i ($z_{ih}^1 = 1$) and the stage j has to be performed to process product i then $z_j^2 = 1$ and constraints are the same as in previous formulation (Section 2.3.1).

$$y_j^1 z_j^2 \geq s_{ihj}^1 z_{ih}^1 + y_{ih}^4 z_{ih}^1 \qquad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3 \quad (2.17)$$

$$y_j^2 z_j^2 \geq s_{ihj}^2 z_{ih}^1 + y_{ih}^4 z_{ih}^1 \qquad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3 \quad (2.18)$$

$$y_j^3 z_j^2 + x_j^1 z_j^2 \geq s_{ihj}^3 z_{ih}^1 + y_{ih}^4 z_{ih}^1 \qquad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3 \quad (2.19)$$

$$y_{ih}^5 z_{ih}^1 + x_j^2 z_j^2 \geq \ln\left[\exp\left(t_{ihj}^0\right) + \exp\left(t_{ihj}^1 + y_{ih}^4 - x_j^1 - y_j^3\right)\right] z_{ih}^1 \qquad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3 \quad (2.20)$$

If stage j is not necessary for the process ($z_j^2 = 0$); then equipment sizes are set to 0 with constraints as (2.21) and no unit is considered to conform that stage (constraint (2.22)):

$$y_j^{1,LO} z_j^2 \leq y_j^1 \leq y_j^{1,UP} z_j^2 \qquad \forall j \in \mathcal{E} \quad (2.21)$$

$$\sum_{m\in\mathcal{M}} z_{jm}^3 = z_j^2 \qquad \forall j \in \mathcal{E} \quad (2.22)$$

Finally, in the planning horizon constraint (2.23) only the terms associated to the selected host per protein are considered.

$$\sum_{(i,h)\in\mathcal{U}} \frac{d_{ih}}{\delta} z^1_{ih} \exp\left(y^5_{ih} - y^4_{ih}\right) \leq 1 + v \tag{2.23}$$

2.3.2 Mixed-Integer linear formulations

To obtain more accurate solutions and, specially in larger instances, in a reasonable running time we present a MILP reformulation, which can be solved using any commercial MILP solver. These models are basically equal to their MINLP counterpart but replacing the non-linear objective and time constraints with sets of linear functions which give arbitrarily good lower or upper approximations of their respective original functions. The actual optimal solution is in between both approximations and the precision level is given by the number of cutting points selected to generate the set of linear functions to replace each non-linear function and the actual selection of the approximation points used for example equispaced or non-equispaced. In this way the accuracy of the solution can be as high as desired at the cost of longer computing time.

Inner and outer approximations

Given a convex function of one variable $g(x) \leq 0$ and a set of points $\{x_k\}_{k=1,...,n}$ in the domain of g then, is easy to see that:

$$\{x|g(\hat{x}_k) + \nabla g(\hat{x}_k)(x - \hat{x}_k) \leq 0 \quad k = 1, ..., n\} \supseteq \{x|g(x) \leq 0\} \tag{2.24}$$

and

$$\left\{x|g(\hat{x}_k) + \frac{g\left(\hat{x}_{k+1}\right) - g\left(\hat{x}_k\right)}{\hat{x}_{k+1} - \hat{x}_k}\left(x - \hat{x}_k\right) \leq 0 \quad k = 1, ..., n\right\} \subseteq \{x|g(x) \leq 0\}, \tag{2.25}$$

which allows for straightforward lower and upper approximations of g. Using this fact, it is easy to find inner and outer approximations of the problems (P1) and (P2).

In fact, for each non-linear constraints of the form $g_j(x) \leq 0$, and considering an arbitrary set of cutting points in the domain $\{x_k\}_{k=1,...,n}$ the consideration of the set of constraints

$$g_j(\hat{x}_k) + \nabla g_j(\hat{x}_k)\left(x - \hat{x}_k\right) \leq 0 \quad k = 1, ..., n \tag{2.26}$$

which leads to a larger feasible set, as can be seen in Figure 2.3a. On the other hand, we consider the set of constraints

$$g_j(\hat{x}_k) + \frac{g_j\left(\hat{x}_{k+1}\right) - g_j\left(\hat{x}_k\right)}{\hat{x}_{k+1} - \hat{x}_k}\left(x - \hat{x}_k\right) \leq 0 \quad k = 1, ..., n-1 \tag{2.27}$$

which leads to a smaller feasible set, as can be seen in Figure 2.3b.

In the same way, the minimization of the cost objective function, $f(x)$, can be replaced by

$$\begin{aligned} \min \quad & v \\ s.t. \quad & v \geq f(\hat{x}_k) + \nabla f(\hat{x}_k)(x - \hat{x}_k) \quad k = 1, ..., n \end{aligned} \tag{2.28}$$

which leads, together with an outer approximation of constraints, to a lower bound of the true cost. The objective function can also be replaced by

$$\begin{aligned} \min \quad & v \\ s.t. \quad & v \geq f(\hat{x}_k) + \frac{f(\hat{x}_{k+1}) - f(\hat{x}_k)}{\hat{x}_{k+1} - \hat{x}_k}(x - \hat{x}_k) \\ & \qquad\qquad\qquad\qquad k = 1, ..., n \end{aligned} \tag{2.29}$$

which leads, together with an inner approximation of constraints, to an upper bound of the true cost.

In what follows, $\nabla f(\hat{x}_k)$ or $\frac{f(\hat{x}_{k+1}) - f(\hat{x}_k)}{\hat{x}_{k+1} - \hat{x}_k} = \alpha_k$ in the lower or upper approximation, respectively, $\hat{x}_k = b_k$ and $f(\hat{x}_k) = \beta_k$.

Reformulation for the equipment-sizing problem (P3)

The assumptions for this model are the same as those for the MINLP proposed in Section 2.3.1.

Objective function Cost functions of equation (2.2) are individually linearized using the approximations given in Section 2.3.2 which leads to equation (2.30):

$$\min \ \text{cost} = \sum_{j \in \mathcal{E}^1} v_j^1 + \sum_{j \in \mathcal{E}^2 \cup \mathcal{E}^3} \left[v_j^1 + v_j^2 + v_j^3 \right] \tag{2.30}$$

Constraints Batch and semi-continuous stages and binary variables for duplication of units constraints in this MILP model are the same as those in the MINLP model shown in Section 2.3.1.

Chromatographic stages Size constraints for feed and product tanks and column size constraint are the same as those in the MINLP model shown in Section 2.3.1.

Time constraint (2.31) is obtained from the linearization of equation (2.10):

$$y_i^5 + x_j^2 \geq \alpha_k^{6ij} \left(y_i^4 - x_j^1 - y_j^3 - b_k^{6ij} \right) + \beta_k^{6ij} \qquad \forall i \in \mathcal{I}, j \in \mathcal{E}^3, k \in \mathcal{K}^6 \tag{2.31}$$

Planning horizon From linearization of equation (2.11) constraint (2.32) is obtained:

$$\sum_{i \in \mathcal{I}} v_i^7 \leq 1 \tag{2.32}$$

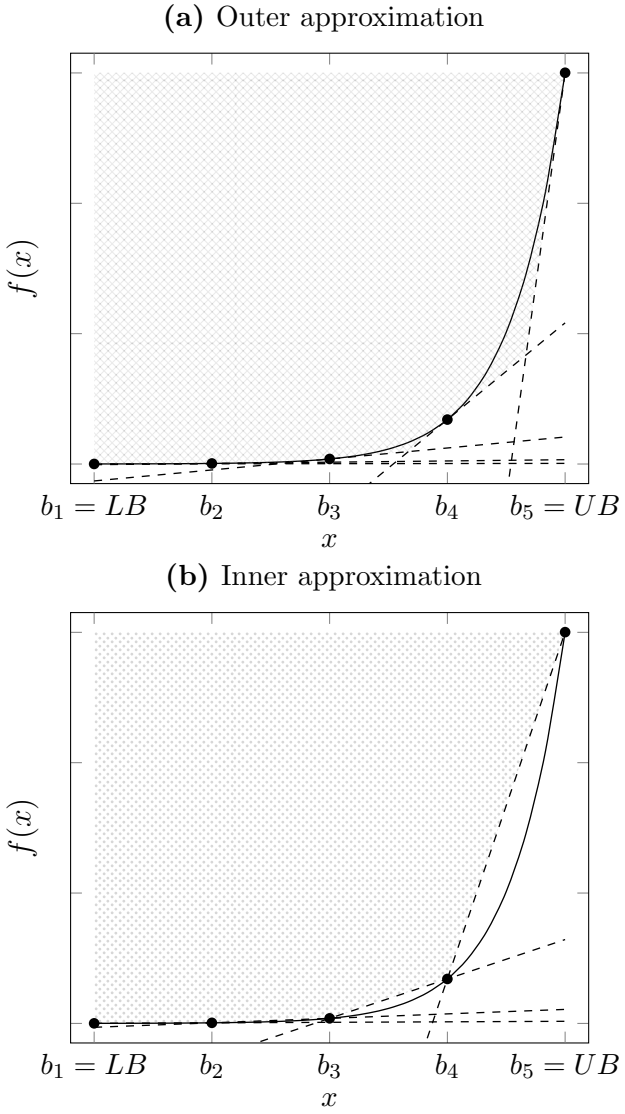


Figure 2.3 – Feasible region (patterned area) of (a) outer and (b) inner approximations (dashed lines) of an exponential function (solid line). Points b_i are the cutting points and LB and UB are the lower and upper bounds of x.

Auxiliary variables Cost functions in the objective function are linearized as shown in equation (2.33) and planning horizon constraint is linearized as shown in equation (2.34):

$$v_j^1 \geq \alpha_k^{1j}\left(x_j^1 + x_j^2 + \gamma_j^1 y_j^1 - b_k^{1j}\right) + \beta_k^{1j} \qquad \forall j \in \mathcal{E}^1, k \in \mathcal{K}^1 \tag{2.33}$$

$$v_i^7 \geq \frac{d_i}{\delta}\alpha_k^{7i}\left(y_i^5 - y_i^4 - b_k^{7i}\right) + \frac{d_i}{\delta}\beta_k^{7i} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}^7 \tag{2.34}$$

Reformulation for the design problem considering route selection and equipment sizing (P4)

Similar to the model presented in Section 2.3.2 this model was built based on its MINLP counterpart and most of constraints remain the same.

The objective function is the same as that from model (P3) and for stages design only differences are encountered for time constraints of chromatographic stages. In this way, applying the inner or outer approximations to equation (2.20) constraint (2.35) is obtained:

$$y_{ih}^5 + x_j^2 \geq \alpha_k^{6ihj}\left(y_{ih}^4 - x_j^1 - y_j^3\right) + \left(\beta_k^{6ihj} - \alpha_k^{6ihj} b_k^{6ihj}\right) z_{ih}^1 \qquad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3, k \in \mathcal{K}^6 \tag{2.35}$$

This set of equations together with constraints of the type of (2.36) for variables y_j^1, y_j^2, y_{ih}^4, y_{ih}^5, x_j^1 and x_j^2 will model the same situation as in (P2).

$$y_j^{3,LO} z_j^2 \leq y_j^3 \leq y_j^{3,UP} z_j^2 \qquad \forall j \in \mathcal{E}^3 \tag{2.36}$$

If $z_j^2 = 0$ constraint (2.35) is trivially satisfied. On the other hand if $z_{ih}^1 = 0$ constraint (2.35) becomes:

$$x_j^2 \geq \alpha_k^{6ihj}\left(-x_j^1 - y_j^3\right) \qquad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3, k \in \mathcal{K}^6 \tag{2.37}$$

Since α_k^{6ihj} is a positive parameter constraint (2.37) will be always satisfied only if $|x_j^1| > |y_j^3|$ or if both variables are bigger than or equal to 0. To assure this data preprocessing is necessary. As x_j^1 is always bigger than 0 we normalized variables y_j^3 by their lower bound. More details in section 2.4.3.

For the case of planning horizon constraint little difference is found between (2.32) and (2.38). The last one takes into account host selection:

$$\sum_{(i,h)\in\mathcal{U}} v_{ih}^7 \leq 1 \tag{2.38}$$

Finally, constraints for auxiliary variables v_j^1, v_j^2, v_j^3 and v_{ih}^7 are different from those for problem (P3) to account for route selection:

$$v_j^1 \geq \alpha_k^{1j}\left(x_j^1 + x_j^2 + \gamma_j^1 y_j^1\right) + \left(\beta_k^{1j} - \alpha_k^{1j} b_k^{1j}\right) z_j^2 \qquad \forall j \in \mathcal{E}^1, k \in \mathcal{K}^1 \tag{2.39}$$

$$v_j^1 \leq z_j^2 v_j^{1,UP} \qquad \forall j \in \mathcal{E}^1 \tag{2.40}$$

$$v_{ih}^7 \geq \frac{d_i}{\delta}\alpha_k^{7ih}\left(y_{ih}^5 - y_{ih}^4\right) + \frac{d_i}{\delta}\left(\beta_k^{7ih} - \alpha_k^{7ih} b_k^{7ih}\right) z_{ih}^1 \qquad \forall (i,h) \in \mathcal{U}, k \in \mathcal{K}^7 \tag{2.41}$$

$$v_{ih}^7 \leq z_{ih}^1 v_{ih}^{7,UP} \qquad \forall (i,h) \in \mathcal{U} \tag{2.42}$$

If $z_j^2 = 0$ constraints (2.39) to (2.42) are trivially satisfied and if $z_{ih}^1 = 0$ then constraints (2.41) and (2.42) are trivially satisfied. Finally, if $z_{ih}^1 = 1$ and $z_j^2 = 1$ then constraints (2.39) to (2.42) are the same as those in the MILP formulation without route selection.

2.4 Methods

2.4.1 Solvers and modelling language

For MINLP problems the open source BONMIN 1.5 and SCIP 3.0.1 solvers were studied. In our computational tests SCIP uses SoPlex 1.7.1 as the LP solver and BONMIN (with its default algorithm, B-Hyb) uses Cbc 2.7.1 as the MIP solver and Ipopt 3.10.0 with MUMPS as linear solver. For the case of BONMIN we tested 3 over 5 available algorithms: B-Hyb the default algorithm, B-Ecp a specific parameter setting of B-Hyb that can be faster in some cases (Bonami & Lee, 2013) and B-OA using CPLEX as the MILP solver that according to Mittelmann (2013) can be faster for convex instances. In preliminary studies solvers as KNITRO and COUENNE were also tested to solve our MINLP formulations, but their performance in our simplest instances were poorer than that for the selected solvers.

For MILP problems the commercial CPLEX solver in its version 12.4.0.0 was used as it is one of the top performer from the literature (Mittelmann, 2013).

All models were coded using the AMPL modelling language.

2.4.2 Execution environment

Each instance was executed using a single thread on a Intel(R) Xeon(R) CPU E5620@2.40GHz with a running time limit of 48 hours, an optimality relative gap of 0.1% for models (P3) and (P4) and 2% for (P1) and (P2), and a maximum memory usage of 6Gb of RAM.

The difference in the prescribed optimality gap for MILP and MINLP solvers is given by the fact that while MINLP problems are solved to find the actual minimum cost function within a defined optimality gap, and therefore an *a priori* optimality gap, MILP models find true upper and lower bounds for the actual cost function leading to an *a posteriori* optimality gap that is computed afterwards. As will be shown in Section 2.5.2, this difference ensure that our results are comparable.

2.4.3 Methodology

Instances

To compare different approaches two set of instances, with randomly generated data between given *reasonable* upper and lower bounds, were built: "sizing instances", to compare simpler models (P1) and (P3), and "routing instances" to compare more complex models (P2) and (P4). We considered a variety of different number of proteins to be produced (4 to 6), number of stages to conform the process (11 to 65), number of routes to synthesize the product (20 to 65) and different cost coefficients values (1% to 110% of nominal values).

Benchmarking

In order to compare the model-solver pairs studied in this work we introduce a new tool for process engineers that was introduced in the optimization field by Dolan & Moré (2002) to compare different optimization software: the performance profile.

As stated by Dolan & Moré (2002) the performance profile for a solver is the "cumulative distribution for a performance metric", for example computing time. In this way things like how many instances a solver is able to solve given some stop criteria like those shown in Section 2.4.2, or how fast it solves different instances of the same type of problem can be seen graphically.

As an example of how to read these plots, in Figure 2.4a it can be seen that when using 17 cutting points 40% of the instances were solved to an *a posteriori* optimality gap up to 2% while when using 33 cutting points leads to an optimality gap under 0.5% for the same amount of instances.

Data pre-processing

It is known that zero-one problems of large-scale are hard combinatorial optimization problems (Crowder et al., 1983; Koch, 2004; Applegate et al., 2007) reason why in order to obtain reliable solutions preprocessing data is necessary. The use of tight bounds and the normalization of the variables are necessary to decrease numerical errors.

Although not all of our instances are big enough to need data preprocessing all were subjected to the same treatment:

- All variable bounds and parameters associated to variables Y_j and Y_i were normalized by their respective lower bounds.
- Size and time factors were normalized and dimensionless considering the respective associated units. For example, as size factor for tanks have units of batch size divided by a volume this parameters are dimensionless by multiplying by the lower bound of the final batch size and dividing by the respective tank lower bound.
- Lower bounds for the cycle time were tightened using time constraints and upper bounds for final batch product were tightened using size constraints.

Table 2.2 – Sizes of sample instances solved using non-linear model (P1).

	Variables		Constraints	
	Discrete	**Continuous**	**Linear**	**Non-linear**
Small	264	57	139	9
Medium1	840	157	407	5

2.5 Results and Discussion

In this section we show the robustness of our proposed MILP transformations and its superiority over classic MINLP formulations with *Big-M* constraints using performance profiles, a methodology borrowed from the optimization literature. Our approach is not only able to find correct solutions in realistic situations unlike MINLP formulations but also in a small fraction of the time required by those approaches. Major implications of these features are the exactness of the solutions that make this information reliable for decision-making; and as time reduction is significant numerous alternatives can be tested with the same formulation or with complexified models that may address the combination of different types of decisions.

This presentation is organized as follows: first, we describe the instances generated for comparison then discuss the selection of the cutting points for the proposed approach and finally, we compare MINLP and MILP formulations in terms of their performance solving the sets of instances using time as the metric.

2.5.1 Size of instances

To compare the most basic and easy to solve problems (P1) and (P3) a set of 186 instances ("sizing instances") were generated varying the number of proteins to be produced (2 to 6), the number of stages that conform each process (11-35) and the cost coefficient values (1% - 110% of nominal values). Sizes of these instances in terms of number of variables and constraints are shown in tables 2.2 and 2.3, where "Small" corresponds to an example of one of the smaller instances solved with different models and "Medium1", to an example of the bigger instances solved for these two models. As it can be seen in both tables new auxiliary variables and the sets of linear functions generated to replace non-linear restrictions makes the problem from 7 to 12 times bigger in terms of linear constraints when 33 cutting points are used for linearization with an increase in about 50% of continuous variables. However, as we will see later, this increase in variables and constraints leads to smaller execution times and more accurate results.

To test models (P2) and (P4), as they were posed to solve more complex scenarios, a set of 249 new and bigger instances ("routing instances") were generated varying the number of proteins to be produced (4 to 6), the number of stages conforming the global process (18 to 65) and the number of routes available to produce the proteins (20 to 40). Sizes of these instances in terms of number of variables and constraints are shown in tables 2.4 and

Table 2.3 – Sizes of sample instances solved using linear model (P3) with 33 cutting points for linear inner approximation.

	Variables		Constraints	
	Discrete	**Continuous**	**Linear**	**Non-linear**
Small	264	87	1357	-
Medium1	840	237	3096	-

Table 2.4 – Sizes of sample instances solved using non-linear model (P2).

	Variables		Constraints	
	Discrete	**Continuous**	**Linear**	**Non-linear**
Small	279	57	251	9
Medium1	881	157	719	5
Medium2	488	152	1262	22
Large	1794	613	15766	462

2.5, where "Medium2" corresponds to an example of one of the smaller "routing instances" solved with models (P2) and (P4) and "Large", to an example of the bigger instances solved in this work. As it can be seen in both tables, in comparison with tables 2.2 and 2.3, the number of discrete variables increases by 5% with the addition of selection variables, z, and the number of linear constraints increases by 15% for (P4) and is around double for (P2). As we will see later, this addition permits the resolution of more complex scenarios, while at the same time not affecting execution time or optimality gap in comparison with the more basic formulation.

Table 2.5 – Sizes of sample instances solved using linear model (P4) with 33 cutting points for linear inner approximation.

	Variables		Constraints	
	Discrete	**Continuous**	**Linear**	**Non-linear**
Small	279	87	1539	-
Medium1	881	237	3602	-
Medium2	488	229	4829	-
Large	1794	926	46489	-

2.5.2 Selection of cutting points

Contrary to MINLP problems (P1) and (P2) that are solved to an *a priori* optimality gap, models (P3) and (P4) give true upper and lower bounds for the actual cost function of each instance and therefore an optimality gap that is computed *a posteriori*. Figure 2.4a shows the performance profiles of the gaps obtained *a posteriori* for the "sizing instances" solved with 17, 33 and 65 cutting points that generate 16, 32 and 64 linear functions for the inner approximations, respectively. Here we can see that our linear model is able to solve all instances with a maximum gap of 5% in less than 16 seconds when using 17 cutting points, and a gap of less than 0.5% in less than 64 seconds when using 65 cutting points. The running time profiles can be seen in Figure 2.4b. If 65 cutting points had been selected, the optimality gap for non-linear solvers would have been around 0.5% as that is the worst gap obtained *a posteriori* with CPLEX (Figure 2.4a). Given this, in our final experiments we use a set of 33 cutting points, since this option gives the largest improvement in gap versus the increase in execution time, and a slightly bigger optimality gap criteria of 2% was selected for non-linear solvers.

Once the number of points is selected, the specific values of these points must be chosen. The most obvious choice is equispaced points which, for the (relevant) exponential function e^x generates small errors for low values of x and large errors for high values. Another alternative is to use the expression (2.43), where N is the total number of cutting points including $-\infty$ as x_1 and $\bar{x}$ as the upper bound of x. This is a good approximation in order to minimize the *maximum value* of the error (see Figure 2.5).

$$x_k = 2\ln\left(\frac{k-1}{N-1}\right) + \bar{x} \qquad \forall k \in 2...N \tag{2.43}$$

We can see in Figure 2.5 that the choice of equispaced points leads to better approximations for low values of x, but much worse for high values. For our numerical experiments, equispaced points work better: while execution time remains the same for both approaches, *a posteriori* gaps were slightly worse for non-equispaced points (Figure 2.6).

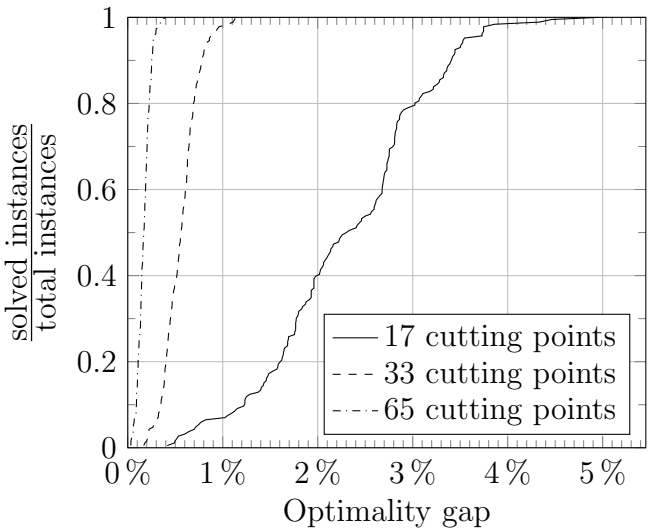
This leaves open important questions about the optimal point selection to improve the precision of upper and lower approximations. Our preliminary simulations seem to indicate that giving substantial attention to smaller values of x could significantly improve the results, but this is left for further research.

2.5.3 Equipment sizing: comparison of problems (P1) and (P3)

We tested three different combinations of solvers-models: the linear model (P3) was solved using CPLEX as solver while the non-linear model, (P1), was solved using SCIP and 3 of the 5 algorithms that are available for using BONMIN which were chosen based on BONMIN users' manual and Mittelmann's benchmarking (Mittelmann, 2013) information. All of the instances were solved using the stopping criteria and conditions mentioned in Section 2.4.2.

Figure 2.7 shows the performance profile of running time using BONMIN-Hyb, BONMIN-Ecp, BONMIN-OAcpx, SCIP and our CPLEX-based approach. From this, we can see that

(a) Relative optimality gap



(b) Running time

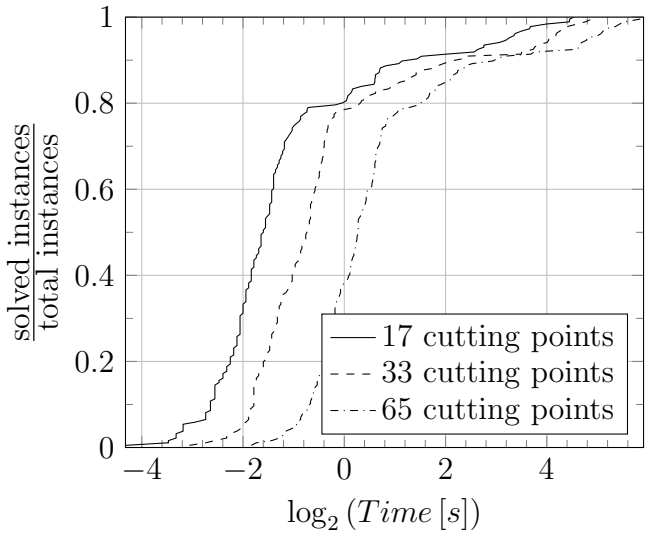


Figure 2.4 – Comparison of performance profiles of (a) relative optimality gap obtained *a posteriori* and (b) the logarithm of the running time of "sizing instances" solved with linear model (P3) using 17, 33 and 65 cutting points for lower and upper approximations with an optimality relative gap of 0.1%.

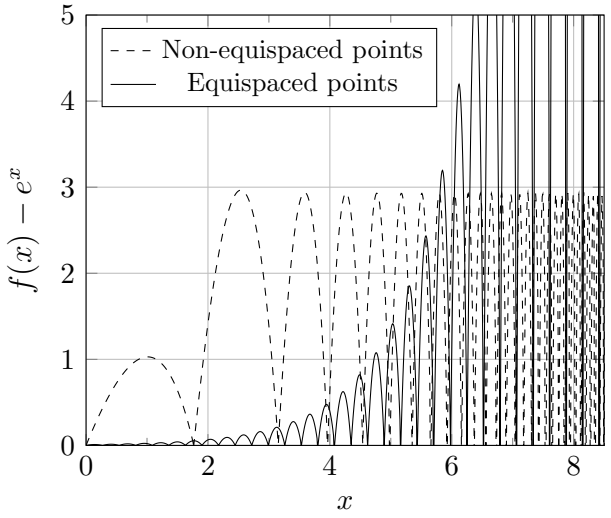


Figure 2.5 – Comparison of absolute errors using 2 different sets of 33 cutting points where $f(x)$ are the linear functions used to approximate the exponential function between 2 cutting points.

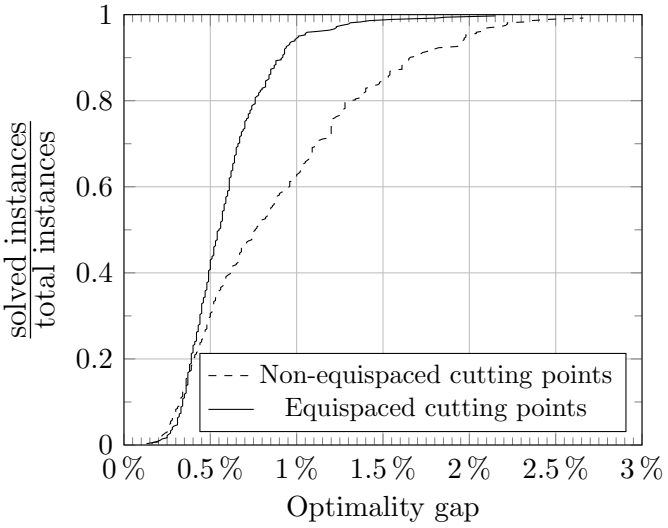


Figure 2.6 – Comparison of performance profiles of *a posteriori* gaps obtained using 33 cutting points to solve model (P4) where $f(x)$ are the linear functions used to approximate the exponential function between 2 cutting points. Time limit was set in 12 hours.

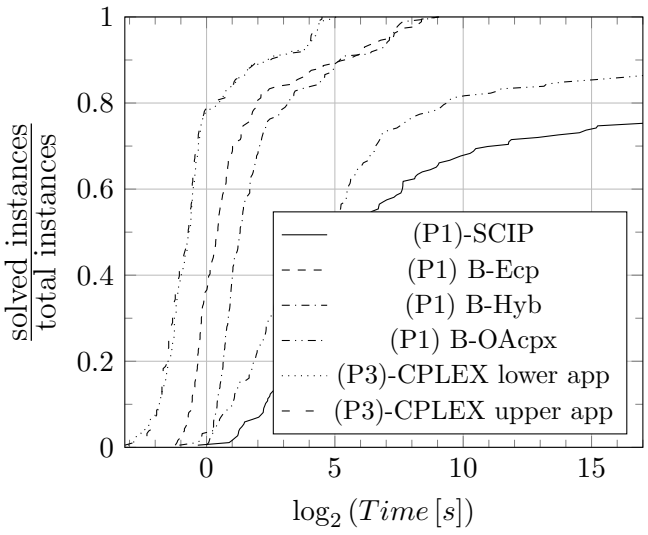


Figure 2.7 – Comparison of performance profiles of the logarithm of running time of "sizing instances" solved using models (P1) and (P3) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

problem (P1) was solved faster using any BONMIN algorithm than using SCIP solver. Moreover, SCIP only worked well in about the 75% of the instances, while BONMIN is able to solve the 85% of the instances using the OA algorithm and the 100% of the studied instances using either the Ecp or the Hyb algorithm. On the other hand, model (P3) solved using CPLEX is able to solve all of the instances studied, as well as BONMIN, but taking much less time than the problem (P1). As B-Ecp and B-Hyb seems to be equally good for those instances that take longer to be solved we decided to use as the performance metric the ratio of the computing time of the model-solver versus the best time of all of the model-solvers, denoted by τ. Those performance profiles are plotted in Figure 2.8 where we can see even more clear than the CPLEX-based approach is always better than all of the other options and that Ecp algorithm is always better than Hyb for this asked optimality gap.

2.5.4 Routes selection: comparison of problems (P2) and (P4)

As a test of correctness, models (P2) and (P4) were solved using the generated "sizing instances" where just one route was available to produce each product. Contrasting these results to those obtained by (P1) and (P3) it can be seen in Figure 2.9 that both formulations, for equipment sizing and considering routes, are consistent solving the same amount of instances in virtually the same amount of time.

Performance profiles of the relative difference between the value of the objective function obtained with simpler -(P1) and (P2)- and more complex formulations -(P3) and (P4)- are presented in Figure 2.10 where it can be seen that the differences between MILP formulations, as well as for MINLP formulations solved using BONMIN, are at most the optimality gap asked for each solver. The case of SCIP is different because in almost the 10% of the instances

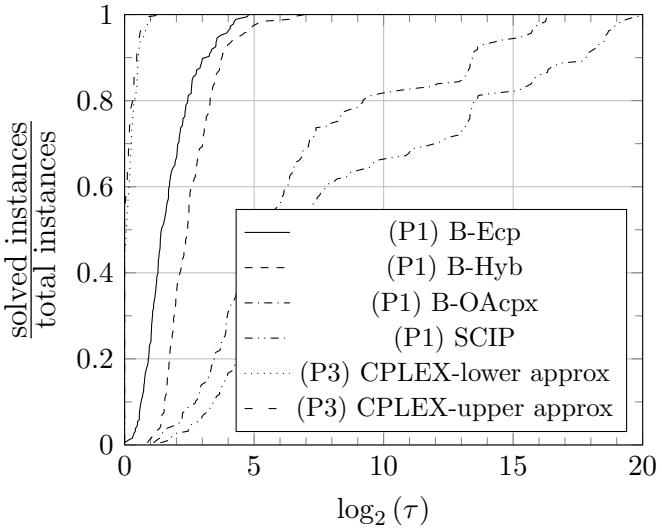


Figure 2.8 – Comparison of performance profiles of the logarithm of the ratio of the computing time of the pair model-solver versus the best time of the pairs model-solvers for "sizing instances" solved with models (P1) and (P3) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

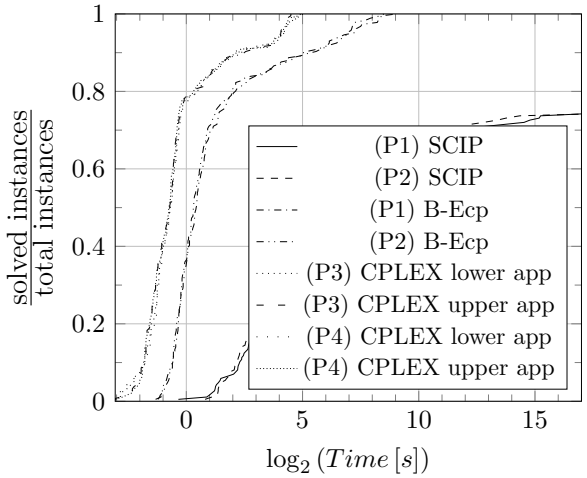


Figure 2.9 – Comparison of performance profiles of the logarithm of running time of "sizing instances" solved using models (P1), (P2), (P3) and (P4) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

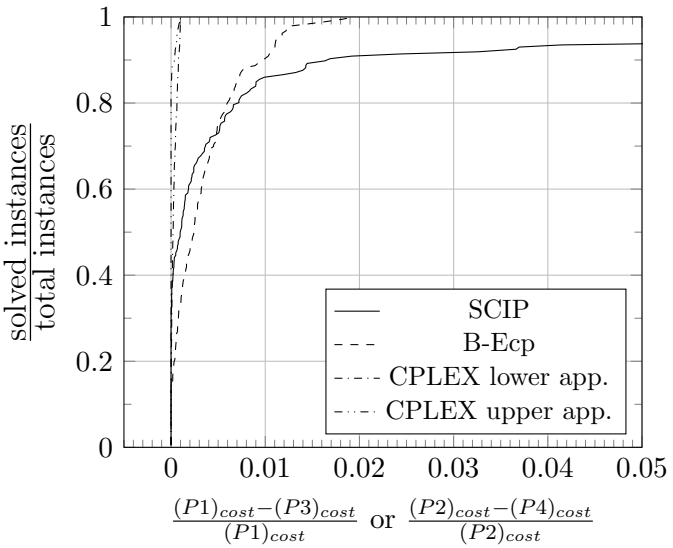


Figure 2.10 – Comparison of performance profiles of relative difference between simple and more complex formulation for "sizing instances". Models (P1) and (P2) were solved to an optimality gap of 2% and (P3) and (P4), to an optimality gap of 0.1%.

the solver gives results with a difference in the objective function between models (P1) and (P2) greater than the optimality gap which shows that this solver is not reliable to solve this type of problems.

As a final step in this work we compare in Figure 2.11 the performance profiles of total running time obtained after solving all instances generated for this work (435 in total). Here we can see that for "routing instances" (P4) is much more robust than (P2), that was not able to solve any of those cases. In average (geometric average), the instances take about 40 seconds to be solved using model (P4), which is about the 3% of the time required by (P2)-BONMIN and less than the 1% of the time required by (P2)-SCIP. Nevertheless, for some punctual instances that represent the 4.6% of the instances studied, the time and/or memory usage were not enough to get the desired optimality gap. From this it can be stated that for the solution of more realistic instances or even to solve real problems considering continuous equipment sizes, the formulation proposed in this work is much more reliable and faster than the usual and widely studied standard MINLP formulation.

2.6 Conclusions

In this work we present a scalable approach to solve, within reasonable running times and quality assurance requirements, the problem of designing a biotechnological multi-product batch plant that support continuous equipment sizes and discrete host and/or process selection, up to sizes of real instances and that can be applicable to any kind of multi-product batch plant.

The proposed method was proved to be more *numerically stable* than other alternative approaches for the same problem giving true optimal solutions, and in general, faster than

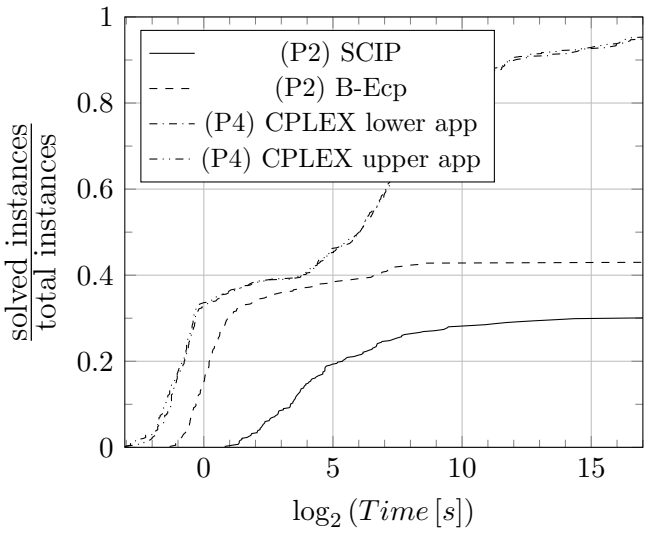


Figure 2.11 – Comparison of performance profiles of the logarithm of running time of "routing" and "sizing instances" solved using models (P2) and (P4) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

other tested approaches. Our method takes advantage of two facts: the continuous relaxation of the feasible region is convex and bounded (which allow us to build, up front, inner or outer approximations of the feasible space, and thus report true upper/lower bounds for each instance); and the fact that mixed-integer linear solvers are much more stable numerically and scalable in size than MINLP algorithms. Also, this approach relies on "off the shelve" optimization and modelling software, which makes it more amiable to practitioners.

To assert our claims, we borrow algorithm comparison tools from the optimization community, which are an interesting form to test the quality of competing algorithms to tackle the same class of problems.

Although, in a real scenario, semi-continuous units such as centrifuges and microfilters, among others, are available only in discrete sizes, unlike tanks that can be built according to customer needs, we feel that the proposed approach is robust enough to consider such issues, however, this was left as a next step in our research. Additionally, to increase the precision of our results for real cases, it is possible to explore a two step approach where after using the proposed method to obtain upper and lower approximations for the objective function we can refine different upper and lower variable bounds making them tighter and perform a re-optimization.

Finally, if true speed is the goal; we know that low-level implementations of dynamic inner/outer approximation can provide further time reductions, however, we feel that this is beyond the scope of this work.

Chapter 3

Optimization of a biotechnological multi-product batch plant design for the manufacture of four different products: a real case scenario

Submitted to Biotechnology and Bioengineering at December 2015.

Abstract

In this work a mixed-integer linear programming (MILP) formulation recently developed by us is used to optimize the equipment sizes of a hypothetical new biotechnological multi-product batch plant, based on information of real known processes for the production of 4 different biotechnological products. Knowing the specific steps conforming the downstream processing of each product, size and time factors were computed and used as parameters to solve the aforementioned MILP reformulation. New constraints were included to permit the selection of some equipment -such as centrifuges and membrane filters- in a discrete set of sizes. For equipment that can be built according to customer needs -such as fermenters and stirred tanks- the original formulation was retained.

Computational results show the ability of this methodology to deal with real data giving reliable solutions for a multi-product batch plant composed of 44 unit operations in a relatively small amount of time showing that in the case studied it is possible to save up to a 66% of the capital investment in equipment given the cost data used.

Keywords: multi-product batch plant, biotechnological products, MILP.

3.1 Introduction

The design of multi-product batch plants using an optimization based approach has been studied for more than 40 years and different approaches to deal with the complexity of the optimization models that result in non-convex mixed-integer non-linear problems (MINLP) goes from the development of new algorithms that are able solve these type of problems (Kocis & Grossmann, 1989; Viswanathan & Grossmann, 1990; Borisenko et al., 2011; Li et al., 2012) to the reformulation of them into mixed-integer linear problems (MILP) (Montagna et al., 2004; Moreno & Montagna, 2011; Sandoval et al., 2016) that can be solved using the known accurate commercial solver CPLEX. In the majority of the cases the aim of these efforts is to be able to find the optimal design of multi-product batch plants in real scenarios which is, according to Barbosa-Póvoa (2007), still a challenge.

In this work the methodology developed by Sandoval et al. (2016) is applied to a real case scenario. The production process of 4 recombinant proteins -Products 1 to 4- are known and with this information a new biotechnological multi-product batch plant is designed. Product 2 and 3 are synthesized as part of inclusion bodies in recombinant *Escherichia coli* while Product 1 and 4 are synthesized by recombinant *Saccharomyces cerevisiae* as intracellular and extracellular products, respectively. Each production process is composed of 8 to 21 processing stages of which about 14 can be shared by 2 or more individual processes. As the equipment involved in each stage may be selected from the equipment available offered by different manufacturers or sized according to the customer needs, a new selection decision was introduced to the routing model presented by Sandoval et al. (2016), which now allows for the selection of a discrete number of possible sizes when it is necessary.

The computational results obtained show that the methodology developed earlier is capable of solving the optimization problem of a real type biotechnological multi-product batch plant -with 44 operational stages- reliably and in a small amount of time.

The rest of this paper is organized as follows. In Section 3.2 known processes to manufacture the 4 proteins studied are presented together to the proposed structure of a multiproduct batch plant that produce them. In Section 3.3 the Problem Formulation and the equations used for parameter estimation is presented. Results are discussed in Section 3.4 and conclusions are presented in Section 3.5.

3.2 Design of a biotechnological multiproduct batch plant

3.2.1 Processes description

In this section the downstream processes of the 4 recombinant proteins studied are described. According to Imperatore & Asenjo (2001) Product 1 is purified in 18 stages, while Products 2 and 3 need 20 and 15 isolation and purification steps, respectively. Product 4 on the other hand, only needs 7 processing stages given its nature of extracellular product. Flowsheets of the 4 processes are shown in Figures 3.1 to 3.4.

Table 3.1 – Production data for the Purification facility (Imperatore & Asenjo, 2001).

Product	Cycle time (h)	Final batch		Suite	Weeks of production	Num. of batches	Max. num. of batches
		size (kg)	volume (L)				
P1	120	6.005	285	1	41	48	49
P2	24	3.777	135	2	15	16	90
				4	20	16	120
P3	32	7.385	285	3	41	95	184
				4	11	25	49
P4	36	14.043	285	2	11	38	44

The original plant is divided into two facilites. In the first, fermentation and primary purification stages are included; in the second, high resolution steps are performed in 4 different suites, each processing one or more products per year. General production data for these suites is presented in Table 3.1. The maximum number of batches per year, in the last column, corresponds to the maximum capacity of the plant given the cycle times, the annual production weeks for each process and 144 work hours per week.

Product 1

Product 1 (P1) is an intracellular protein synthesized by *S. cerevisiae* in a fermentation process that lasts 23 hours. After this step a concentration stage follows performed by a microfilter were cells are collected in the retentate stream for a subsequent homogenization step. In order to wash the homogenization buffer a concentration together with a diafiltration process follows before cellular deactivation in a stirred tank. After that, a new homogenization step is performed followed by a sucrose gradient centrifugation for the collection of a small fraction containing the protein of interest.

At another facility, the purification process is continued with a solubilization process and a centrifugation to concentrate the product. Then the mixture is subjected to a reducing environment followed by a gel filtration chromatography and a reaction of oxidation to refold the protein. Before the next chromatographic step -a reversed phase chromatography- a cross-flow ultrafiltration is performed to concentrate the product mixture. The protein of interest is then precipitated and centrifuged before a dissolution step to be later fed to a gel filtration chromatographic colum. Finally a concentration and diafiltration step is performed.

Product 2

Product 2 (P2) is an intracellular protein synthesized by *E. coli* as inclusion bodies in a fermentation process that lasts 11.5 hours. Cell harvest is performed using a centrifugation step followed by the cell lysis and the harvesting of the inclusion bodies. After 3 consecutive washing and centrifugation steps the protein is subjected to a an oxidative environment followed by the refolding of the protein. Before a cation exchange chromatography a

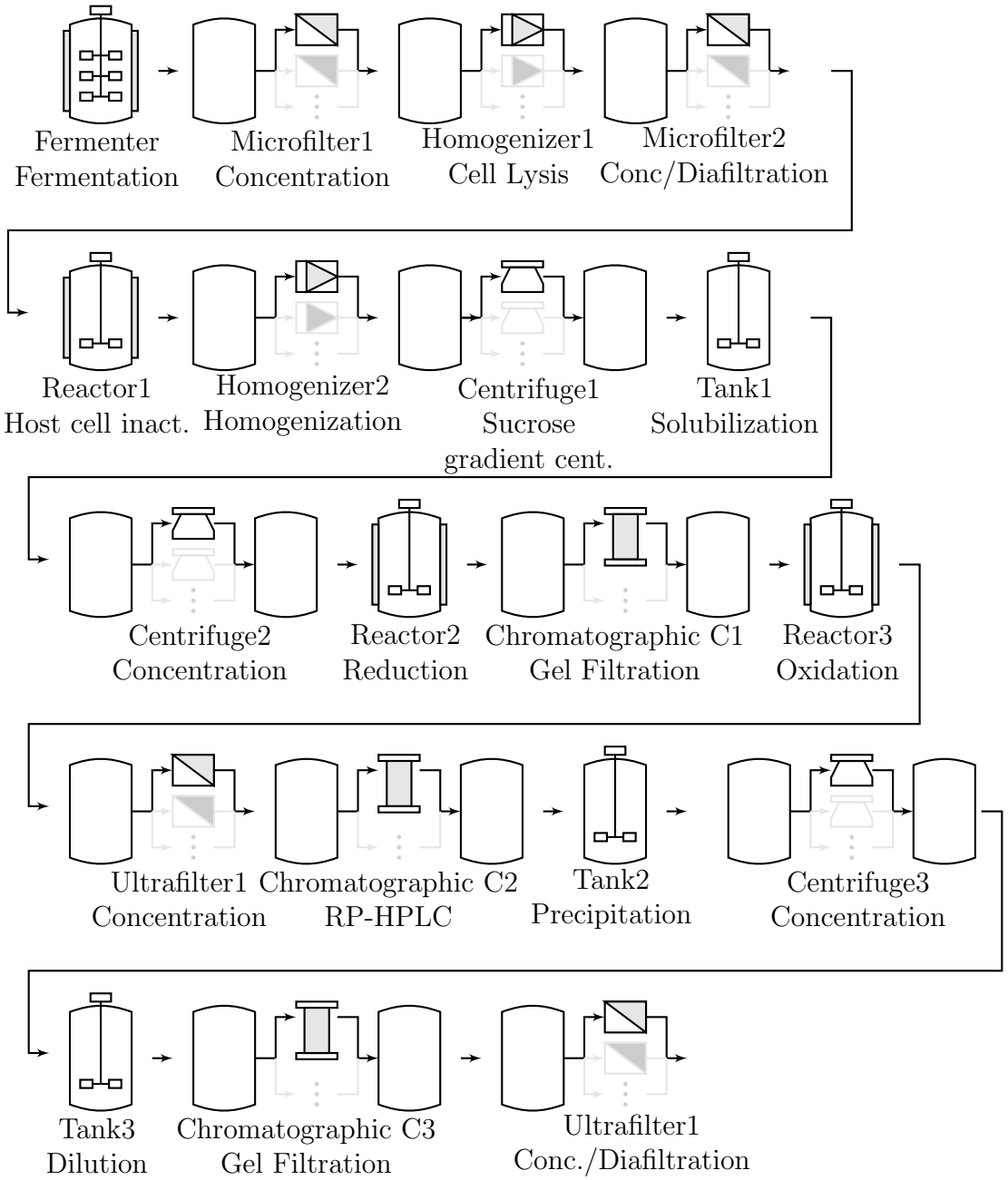


Figure 3.1 – Production process for Product 1, an intracellular protein synthesized in *Saccharomyces cerevisiae.*

microfiltration setp is carried out in order to concentrate the mixture and then, in the second facility, 2 more chromatographic stages are carried out: gel filtration and anion exchange. A dilution and concentration steps followed by a new gel filtration and a hydrophobic interaction chromatography precede a concentration and diafiltration step as the last stage of the process.

Product 3

Product 3 (P3) is also an intracellular protein synthesized by *E. coli* as inclusion bodies but in a fermentation process that lasts 25 hours. Cell harvest is performed using a centrifugation step followed by a cell washing, a cell concentration and followed by cell disruption. A new centrifugation step is carried out in order to capture the inclusion bodies which are then suspended an subjected to a reducing environment. The next stage, in which the protein is refolded, is followed by a concentration, a cation exchange chromatography and a concentration and diafiltration step. In the second facility, three chromatographic stages -anion exchange, hydrophobic interaction and cation exchange- are performed before the last concentration and diafiltration step.

Product 4

Product 4 (P4) is a recombinant protein synthesized by *S. cerevisiae* as an extracellular product in a fermentation process that lasts 122 hours. After fermentation a centrifugation step to discard cells is performed followed by a microfiltration step to clarify the stream before two chromatographic stages: a cationic exchange chromatography and an hydrophobic interaction chromatography.

In the second facility, two ion exchange chromatographies follow the process -anion and cation exchange, respectively- ending with a concentration and diafiltration stage.

3.2.2 Estimation of processes data and plant cost

Equipment sizes

Based on mass balances and the processes data given by Imperatore & Asenjo (2001), different equipment sizes for each productive process were estimated. A 20% safety factor was considered for tanks in general and a 15% for semi-continuous items different from chromatographic columns that were sized using a relatively low capacity usage.

Estimated equipment sizes are given in Tables 3.2 to 3.5. The *Times* column corresponds to the number of times that the stage is used before passing to the next step. V, V^1 and V^2 are the sizes of batch, feed and product tanks, respectively; R is the semi-continuous item size; V^3 the chromatographic column size and T the operation time of the stage in hours. Tanks, reactors and chromatographic columns are sized in L, membrane filtration systems in m^2, centrifuges in $1000 \cdot m^2$ and homogenizers in L/h. For all products, V^2 in the last membrane filtration step is the size of the final product.

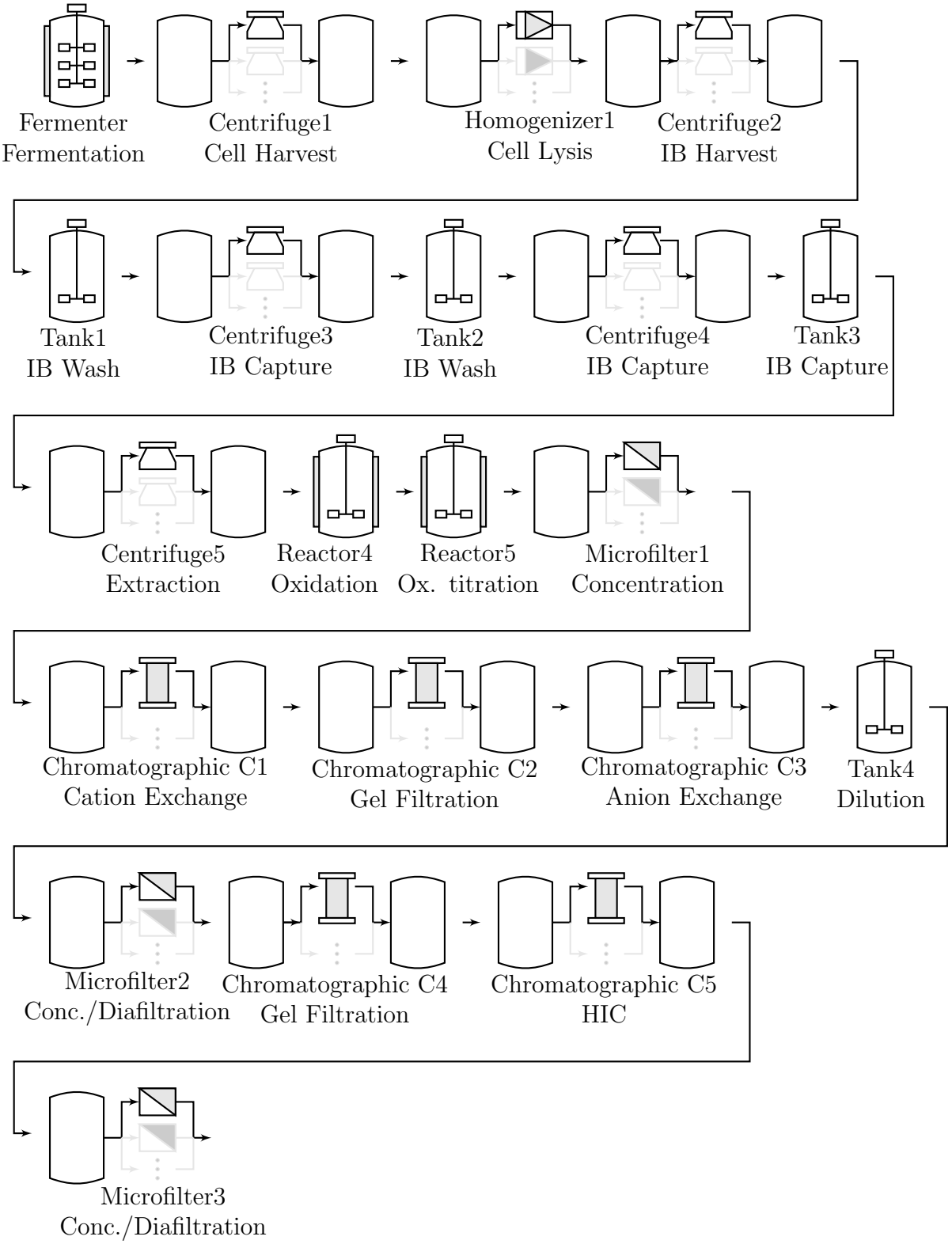


Figure 3.2 – Production process for Product 2, an intracellular protein synthesized in *Escherichia coli.*

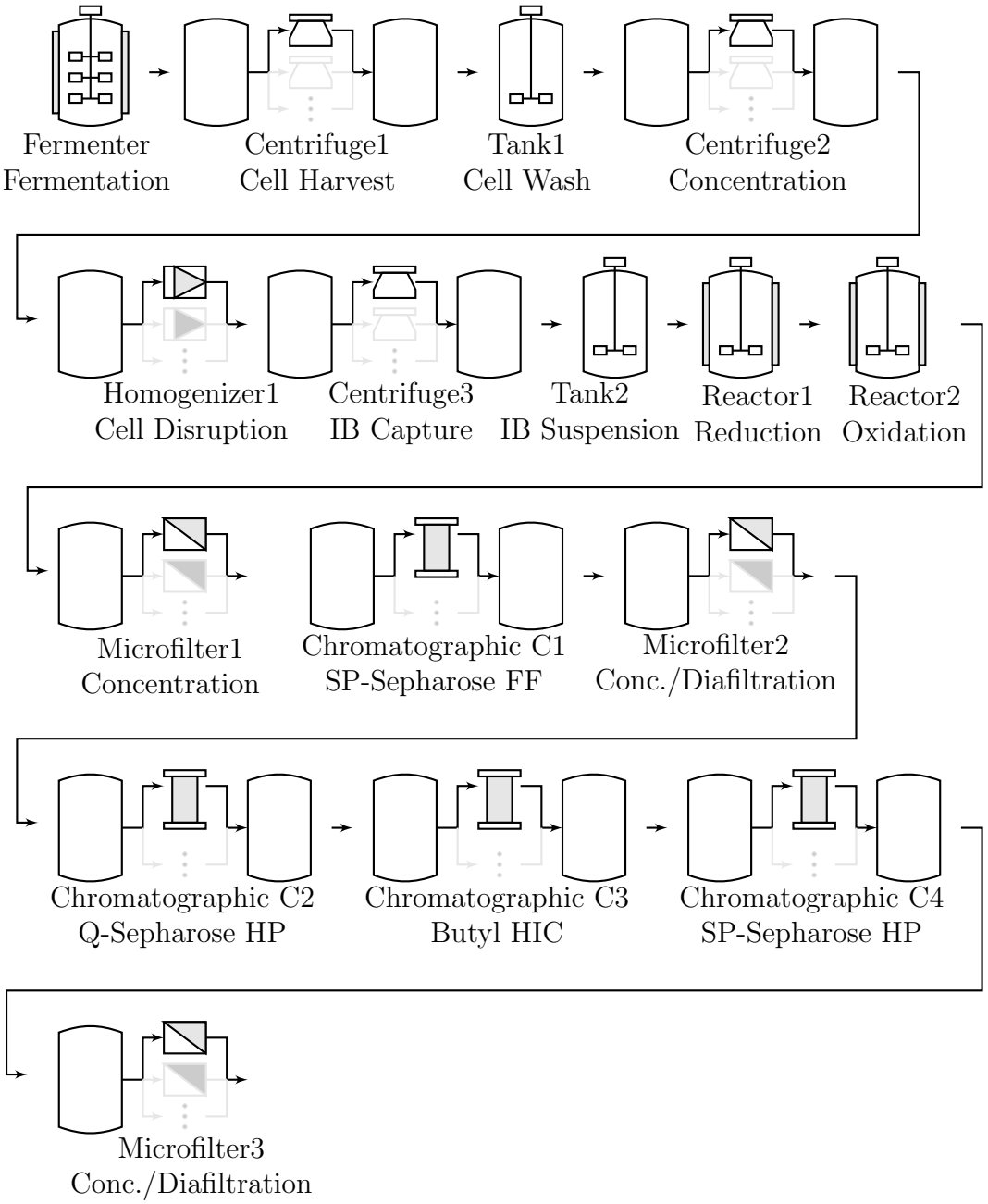


Figure 3.3 – Production process of Product 3, an intracellular protein synthesized in *Escherichia coli.*

Table 3.2 – Original equipment sizes and operation times for the production of Product 1, an intracellular protein synthesized in *S. cerevisiae*.

	Stage		Times	V/V[1]	V[2]	V[3]/R	T
Fermentation	1	fer	1	12500			23
	2	mf1	1	12500		21	2.5
	3	hom1	1	1964		3697	3
	4	mf2	1	1964		17	8
	5	rct1	1	1473			8
	6	hom2	1	1621		4067	3
	7	cnt1	1	17826	463	56	15
Purification	8	tnk1	8	250			2
	9	cnt2	8	250	38	95	5
	10	rct2	4	225			1
	11	chr1	4	225	203	453	24
	12	rct3	4	405			6
	13	uf1	1	1620		3	2
	14	chr2	8	176	159	50	4
	15	tnk2	8	635			1.5
	16	cnt3	8	635	63	399	3
	17	tnk3	5	226			2
	18	chr3	5	226	90	254	24
	19	mf3	1	452	356	21	6

Table 3.3 – Original equipment sizes and operation times for the production of Product 2, an intracellular protein synthesized in *E. coli*.

	Stage		**Times**	**V/V¹**	**V²**	**V³/R**	**T**
Fermentation	1	fer	1	15000			11.5
	2	cnt1	1	15000	1500	15	5
	3	hom1	1	1650		2718	4
	4	cnt2	1	1650	413	39	2
	5	tnk1	1	1650			1
	6	cnt3	1	1650	413	52	1.5
	7	tnk2	1	1650			1
	8	cnt4	1	1650	413	78	1
	9	tnk3	1	1650			1
	10	cnt5	1	1650	413	389	8
	11	rct1	1	743			12
	12	rct2	1	2673			18
	13	mf1	1	2673		16	4
	14	chr1	1	668	601	308	10
Purification	15	chr2	1	601	541	5600	24
	16	chr3	2	271	244	51	10
	17	tnk4	1	1063			2
	18	mf2	1	1063		3	4
	19	chr4	1	750	675	1800	24
	20	chr5	2	338	304	141	5.6
	21	mf3	1	608	169	10	7.5

Table 3.4 – Original equipment sizes and operation times for the production of Product 3, an intracellular protein synthesized in *E. coli*.

	Stage		Times	V/V^1	V^2	V^3/R	T
Fermentation	1	fer	1	12500			25
	2	cnt1	1	12500	1250	15	4
	3	tnk1	1	12500			1
	4	cnt2	1	12500	2500	7	9
	5	hom1	1	2750		4529	4
	6	cnt3	1	2750	344	11	12
	7	tnk2	1	375			1
	8	rct1	1	3125			1
	9	rct2	1	4688			26
	10	mf1	8	586		5	3
	11	chr1	1	1000	1100	462	9
	12	mf2	1	1100		34	6
Purification	13	chr2	1	550	495	283	12
	14	chr3	4	124	111	126	8
	15	chr4	4	111	100	63	8
	16	mf3	1	401	356	31	4

Table 3.5 – Original equipment sizes and operation times for the production of Product 4, an extracellular protein synthesized in *S. cerevisiae*.

	Stage		Times	V/V^1	V^2	V^3/R	T
Fermentation	1	fer	1	9125			122
	2	cnt1	1	9125	1014	1	12
	3	mf1	2	507	425	1	5
	4	chr1	3	283	255	308	15
	5	chr2	6	128	115	173	8
Purification	6	chr3	3	230	207	60	12
	7	chr4	3	207	186	156	8
	8	mf2	1	558	356	9	15

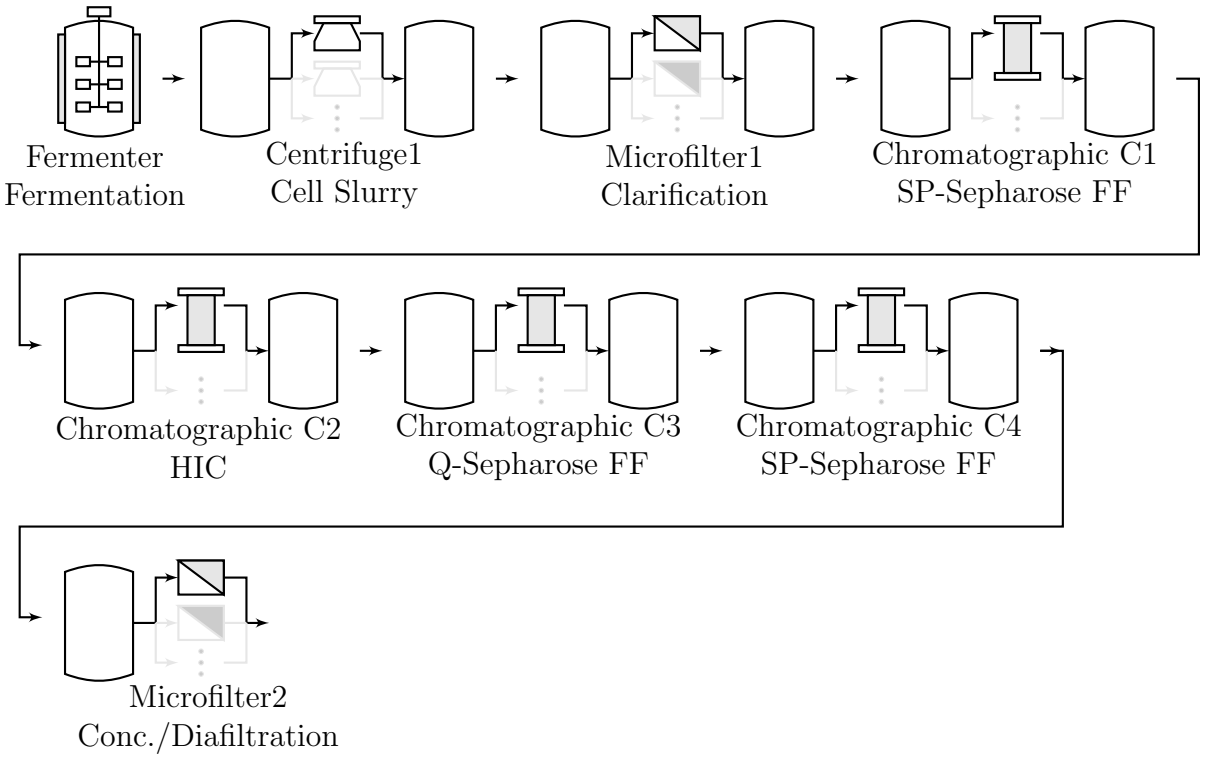


Figure 3.4 – Production process of Product 4, an extracellular protein synthesized in *Saccharomyces cerevisiae*.

Equipment costs

Units suchs as tanks, reactors and fermenters may be built according to customers needs therefore the information needed for their sizing corresponds to cost coeficients for functions of the type $c_j V_j^{\gamma_j}$ and lower and upper size bounds. Original data for fermenter and reactor was taken from Iribarren et al. (2004) and for tanks, from Harrison et al. (2015). The corresponding data is presented in Table 3.6.

For the case of the semi-continuous items costs are determined according to the selected unit offered by the manufacturer. For these cases cost and equipment sizes are given in discrete sets. Data used in this paper is presented in Table 3.7.

3.2.3 Multiproduct batch plant

Given the description of the 4 downstream processes it is possible to suggest that some equipment can be shared by 2 or more processes as is the case of the centrifugation step used for cell harvesting in the isolation processes of Products 2 and 3.

Isolation and purification processes for each individual protein and the identification of the stages than can be shared by 2 or more processes in the proposed multiproduct batch plant is shown in Table 3.8. For comparison purposes, stages are organized as “Fermentation” and “Purification” steps.

Table 3.6 – Cost coefficients and variable bounds needed to size batch units. Costs can be calculated in U.S.$ with the function $c_j V_j^{\gamma_j}$. Data was actualized to year 2012 using CE index: year 2000, 394.1 ; year 2012, 584.6.

Item	Cost coefficients		Size bounds (L)	
	c_j	γ_j	lower	upper
Fermenter	1491	0.6	20	100000
Reactor	1454	0.5	20	100000
Tank	35945	0.1168	200	5000
	492	0.6217	5000	50000

3.3 Problem formulation

3.3.1 Mathematical modeling

The routing formulation proposed by Sandoval et al. (2016) permits the sizing of the equipment according to each of the different type of stages, j, present in a biotechnological multi-product batch plant. In this article, that formulation was modified to account for the selection of semi-continuous items in a discrete set of sizes and costs. The model considers the duplication of units in parallel working in-phase, using continuous variables x_j^1, and out-of-phase, using continuous variables x_j^2; it also accounts for the selection of the global process selecting routes which were defined as different sets of downstream processing stages used to purify a protein given a certain host that synthesizes it. The selection of one route per protein and therefore the use of just some of the possible stages is carried out by the variables z_{ih}^1 and z_j^2. The former is 1 when for product i synthesis host h is selected and 0 otherwise and the last is 1 when stage j is part of at least one of the routes selected and 0 otherwise.

Main variations are presented in the following paragraphs.

Objective function

The objective is to minimize the investment cost of the main equipment of the plant given fixed production targets, d_i, over a time horizon δ. As the original cost functions are non-linear, continuous variables v_j are defined to transform former functions into sets of linear functions. The resulting objective function is given by equation (3.1) that is associated to constraints as (3.2) for batch stages different to stirred tanks, such as fermenters and reactors in subset $\hat{\mathcal{E}}^1$.

$$\min cost = \sum_{j \in \mathcal{E}^1} v_j^1 + \sum_{j \in \mathcal{E}^2 \cup \mathcal{E}^3} \left[v_j^1 + v_j^2 + v_j^3 \right] \tag{3.1}$$

$$v_j^1 \geq \alpha_k^{1j} \left(x_j^1 + x_j^2 + \gamma_j^1 y_j^1 \right) + \left(\beta_k^{1j} - \alpha_k^{1j} b_k^{1j} \right) z_j^2 \qquad \forall j \in \hat{\mathcal{E}}^1, k \in \mathcal{K}^1 \tag{3.2}$$

Table 3.7 – Available cost and equipment sizes for semi-continuous units. Costs are in 1000 U.S.$. Data actualized with CE index: year 1998, 389.5 ; year 2012, 584.6.

Item	**Size**	**Cost**
Micro/ultrafilter (m^2)	5	150
(Harrison et al., 2015)	15	175
	30	210
	55	230
Centrifuge (1000 m^2)	10	60
(Harrison et al., 2015)	50	85
	100	190
	210	505
Homogenizer (L/h)	55	15
(Harrison et al., 2003)	105	38
	290	53
	700	83
	2100	110
	4500	155
Chromatographic column (L)	1	8
(Harrison et al., 2015)	2	10
	4	12
	8	17
	15	20
	35	90
	65	190
	150	240
	250	300
	400	400
	580	600
	1000	800
	1600	1100

Table 3.8 – Downstream processing stages that conform a multiproduct biotecnological batch plant that produces 4 different recombinant proteins synthesized in *E. coli* and *S. cerevisiae* as intra and extracellular products.

	Stage		Description	P1	P2	P3	P4
Fermentation	1	fer	Fermentation	x	x	x	x
	2	mf1	Concentration	x			
	3	cnt1	Cell harvest		x	x	
	4	tnk1	Cell wash			x	
	5	cnt2	Cell concentration			x	
	6	hom1	Cell lysis	x	x	x	
	7	mf2	Conc./ Diafiltration	x			
	8	rct1	Cellular inactivation	x			
	9	hom2	Homogenization	x			
	10	cnt3	Concentration	x			
	11	cnt4	IB harvest		x		
	12	tnk2	IB wash		x		
	13	cnt5	IB capture		x		
	14	tnk3	IB wash		x		
	15	cnt6	IB capture		x	x	
	16	tnk4	IB suspension		x	x	
	17	cnt7	Extraction		x		
	18	rct2	Reduction			x	
	19	rct3	Oxidation		x	x	
	20	rct4	Oxidation titration		x		
	21	mf3	Concentration		x	x	
	22	cnt8	Cell slurry				x
	23	mf4	Clarification				x
	24	chr1	SP-Sepharose FF		x	x	x
	25	mf5	Conc./ Diafiltration			x	
	26	chr2	HIC				x
Purification	27	tnk5	Solubilization	x			
	28	cnt9	Concentration	x			
	29	rct5	Reduction	x			
	30	chr3	Gel filtration	x	x		
	31	rct6	Oxidation	x			
	32	uf1	Concentration	x			
	33	chr4	RP-HPLC	x			
	34	tnk6	Precipitation	x			
	35	cnt10	Concentration	x			
	36	chr5	Q-Sepharose FF		x		x
	37	chr6	Q-Sepharose HP			x	
	38	tnk7	Dilution	x	x		
	39	mf6	Concentration		x		
	40	chr7	Gel filtration	x	x		
	41	chr8	HIC		x	x	
	42	chr9	SP-Sepharose FF				x
	43	chr10	SP-Sepharose HP			x	
	44	mf7	Conc./ Diafiltration	x	x	x	x

where α_k^{1j}, b_k^{1j} and β_k^{1j} are the parameters used to build the set of linear constraints and $\mathcal{K}^1$ the set of cutting points used.

For these mentioned batch stages former cost functions are given by equations as (3.3).

$$cost = c_j^1 \cdot \exp\left(x_j^1 + x_j^2 + \gamma_j^1 y_j^1\right) \qquad \forall j \in \hat{\mathcal{E}}^1 \tag{3.3}$$

And for the particular case of stirred tanks in set $\bar{\mathcal{E}}^1$ -as a singular batch step or as a part of a semi-continuous stage-, found cost data (see Table 3.6) shows a two piecewise function that was modeled using constraint (3.4).

$$cost = c_j^{1s} \cdot \exp\left(x_j^3 + \gamma_j^{1s} y_j^1\right) + c_j^{1b} \cdot \exp\left(x_j^3 + \gamma_j^{1b} y_j^1\right) \qquad \forall j \in \bar{\mathcal{E}}^1 \tag{3.4}$$

where $x_j^3 = x_j^1 + x_j^2$ if the stirred tank is a batch stage or $x_j^3 = x_j^2$ if the tank is part of a semi-continuos stage.

When stirred tanks form part of semi-continuous or chromatographic stages, variables v_j^1 and v_j^2 are constrained with equations as (3.5) and (3.6).

$$v_j^1 \geq \alpha_k^{1sj}\left(x_j^3 + \gamma_j^{1s} y_j^1\right) + \left(\beta_k^{1sj} - \alpha_k^{1sj} b_k^{1sj}\right) z_j^2 \qquad \forall j \in \bar{\mathcal{E}}^1, k \in \mathcal{K}^1 \tag{3.5}$$

$$v_j^1 \geq \alpha_k^{1bj}\left(x_j^3 + \gamma_j^{1b} y_j^1\right) + \left(\beta_k^{1bj} - \alpha_k^{1bj} b_k^{1bj}\right) z_j^2 \qquad \forall j \in \bar{\mathcal{E}}^1, k \in \mathcal{K}^1 \tag{3.6}$$

with α_k^{1bj}, α_k^{1sj}, b_k^{1bj}, b_k^{1sj}, β_k^{1bj} and β_k^{1sj} being the parameters used to build the set of linear constraints and $\mathcal{K}^1$ the set of cutting points used.

Finally, the costs of semi-continuous and chromatographic units are constrained using equations as (3.7). c_j^{3*}, in dimensions equivalent to the cost of other equipment, is calculated as y_j^3 in constraints (3.8) and (3.9).

$$v_j^3 \geq \alpha_k^{3j}\left(x_j^1 + x_j^2 + c_j^{3*}\right) + \left(\beta_k^{3j} - \alpha_k^{3j} b_k^{3j}\right) z_j^2 \qquad \forall j \in \mathcal{E}^2 \cup \mathcal{E}^3, k \in \mathcal{K}^3 \tag{3.7}$$

Constraints

Sizing and timing constraints are basically the same as those presented by Sandoval et al. (2016). Major differences are given by the introduction of binary variables for the selection of the available equipment sizes and costs.

As was previously stated semi-continuous items including chromatographic columns are only available in a discrete number of sizes. In order to achieve this type of selection variables as z_{jk}^5 were introduced to the model which are restricted by constraints (3.8) and (3.9) that are of the same type of those used to define the number of parallel units in each stage.

$$y_j^3 = \sum_{k \in \mathcal{K}^8} z_{jk}^5 \ln(u_k) \qquad \forall j \in \mathcal{E}^* \tag{3.8}$$

$$\sum_{k \in \mathcal{K}^8} z_{jk}^5 = z_j^2 \qquad \forall j \in \mathcal{E}^* \tag{3.9}$$

where u_k is k-th element of the set of available equipment sizes, $\mathcal{U}_{\mathcal{E}^*}$, defined for each subset of stages $\mathcal{E}^*$, that is to say, centrifuges, micro/ultrafilters, homogenizers and chromatographic stages.

Table 3.9 – Data used to estimate size and time factors for Product 1, an intracellular protein syntehsized in *S. cerevisiae*.

Stage	η	X	f	N	J_{conc}	NP	v_s
fer	1.000	1.000	0.000				
mf	0.990	7.000	0.000	0	0.200		
hom	0.900	1.000	0.100			6	
mf	0.991	2.000	0.000	8	0.080		
rct	0.950	1.000	0.500				
hom	0.810	1.000	0.100			8	
cnt	0.850	24.971	10.000				20.0
tnk	0.990	1.000	3.202				
cnt	0.850	6.667	0.000				0.5
rct	0.950	1.000	2.000				
gf	0.950	1.667	0.000				
rct	0.950	1.000	1.000				
mf	0.998	1.148	0.000	0	0.030		
chr	0.980	1.667	0.000				
tnk	0.750	1.000	3.000				
cnt	0.850	9.000	0.000				0.5
tnk	0.980	1.000	1.000				
gf	0.950	1.667	0.000				
mf2	0.993	1.267	0.000	8	0.030		

3.3.2 Size and time factors

Given the knowledge of each individual downstream processing stage and of the corresponding equipment parameters mass balances were solved and constant size and time factors were computed. Size factors for product i in stage j (in g/L) for items in batch stages, S_{ij}; and for feed and product tanks, S^1_{ij} and S^2_{ij}, respectively, are calculated using equation (3.10):

$$S_{ij} = \frac{V_j}{B_i} \tag{3.10}$$

with $V_j\,[L]$ being the actual tank volume in Tables 3.2 to 3.5; and $B_i\,[g]$, equal to Y^4_i in the nomenclature used by Sandoval et al. (2016), the batch size of the product i at the end of the process shown in Table 3.1. This last parameter was estimated using the broth volume in the fermenter (80% of the tank volume), the concentration of product in the fermentation broth -2.5 g/L for Product 1; 2 g/L for Products 2 and 3; 3 g/L for Product 4-, and the overall mass yield given by the product of the yields in each stage, η, shown in Tables 3.9 to 3.12.

Operation time for each non-batch stage is modelled with equation (3.11):

Table 3.10 – Data used to estimate size and time factors for Product 2, an intracellular protein syntehsized in *E. coli*.

Stage	η	**X**	**f**	**N**	$\mathbf{J_{conc}}$	**NP**	$\mathbf{v_s}$
fer	1.000	1.000					
cnt	0.850	10.000					200.0
hom	0.810	1.000	0.100			7	
cnt	0.800	3.000					20.0
tnk	0.950	1.000	2.000				
cnt	0.800	4.000					20.0
tnk	0.950	1.000	2.000				
cnt	0.800	4.000					20.0
tnk	0.950	1.000	2.000				
cnt	0.800	4.000					0.5
rct	0.950	1.000	24.000				
rct	0.950	1.000	2.600				
mf	0.983	30.000		0	0.030		
chr	0.950	1.111					
gf	0.950	1.111					
chr	0.950	1.111					
tnk	0.950	1.000	1.094				
mf	0.997	1.417		0	0.030		
gf	0.950	1.111					
chr	0.950	1.111					
mf2	0.991	3.600		8	0.030		

Table 3.11 – Data used to estimate size and time factors for Product 3, an intracellular protein syntehsized in *E. coli*.

Stage	η	**X**	**f**	**N**	$\mathbf{J_{conc}}$	**NP**	$\mathbf{v_s}$
fer	1.000	1.000					
cnt	0.800	10.000					200.0
tnk	0.980	1.000	9.000				
cnt	0.800	5.000					200.0
hom	0.910	1.000	0.100			7	
cnt	0.950	8.000					20.0
tnk	0.950	1.000	0.091				
rct	0.950	1.000	24.000				
rct	0.950	1.000	0.500				
mf	0.987	14.063		0	0.030		
chr	0.950	0.909					
mf	0.991	2.000		8	0.030		
chr	0.950	1.111					
chr	0.950	1.111					
chr	0.950	1.111					
mf2	0.992	1.125		8	0.030		

Table 3.12 – Data used to estimate size and time factors for Product 4, an extracellular protein syntehsized in *S. cerevisiae*.

Stage	η	**X**	**f**	**N**	$\mathbf{J_{conc}}$	**NP**	$\mathbf{v_s}$
fer	1.000	1.000					
cnt	0.800	9.000					7000.0
mf3	0.992	6.186		3	0.200		
chr	0.950	1.111					
chr	0.950	1.111					
chr	0.950	1.111					
chr	0.950	1.111					
mf2	0.992	1.565		8	0.030		

$$T_{ij} = T_{ij}^{0} + T_{ij}^{1}\frac{B_i}{R_j} \tag{3.11}$$

with $T_{ij}\,[h]$ being the time needed for the step j to process i; $T_{ij}^{0}\,[h]$ the equipment startup time; T_{ij}^{1} a constant time factor or duty factor; and R_j, equal to Y_j^3 in previous nomenclature, the semi-continuous equipment size in units according to the equipment type (see Table 3.7).

Data used to estimate time or duty factors are given in Tables 3.9 to 3.12. In these tables η is a mass yield, X is a volume reduction factor and f a dilution factor. For membrane filtration systems, N is a ratio between the buffer volumes added and the fixed retentate volume; and J_{conc} in $\frac{m^3}{m^2 \cdot h}$ is a flux in the concentration process. For the case of homogenizers, NP is the number of passes through the homogenizer; and finally, v_s in 10^3 mm/h is the settling velocity needed in the case of centrifuges.

Cross-flow filtration

This step is used for the removal of suspended particles, recovery of cells from fermentation broth, and clarification of homogenates containing cell debris and both of them may be followed by the diafiltration of the retentate (stream with larger particles) (Green & Perry, 2007). This last step is essentially a washing step that can be used either to remove more impurities or to increase yield by recovering more product as permeate in clarification process. Diafiltration is performed maintaining constant the level of the feed or retentate tank by the addition of a suitable solvent while the permeate is removed through the membrane (Hearn, 2000).

For a batch operation the design equation of the filtration unit is given by equation (3.12) (Green & Perry, 2007):

$$A = \frac{V_0}{t}\left(\frac{1-\frac{1}{X}}{J_{conc}} + \frac{\frac{N}{X}}{J_{diaf}}\right) \tag{3.12}$$

where A is the membrane area, V_0 is the initial retentate volume, $X < 40$ is the volume reduction factor given by the ratio between the initial retentate volume and the final retentate volume, N is the ratio between the buffer volumes added and the fixed retentate volume. J_{conc} is the flux in the concentration process (volumetric permeate flow rate/membrane area) and J_{diaf} is the flux in the diafiltration process. Values for J_{conc} went from 0.2 in first stages to 0.03 in the final step (Iribarren et al., 2004). J_{diaf} should be smaller than J_{conc} (Hearn, 2000); their value was considered as $\frac{3}{4}J_{conc}$ when necessary.

Operation time can be written from equation (3.12) as equation (3.13):

$$T_{i,mf}\,[h] = T_{i,mf}^{0}\,[h] + \frac{V_0\,[m^3]}{B_i\,[kg]}\left(\frac{1-\frac{1}{X}}{J_{conc}\left[\frac{m^3}{m^2 \cdot h}\right]} + \frac{\frac{N}{X}}{J_{diaf}\left[\frac{m^3}{m^2 \cdot h}\right]}\right)\frac{B_i[kg]}{A_{mf}[m^2]} \tag{3.13}$$

Then, time factor $T_{i,mf}^{1}$, can be computed using equation (3.14):

$$T_{i,mf}^{1}\left[\frac{m^2 \cdot h}{g}\right] = \frac{V_0[m^3]}{B_i[kg]}\left(\frac{1-\frac{1}{X}}{J_{conc}\left[\frac{m^3}{m^2 \cdot h}\right]} + \frac{\frac{N}{X}}{J_{diaf}\left[\frac{m^3}{m^2 \cdot h}\right]}\right) \cdot \frac{1[kg]}{1000[g]} \cdot \frac{1}{0.85} \tag{3.14}$$

Finally, this stage yield is given by equation (3.15a) when this step is used for concentration and by equation (3.15b) when this step is used for clarification. S_i is the observed solute passage that can be computed as the ratio between the concentration of the protein in the permeate and its concentration in the feed stream. If the solute is fully retained $S_i = 0$ and 1 for a fully passing solute (Green & Perry, 2007).

$$\eta_{i,mf} = e^{-S_i(N+\ln X)} \tag{3.15a}$$

$$\eta_{i,mf} = \left(1 - X^{-S_i}\right)\left(1 - e^{-S_i N}\right) \tag{3.15b}$$

Centrifugation

Centrifugation utilizes the density difference between the solids and the surrounding fluid and is often used when solid particles are small and hard to filter (Bell, 1989).

Centrifuges costs are estimated using a Σ factor that is equivalent to a transversal area. Based on the settling velocity of the solid, v_s, and the volumen to be treated, V_0, this factor can be computed using equation (3.16):

$$\Sigma[m^2] = \frac{(V_0\,[m^3])}{(t\,[h])\left(v_s\,\left[\frac{m}{h}\right]\right)} \tag{3.16}$$

Then, time factor $T^1_{i,cnt}$, is computed using equation (3.17):

$$T_{i,cnt}\,[h] = T^0_{i,cnt}\,[h] + \frac{V_0\,[m^3]}{v_s\,\left[\frac{m}{h}\right]\cdot B_i\,[g]}\frac{1}{10^6}\frac{B_i\,[g]}{\Sigma\,[10^3\,m^2]} \tag{3.17}$$

from where

$$T^1_{i,cnt}\,\left[\frac{m^2\cdot h}{g}\right] = \frac{V_0\,[m^3]}{10^3\cdot v_s\,\left[\frac{mm}{h}\right]\cdot B_i\,[g]}\frac{1}{0.85} \tag{3.18}$$

Based on Hatti-Kaul & Mattiasson (2003) the settling velocity was estimated to be $0.2\,mm/h$ for *E. coli* and $7\,mm/h$ for *S. cerevisiae*.

Homogenization

For high pressure homogenizers, the fraction of protein released depends on the operational pressure and the number of passes through the homogenizer (Doran, 2012).

According to Pinto et al. (2001) homogenization time is proportional to the volume fed to the homogenizer, $V_{i,hom}\,(L) = V_{0,hom}\cdot NP_{i,hom}$, and inversely proportional to the homogenizer capacity, $Cap_{i,hom}\,(L/min)$, where $V_{0,hom}$ corresponds to the volume received from previous stage plus a 10% extra volume of lysis buffer if needed (Harrison et al., 2003). With this, equation (3.11) takes the form of equation (3.19):

$$T_{i,hom}\,[h] = T^0_{i,hom}\,[h] + \frac{(V_{0,hom}\,[L])\,(NP_{i,hom})}{B_i\,[g]}\frac{B_i\,[g]}{Cap_{i,hom}\,\left[\frac{L}{h}\right]} \tag{3.19}$$

Therefore time factor $T^1_{i,hom}$ can be calculated using equation (3.20),

$$T^1_{i,hom}\left[\frac{L}{g}\right] = \frac{(V_{0,hom}\,[L])\,(NP_{i,hom})}{B_i\,[g]}\frac{1}{0.85} \tag{3.20}$$

Based on data given in Bell (1989) and Clonis (1990) the number of passes NP was considered to be 7 in the case of *E. coli* and 8 for *S. cerevisiae*. As no especific kinetic data was known, yields were considered to be between 0.8 and 0.9.

Chromatographic separations

As explained by Doran (2012), the basis of chromatography is the selective retardation of solute molecules during passage through a bed of resin particles.

In the processes studied, two major types of liquid chromatographic separations can be found: gel filtration and adsorption chromatography.

Adsorption chromatographic columns are sized taking into account the amount of protein that can be adsorbed into the column, $B_{i,chr}$, which is in turn related to the final batch size (Pinto et al., 2001):

$$B_{i,chr}[kg] = \left(V_{chr}[m^3]\right)(\pi_i)\left(\beta_{i,chr}\left[\frac{kg}{m^3}\right]\right) = \frac{B_i\,[kg]}{\prod_{n=chr}^{NE}\eta_{in}} \tag{3.21}$$

with V_{chr} being the column volume; $\beta_{i,chr}$ the column capacity; π_i the fraction of the maximum capacity that is being used by the adsorbed protein; and NE the number of total stages. From this, the column size factor can be computed with equation (3.22):

$$S^3_{i,chr}\left[\frac{L}{g}\right] = \frac{1}{\pi_i\beta_{i,chr}\prod_{n=chr}^{NE}\eta_{in}} = \frac{V_{chr}\,[L]}{B_i\,[g]} \tag{3.22}$$

The operation time of the chromatographic step is given by (3.23):

$$T_{i,chr} = \frac{V_{feed} + V_{wash} + V_{elution} + V_{regeneration}}{A_{chr}v_{chr}} \tag{3.23}$$

with V_{feed} the volume of solution with protein to be purified, V_{wash} the buffer wash volume used to eliminate proteins not bound to the resin, $V_{elution}$ the volume of buffer used to recover the protein and $V_{regeneration}$ the volume of buffer used to regenerate the column.

Considering the time factors, the operation time can be computed using equation (3.24):

$$T_{i,chr}\;[h] = T^0_{i,chr}\;[h] + \frac{([V_{feed} + V_{elution}]\;[m^3])\,(h\,[m])}{(B_i\,[kg])\left(v_{chr}\;\left[\frac{m}{h}\right]\right)}\frac{B_i\,[kg]}{(A_{chr}\,[m^2])(h[m])} \tag{3.24}$$

If the column hight is set to $0.25\,m$ based on data of Imperatore & Asenjo (2001), and 3 column volume are used to wash and regenerate the resin, respectively, then the constant time factor can be computed as:

$$T^0_{i,chr}\,[h] = \frac{6 \cdot A[m^2] \cdot h[m]}{A[m^2] \cdot v\left[\frac{m}{h}\right]} = \frac{1.5}{v} \tag{3.25}$$

Finally, factor T^1 is calculated using equation (3.26).

$$T^1_{i,chr}\left[\frac{L \cdot h}{g}\right] = \frac{([V_{feed} + V_{elution}]\ [m^3])\,(h\,[m])}{(B_i\,[kg])\,(v_{chr}\ \left[\frac{m}{h}\right])} = \frac{\left(T_{i,chr} - T^0_{i,chr}\right)\ [h] \cdot V_{chr}\,[m^3]}{B_i\,[kg]} \tag{3.26}$$

In gel filtration, a constant time was considered independent of the feed stream. For adsorption chromatography a velocity of 5.5 m/h was considered for ionic-exchange resins and for hydrophobic resins, 4 m/h. These velocities were defined based on GE Healthcare Life Sciences handbooks that can be downloaded from their web page (*http://www.gelifesciences.com/*).

3.3.3 Computational tools / Execution environment

The MILP model studied was coded using the AMPL modelling language and different instances were solved using the commercial CPLEX solver in its version 12.4.0.0. The execution environment was given by a single thread on a Intel(R) Xeon(R) CPU E5620@2.40GHz with an optimality relative gap of 0.1% and 256 cutting points for an *a posteriori* gap up to 0.12%.

3.4 Results and Discussion

3.4.1 Cost of the real plant

The real plant considers both fermentation and purification facilities.

The fermentation facility produces the 4 products in a single productive line, therefore cost was estimated based on a plant configuration as that shown in Table 3.8. The equipment sizes for each stage were set as the maximum size needed among the 4 processes; and for semi-continuos items, that size was selected among those available.

The purification facility cost was estimated as the addition of the 4 suites, each one sharing the same stages as in Table 3.8. For those units, the bigger size among the processes was considered.

Estimated costs are summarized in Table 3.14 along with a comparison between the original and the optimized facilities. These results are discussed in Sections 3.4.3 and 3.4.4.

3.4.2 Number of cutting points and *a posteriori* gaps

As established in previous work (Sandoval et al., 2016) the approach of defining lower and upper approximations for non-linear functions in constraints and the objective function gives true upper and lower approximations for the actual optimal plant cost. Therefore a small *a posteriori* gap between both approximations is expected.

Table 3.13 – Average execution time and relative *a posteriori* gaps for the purification facility instance solved 100 times.

Cutting points	Execution time (s)		gap (%)
	upper approx.	lower approx.	
32	3.78	1.83	0.54
64	5.05	6.46	0.30
128	13.10	13.79	0.13
256	38.65	32.09	0.11

Taking the purification facility as an example to study, different number of cutting points were considered to solve the instance: 32, 64, 128 and 256. Execution times and *a posteriori* gaps were obtained for a set of 100 runs. Results are shown in Table 3.13. 256 cutting points were selected to solve the instances studied since this gives an *a posteriori* gap very close to the solver optimality relative gap and take less than a minute to solve the instance. For bigger instances such as that for the overall multiproduct batch plant, a larger time was obtained but is still small enough (about 6 minutes) considering that the *a posteriori* gap is close 0.11%.

3.4.3 Original purification facility versus corresponding stages in a multi-product batch plant

The original purification facility is divided into 4 suites and the maximum capacity of production for a time horizon of one year is shown in Table 3.1. With this information plus the equipment cost data in Tables 3.6 and 3.7 the optimization of the purification plant defined as a part of the hyphothetical multiproduct batch plant in Table 3.8 was carried out. The optimization results together to a comparison between real and optimized costs are presented in Table 3.14.

As expected, if the optimization is performed over each individual suite the optimized suite is between 20 to 46% less expensive, and more over, if a unique production line is considered, costs can be saved up to a 32% in comparison to the real case. The difference is even bigger if the actual production plant is taken into account with differences up to 68% in the equipment costs (data not shown). In that scenerio an optimization model as the one presented in this article is not just useful in the evaluation of the project but very necessary to design an optimal and not oversized plant.

Table 3.14 – Comparison between the costs of the original and the optimized facilities considering the maximum capacity of the original purification plant and a time horizon given by the number of production weeks. Costs are calculated in U.S.$ based on data for year 2012.

Plant	Original	Optimized	Difference (%)
Fermentation Plant	7 317 406		
Purification Plant	21 487 703	14 561 500	32
Suite 1	3 494 168	1 893 510	46
Suite 2	8 010 547	6 182 200	23
Suite 3	1 511 030	1 117 190	26
Suite 4	8 417 957	6 619 020	22

3.4.4 Optimization of the "global" multiproduct batch plant of 44 stages

The structure of the multiproduct batch plant proposed in section 3.2.3 was determined for a time horizon of 5 904 hours and a production target estimated with the final batch size and the maximum number of batches in Table 3.1. The structure of the plant was optimized to a minimum cost of U.S.$ 26 000 900 and the equipment sizes shown in Table 3.15. Estimated final batch size and cycle times for each product are presented in Table 3.16.

Notice that both duplication, in-phase and out-of-phase, were used in the optimized plant decreasing the large time needed for the fermentation in Product 4 process and permiting small semi-continuous equipment sizes in general.

A comparison to the real plant was not straightforward in this case given that the fermentation plant did not produce the 4 proteins in the one year period. Diferent time horizon, based on production target, were studied and results are shown in Table 3.17. Maximum capacity refers to the production target computed with the maximum number of batches that can be produced in the purification plant (in Table 3.1); and the actual capacity to the actual number of batches produced in the same facility.

Based on real data of cycle times and the aforementioned capacity, fermentation plant should take about 9 436 hours to produce the 4 products. An optimization of the 26 stages conforming that plant over that time horizon and the maximum capacity computed from Table 3.1 gives a cost of U.S.$ 5 378 660, a 26% lower than the calculated based on real sizes.

Finally, real data indicates that the total production of the 4 products given the real used capacity takes about 15 004 hours. Running the optimization based on that data, the plant cost reaches U.S.$ 9 899 380 that is almost a third of the cost calculated for the real plant. This result makes even more evident the benefits of using this type of models for plant design.

Table 3.15 – Optimized structure of the multiproduct batch plant over a time horizon of 5 904 hours. The cost function is equal to U.S.$ 26 000 900.

Stage	X^1	X^2	V^1	V^2	V^3/R
1	1	3	25714.3	.	.
2	1	1	16479.7	.	5
3	1	1	25714.3	2571.43	50
4	1	1	21827.9	.	.
5	1	1	21827.9	4365.58	50
6	1	1	4802.14	.	4500
7	1	1	2589.67	.	15
8	1	1	1942.25	.	.
9	1	1	2136.47	.	700
10	1	1	23501.2	941.141	50
11	1	1	2828.57	942.857	10
12	1	1	2828.57	.	.
13	1	1	2828.57	707.143	10
14	1	1	2121.43	.	.
15	1	1	4802.14	600.267	50
16	1	1	1591.07	.	.
17	3	1	1591.07	397.768	50
18	1	1	16372.3	.	.
19	1	3	24558.4	.	.
20	1	1	35799.1	.	.
21	2	1	35799.1	.	55
22	1	1	18559.3	2062.15	10
23	1	1	2062.15	1728.79	5
24	1	1	1746.38	1921	1000
25	1	1	1921	.	55
26	1	1	1556.07	1400.6	400
27	1	1	3954.67	.	.
28	7	1	3954.67	593.171	50
29	1	1	1779.51	.	.
30	6	1	1779.51	1067.49	1600
31	1	1	2134.99	.	.
32	1	1	2134.99	.	5
33	6	1	1859.75	1115.62	15
34	1	1	4462.5	.	.
35	7	1	4462.5	495.833	50
36	8	1	1400.6	1260.67	15
37	2	1	960.498	864.457	400
38	1	1	1821.66	.	.
39	1	1	1821.66	.	5
40	2	1	1285.85	1157.28	1600
41	1	1	1157.28	1041.56	1000
42	1	1	1260.67	1134.71	400
43	1	1	778.019	700.224	400
44	1	1	1134.71	.	30

Table 3.16 – Final batch size and cycle time of the 4 products produced in the multiproduct batch plant optimized over a time horizon of 5 904 hours.

Product	Final Batch size (kg)	Cycle time (h)
P1	7.916	24
P2	6.410	24
P3	12.831	8.67
P4	28.562	40.67

Table 3.17 – Comparison between the costs of the original and the optimized facilities considering different production targets and time horizon. Costs are calculated in U.S.$ based on data for year 2012.

Capacity	Time horizon (h)	Optimized cost (U.S.$)	Difference (%)
Multi-product Batch Plant. Actual cost U.S.$ 28 805 109.			
Maximum capacity	5 904	26 000 900	10
	23 744	12 291 600	57
Actual capacity	5 904	14 826 000	49
	15 004	9 899 380	66
Fermentation Plant. Actual cost U.S.$ 7 317 406.			
Maximum capacity	5 904	8 629 070	-18
	16 333	5 890 440	20
Actual capacity	5 904	6 163 610	16
	9 436	5 378 660	26
Purification Plant. Actual cost U.S.$ 21 487 703.			
Purification Plant	5 904	14 561 500	32

3.5 Conclusions

In this article a modification of the model presented in a former article has been presented in order to apply the model to the design of a multiproduct batch plant that produces 4 recombinant proteins with known processes. The new model includes discrete costs and equipment sizes for semi-continuous items and preserves the selection of costs and sizes in a continuous range for stirred tanks, reactors and fermenters.

The application of the model permited cost savings up to 66% of the cost of the main equipment, showing that this tool is not just useful but necessary in order to design a plant of the optimal and necessary size.

Lower level implementations (in C, C^{++}) could include the effect of variable cost and production target parameters but this is beyond the scope of this article and was left for a future investigation. In addition, as the time required to solve each instance is less than 30 seconds, there is plenty of space for continuing the addition of new and more complex constraints.

Chapter 4

Main conclusions

In this work a scalable approach that can be applied for the design of real multi-product batch plants is presented.

The proposed methodology was proven to be more numerically stable than other alternative approaches for the same problem giving true optimal solutions, and in general, faster than other tested approaches. The developed method takes advantage of two facts: that the continuous relaxation of the feasible region is convex and bounded (which allows to build, up front, inner or outer approximations of the feasible space, and thus reports true upper/lower bounds for each instance); and the fact that mixed-integer linear solvers are much more stable numerically and scalable in size than MINLP algorithms.

The incorporation of *performance profiles* -borrowed from the optimization community- allowed an easy comparison between the proposed MILP formulations and their former MINLP forms.

The linear nature of the proposed formulation permited the modification for the inclusion of the selection of equipmtent sizes in a discrete set of available sizes. The inclusion of both, discrete and continuous sizes, allowed for a better modeling for a real case scenario where tanks and fermenters may be built according to customer needs; and semi-continuous items such as centrifuges or membrane filtration systems may be found in discrete sets of sizes given by their manufacturers.

The application of the model to the particular case of a plant that produces 4 proteins in a downstream process of 43 stages showed that costs of main equipment may be saved up to a 66%. This shows that this tool is not just useful but necessary to design an optimal and not oversized plant. For other cases the application of this methodology should be straightforward.

Lower level implementations (in C, C^{++}) could permit the study of more complex constraints such as variable costs and production targets, nevertheless this is beyond the scope of this work. In addition to that, as the model takes at most a few minutes to find a global optimum, there is plenty of space to complexify the model for example including operating costs in the objective function.

References

Applegate, D. L., Cook, W., Dash, S., & Espinoza, D. G. (2007). Exact solutions to linear programming problems. *Operations Research Letters*, 35, 693–699.

Asenjo, J. A., Montagna, J. M., Vecchietti, A. R., Iribarren, O. A., & Pinto, J. M. (2000). Strategies for the simultaneous optimization of the structure and the process variables of a protein production plant. *Computers & Chemical Engineering*, 24, 2277–2290.

Barbosa-Póvoa, A. P. (2007). A critical review on the design and retrofit of batch plants. *Computers & Chemical Engineering*, 31, 833–855.

Bell, G. (1989). *Bioseparations: Downstream processing for biotechnology*, volume 11. John Wiley and Sons Inc., New York, NY.

Biegler, L. T., Grossmann, I. E., & Westerberg, A. W. (1997). *Systematic methods for chemical process design*. Prentice Hall, Old Tappan, NJ (United States).

Bonami, P. & Lee, J. (2013). BONMIN Users's Manual.

Borisenko, A., Kegel, P., & Gorlatch, S. (2011). Optimal design of multi-product batch plants using a parallel branch-and-bound method. In V. Malyshkin (Ed.), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6873 LNCS of *Lecture Notes in Computer Science* (pp. 417–430). Springer Berlin Heidelberg.

Bosch, R. & Trick, M. (2005). Integer Programming. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer.

Clonis, Y. D. (1990). *Separation processes in biotechnology. Process affinity chromatography.*, volume 9. CRC Press.

Corsano, G., Aguirre, P. A., & Montagna, J. M. (2009). Multiperiod design and planning of multiproduct batch plants with mixed-product campaigns. *AIChE journal*, 55, 2356–2369.

Corsano, G., Iribarren, O. a., Montagna, J. M., Aguirre, P. a., & González Suarez, E. (2005). Optimization model for the synthesis, design and operation of a batch-semi continuous ethanol plant. In *2nd Mercosur Congress on Chemical Engineering; 4th Mercosur Congress on Process Systems Engineering* (pp. 1–10).

Crowder, H., Johnson, E. L., & Padberg, M. (1983). Solving large-scale zero-one linear programming problems. *Operations Research*, 31, 803–834.

Czyzyk, J., Mesnier, M. P., & More, J. J. (1998). The NEOS Server. *Computational Science & Engineering, IEEE*, 5, 68–75.

Dietrich, B. L., Escudero, L. F., & Chance, F. (1993). Efficient reformulation for 0–1 programs—methods and computational results. *Discrete Applied Mathematics*, 42, 147–175.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2005). A Framework for Multiproduct Batch Plant Design with Environmental Consideration: Application To Protein Production. *Industrial & Engineering Chemistry Research*, 44, 2191–2206.

Dolan, E. D. (2001). *NEOS Server 4.0 administrative guide*. Technical report, Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory.

Dolan, E. D. & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91, 201–213.

Doran, P. M. (2012). *Bioprocess Engineering Principles, 2nd Edition*. Academic Press.

Durand, G. A., Mele, F. D., & Bandoni, J. A. (2012). Determination of storage tanks location for optimal short-term scheduling in multipurpose/multiproduct batch-continuous plants under uncertainties. *Annals of Operations Research*, 199, 225–247.

Durand, G. A., Moreno, M. S., Mele, F. D., Montagna, J. M., & Bandoni, A. (2014). Comparing the performances of two techniques for the optimization under parametric uncertainty of the simultaneous design and planning of a multiproduct batch plant. *Iberoamerican Journal of Industrial Engineering*, 5, 42–54.

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press.

Fumero, Y., Corsano, G., & Montagna, J. M. (2011). Detailed Design of Multiproduct Batch Plants Considering Production Scheduling. *Industrial & Engineering Chemistry Research*, 50, 6146–6160.

Fumero, Y., Corsano, G., & Montagna, J. M. (2012a). Planning and scheduling of multistage multiproduct batch plants operating under production campaigns. *Annals of Operations Research*, 199, 249–268.

Fumero, Y., Montagna, J. M., & Corsano, G. (2012b). Simultaneous design and scheduling of a semicontinuous/batch plant for ethanol and derivatives production. *Computers & Chemical Engineering*, 36, 342–357.

Galiano, F. C. & Montagna, J. M. (1993). Optimal allocation of intermediate storage in multiproduct batch chemical plants. *Mathematical and Computer Modelling*, 18, 111–129.

Geißler, B., Martin, A., Morsi, A., & Schewe, L. (2012). Using piecewise linear functions for solving MINLPs. In *Mixed Integer Nonlinear Programming* (pp. 287–314). Springer.

Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23, 5–48.

Green, D. & Perry, R. (2007). *Perry's Chemical Engineers' Handbook, Eighth Edition*. McGraw Hill professional. McGraw-Hill Education.

Gropp, W. & Moré, J. (1997). Optimization environments and the NEOS server. *Approximation theory and optimization*, (pp. 167–182).

Grossmann, I. E., Caballero, J. A., & Yeomans, H. (2000). Advances in mathematical programming for the synthesis of process systems. *Latin American Applied Research*, 30, 263–284.

Grossmann, I. E. & Guillén-Gosálbez, G. (2010). Scope for the application of mathematical programming techniques in the synthesis and planning of sustainable processes. *Computers & Chemical Engineering*, 34, 1365–1376.

Gupta, S. & Karimi, I. A. (2003). An Improved MILP Formulation for Scheduling Multiproduct, Multistage Batch Plants. *Industrial & Engineering Chemistry Research*, 42, 2365–2380.

Harrison, R. G., Todd, P., Rudge, S. R., & Petrides, D. P. (2003). *Bioseparations Science and Engineering*. Oxford University Press.

Harrison, R. G., Todd, P. W., Rudge, S. R., & Petrides, D. P. (2015). *Bioseparations science and engineering*. Oxford University Press, second edition.

Hatti-Kaul, R. & Mattiasson, B. (2003). *Isolation and Purification of Proteins*. CRC Press.

Hearn, M. T. (2000). *Handbook of Bioseparations*, volume 2. Academic Press.

Imperatore, C. & Asenjo, J. A. (2001). *Case study in a Multiproduct Multifunctional Plant*. Technical report, Chiron Corporation and University of Chile.

Iribarren, O. a., Montagna, J. M., Vecchietti, A. R., Andrews, B., Asenjo, J. a., & Pinto, J. M. (2004). Optimal Process synthesis for the production of multiple recombinant proteins. *Computer Aided Chemical Engineering*, 18, 427–432.

Koch, T. (2004). The final NETLIB-LP results. *Operations Research Letters*, 32, 138–142.

Kocis, G. & Grossmann, I. (1989). Computational experience with dicopt solving MINLP problems in process systems engineering. *Computers & Chemical Engineering*, 13, 307–315.

Kocis, G. R. & Grossmann, I. E. (1988). Global optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis. *Ind. Eng. Chem. Res.*, 27, 1407–1421.

Li, X., Chen, Y., & Barton, P. I. (2012). Nonconvex generalized benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. *Industrial and Engineering Chemistry Research*, 51, 7287–7299.

Margot, F. (2009). Testing cut generators for mixed-integer linear programming. *Mathematical Programming Computation*, 1, 69–95.

Mittelmann, H. (2013). Decision Tree for Optimization Software.

Montagna, J. M., Iribarren, O. a., & Vecchietti, A. R. (2004). Synthesis of Biotechnological Processes Using Generalized Disjunctive Programming. *Industrial & Engineering Chemistry Research*, 43, 4220–4232.

Montagna, J. M., Vecchietti, a. R., Iribarren, O. a., Pinto, J. M., & Asenjo, J. a. (2000). Optimal design of protein production plants with time and size factor process models. *Biotechnology progress*, 16, 228–37.

Moreno, M. S., Iribarren, O. A., & Montagna, J. M. (2009a). Design of Multiproduct Batch Plants with Units in Series Including Process Performance Models. *Industrial & Engineering Chemistry Research*, 48, 2634–2645.

Moreno, M. S., Iribarren, O. A., & Montagna, J. M. (2009b). Optimal design of multiproduct batch plants considering duplication of units in series. *Chemical Engineering Research and Design*, 87, 1497–1508.

Moreno, M. S. & Montagna, J. M. (2007a). New alternatives in the design and planning of multiproduct batch plants in a multiperiod scenario. *Industrial & engineering chemistry research*, 46, 5645–5658.

Moreno, M. S. & Montagna, J. M. (2007b). Optimal Simultaneous Design and Operational Planning of Vegetable Extraction Processes. *Food and Bioproducts Processing*, 85, 360–371.

Moreno, M. S. & Montagna, J. M. (2011). Multiproduct batch plants design using linear process performance models. *AIChE Journal*, 57, 122–135.

Moreno, M. S. & Montagna, J. M. (2012). Multiperiod production planning and design of batch plants under uncertainty. *Computers & Chemical Engineering*, 40, 181–190.

Moreno, M. S., Montagna, J. M., & Iribarren, O. A. (2007). Multiperiod optimization for the design and planning of multiproduct batch plants. *Computers & Chemical Engineering*, 31, 1159–1173.

Moreno-Benito, M., Espuña, A., & Puigjaner, L. (2014). Flexible batch process and plant design using mixed-logic dynamic optimization: single-product plants. *Industrial & Engineering Chemistry Research*, 53, 17182–17199.

Nikolopoulou, A. & Ierapetritou, M. G. (2012). Optimal design of sustainable chemical processes and supply chains: A review. *Computers & Chemical Engineering*, 44, 94–103.

Nowak, I. (2005). Overview of Global Optimization Methods. In *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming* (pp. 121–128). Springer.

Pinto, J. M., Montagna, J. M., Vecchietti, a. R., Iribarren, O. a., & Asenjo, J. a. (2001). Process performance models in the optimization of multiproduct protein production plants. *Biotechnology and bioengineering*, 74, 451–465.

Pinto-Varela, T., Barbosa-Povoa, A. P. F. D., & Novais, A. Q. (2009). Design and Scheduling of Periodic Multipurpose Batch Plants under Uncertainty. *Industrial & Engineering Chemistry Research*, 48, 9655–9670.

Ponsich, A., Azzaro-Pantel, C., Domenech, S., & Pibouleau, L. (2007). Mixed-integer nonlinear programming optimization strategies for batch plant design problems. *Industrial & engineering chemistry research*, 46, 854–863.

Ravemark, D. E. & Rippin, D. W. T. (1998). Optimal design of a multi-product batch plant. *Computers & Chemical Engineering*, 22, 177–183.

Rebennack, S., Kallrath, J., & Pardalos, P. M. (2011). Optimal storage design for a multi-product plant: A non-convex MINLP formulation. *Computers & chemical engineering*, 35, 255–271.

Reklaitis, G. V. (1990). Progress and issues in computer-aided batch process design. In *Proc. of the Third Int. Conf. on Foundations of Computer-Aided Process Design, Snowmass Village, Colorado, July* (pp. 10–14).: CACHE/Elsevier.

Rippin, D. (1993). Batch process systems engineering: A retrospective and prospective review. *Computers & Chemical Engineering*, 17, S1–S13.

Robinson, J. D. & Loonkar, Y. R. (1972). Minimizing capital investment for multi-product batch-plants. *Process Technology*, 17, 861.

Salomone, H. E., Montagna, J. M., & Iribarren, O. A. (1994). Dynamic Simulations in the Design of Batch Processes. *Computers chem Engng*, 18, 191–204.

Samsatli, N. J. & Shah, N. (1996a). An Optimization Based Design Procedure for Biochemical Processes: Part I: Preliminary Design and Operation. *Food and Bioproducts Processing*, 74, 221–231.

Samsatli, N. J. & Shah, N. (1996b). An Optimization Based Design Procedure for Biochemical Processes: Part II: Detailed Scheduling. *Food and Bioproducts Processing*, 74, 232–242.

Sandoval, G., Espinoza, D., Figueroa, N., & Asenjo, J. A. (2016). MILP reformulations for the design of biotechnological multi-product batch plants using continuous equipment sizes and discrete host selection. *Computers & Chemical Engineering*, 84, 1–11.

Srinivasan, B., Bonvin, D., Visser, E., & Palanki, S. (2003). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Computers & Chemical Engineering*, 27, 27–44.

Vásquez-Alvarez, E., Lienqueo, M. E., & Pinto, J. M. (2001). Optimal Synthesis of Protein Purification Processes. *Biotechnology Progress*, 17, 685–696.

Verderame, P. M., Elia, J. A., Li, J., & Floudas, C. A. (2010). Planning and scheduling under uncertainty: a review across multiple sectors. *Industrial & engineering chemistry research*, 49, 3993–4017.

Vielma, J. P. (2013). Mixed integer linear programming formulation techniques.

Viswanathan, J. & Grossmann, I. (1990). A combined penalty function and outer-approximation method for MINLP optimization. *Computers & Chemical Engineering*, 14, 769–782.

Voudouris, V. T. & Grossmann, I. E. (1992). Mixed-integer linear programming reformulations for batch process design with discrete equipment sizes. *Industrial & Engineering Chemistry Research*, 31, 1315–1325.

Voudouris, V. T. & Grossmann, I. E. (1993). Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations. *Industrial & Engineering Chemistry Research*, 32, 1962–1980.

Wang, Z., Jia, X.-P., & Shi, L. (2010). Optimization of multi-product batch plant design under uncertainty with environmental considerations. *Clean Technologies and Environmental Policy*, 12, 273–282.

Yi, G. & Reklaitis, G. V. (2011). Optimal design of multiperiod batch-storage network including transportation processes. *AIChE Journal*, 57, 2821–2840.

Appendices

Appendix A

Published article

Computers and Chemical Engineering 84 (2016) 1–11

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

MILP reformulations for the design of biotechnological multi-product batch plants using continuous equipment sizes and discrete host selection

G. Sandoval [a], D. Espinoza [b], N. Figueroa [c], J.A. Asenjo [a,*]

[a] Center of Biotechnology and Bioengineering, CeBiB, Departamento de Ingeniería Química y Biotecnología, Universidad de Chile, Santiago, Chile
[b] Departamento de Ingeniería Industrial, FCFM, Universidad de Chile, Santiago, Chile
[c] Instituto de Economía, Pontificia Universidad Católica de Chile, Santiago, Chile

ARTICLE INFO

Article history:
Received 20 August 2014
Received in revised form 24 July 2015
Accepted 1 August 2015
Available online 19 August 2015

Keywords:
Multi-product batch plant
MINLP
MILP
Production path

ABSTRACT

In this article we present a new approach, relying on mixed-integer linear programming (MILP) formulations, for the design of multi-product batch plants with continuous sizes for processing units and host selection. The main advantage of the proposed approach is its scalability, that allows us to solve, within *reasonable* precision requirements, realistic instances. Furthermore, we show that many other alternatives are either numerically unstable (for the problem sizes that we are interested in), unable to solve large instances, or much slower than the proposed method. We present extensive computational experiments, which show that we are able to solve almost all tested instances, and, in average, we are ten times faster than alternative approaches. As we use a high level implementation language (AMPL) we should get further time improvements if lower level implementations are used (C, C++).

Reproducibility of our results can be tested using our models and data available on-line at BPLIB.[1]

1. Introduction

Conventional multi-product batch process literature using an optimization-based approach model the design and synthesis of such plants with Mixed-Integer Non-Linear Programming (MINLP) formulations (Floudas, 1995). The usual objective is to minimize the investment cost subject to the fulfillment of the production targets of a given set of products. Major drawbacks are given by the combinatorial nature of mixed-integer programming and possible nonconvexities due to non-linearities. In computational optimization numerical issues of these formulations given by rounding errors, numerical instabilities and approximation errors are well-documented (Goldberg, 1991; Koch, 2004; Margot, 2009; Vielma, 2013).

Since Robinson and Loonkar (1972) different procedures have been proposed to tackle these problems (Reklaitis, 1990; Rippin, 1993; Barbosa-Póvoa, 2007; Verderame et al., 2010; Nikolopoulou and Ierapetritou, 2012) but a method that is more efficient for a particular example is hardly predictable (Ponsich et al., 2007) and nowadays the development of effective solution approaches and algorithms remains very necessary (Grossmann and Guillén-Gosálbez, 2010).

The logarithmic change of variables proposed by Kocis and Grossmann (1988) linearizes most of the functions and leads to a convex MINLP problem, approach used by Ravemark and Rippin (1998) and Montagna et al. (2000) among others. Another approach chosen by Pinto et al. (2001) and Ponsich et al. (2007) among others is the use of specially designed solvers which can usually find good feasible solutions by the use of heuristic procedures (Grossmann et al., 2000). In practice the best *off the shelf* solvers for this kind of problems are the open source codes BONMIN and SCIP and the commercial solvers BARON and DICOPT that stand out in Mittelmann's benchmarks for optimization software (Mittelmann, 2013). Nevertheless none of them guarantee convergence to a global optimum, converging in some instances to local optima or not converging altogether. For the particular case of BARON and DICOPT performance failures are reported for non-convex models (Ponsich et al., 2007; Rebennack et al., 2011; Li et al., 2012); nevertheless even in cases where theoretically the algorithms work, we found that in practice, they do not converge to the global optimum. We have run precise experiments that demonstrate these failures in convex MINLP formulations (see Section 2).

It is a fact that there is a huge gap between Mixed-Integer Linear Programming (MILP or MIP) and MINLP solvers technology (Nowak,

* Corresponding author. Tel.: +56 229784723; fax: +56 226991084.
E-mail addresses: gsandova@ing.uchile.cl (G. Sandoval), daespino@dii.uchile.cl (D. Espinoza), nicolasf@uc.cl (N. Figueroa), juasenjo@ing.uchile.cl (J.A. Asenjo).

[1] Available at: http://www.dii.uchile.cl/ daespino/

http://dx.doi.org/10.1016/j.compchemeng.2015.08.001
0098-1354/

Nomenclature

Indices

h	host
i	product
j	stage
m	number of duplicated units
UP,LO	upper and lower bounds

Sets

$\mathcal{E}^1$	set of batch stages, $\subset \mathcal{E}$
$\mathcal{E}^2$	set of semi-continuous stages, $\subset \mathcal{E}$
$\mathcal{E}^3$	set of chromatographic stages, $\subset \mathcal{E}$
$\mathcal{E}$	set of all stages
$\mathcal{H}$	set of hosts
$\mathcal{I}$	set of products
$\mathcal{M}$	set of available units operating in parallel in-phase or out-of-phase
$\mathcal{R}$	set of routes: stages j needed to process the product i synthesized by host h
$\mathcal{U}$	set of available hosts h for product i synthesis

Parameters

δ	time horizon
c_j^n, γ_j^n	cost coefficients related to Y_j^n with $n \in \{1, 2, 3\}$
d_i	production target for product i
S_{ij}^n, S_{ij}^n	size factor for product i in stage j related to Y_j^n. $s_{ij}^n = \ln(S_{ij}^n) n \in \{1, 2, 3\}$
T_{ij}^0, T_{ij}^0	time factor for product i in the batch or chromatographic stage j. $t_{ij}^0 = \ln(T_{ij}^0)$
T_{ij}^1, t_{ij}^1	time factor for product i in the semi-continuous or chromatographic stage j. $t_{ij}^1 = \ln(T_{ij}^1)$

Variables

v	slack variable
X_j^n, X_j^n	number of units operating in parallel in-phase (X_j^1) and out-of-phase (X_j^2) in stage j. $x_j^n = \ln(X_j^n)$
Y_j^1, y_j^1	volumetric capacity for batch units and retentate or feed tank for semi-continuous or chromatographic stages. $y_j^1 = \ln(Y_j^1)$
Y_j^2, Y_j^2	volumetric capacity for permeate of product tanks for semi-continuous or chromatographic stages. $y_j^2 = \ln(Y_j^2)$
Y_j^3, Y_j^3	capacity of semi-continuous items. $y_j^3 = \ln(Y_j^3)$
Y_i^4, Y_i^4	batch size for final product i. $y_i^4 = \ln(Y_i^4)$
Y_i^5, Y_i^5	cycle time for product i. $y_i^5 = \ln(Y_i^5)$
z_{ih}^1	binary variable: 1 if protein i is synthesized by host h; 0 otherwise
z_j^2	binary variable: 1 if stage j is used to process at least one of the products
z_{jm}^3	binary variable: 1 if m units are operating in parallel in-phase in stage j; 0 otherwise
z_{jm}^4	binary variable: 1 if m units are operating in parallel out-of-phase in stage j; 0 otherwise

2005). Nowadays mixed-integer linear techniques are fast, robust and able to provide solutions to problems with up to millions of variables (Geißler et al., 2012). Taking advantage of this Voudouris and Grossmann (1992) used reformulation schemes to develop MILP models for the preliminary design of multi-product batch plants, introducing binary variables for the selection of discrete available equipment sizes. From this point, to the design decisions other were included as synthesis, production planning and scheduling (Voudouris and Grossmann, 1993); design and planning in a multiperiod scenario (Moreno and Montagna, 2007); design of multi-product batch plants considering duplication of units in series (Moreno et al., 2009) and the design and planning of multi-product batch plants using mixed-product campaigns (Corsano et al., 2009). Most recently these MILP formulations have been used to account for the design and scheduling of this type of plants (Fumero et al., 2011, 2012a,b) and for the design under uncertainty considering different types of decisions (Durand et al., 2012, 2014; Moreno and Montagna, 2012; Moreno-Benito et al., 2014).

A key feature in these design problems is the use of *Big-M* constraints to account for selection decisions despite being problematic (Bosch and Trick, 2005). Some authors that have included this type of constraints in their formulations are Gupta and Karimi (2003), Corsano et al. (2009), Moreno et al. (2009), Moreno and Montagna (2012). Obviously, these authors have found that the value of the *Big-M* parameters has a tremendous impact on the solution time; see for example Moreno et al. (2007). In addition it has been proven experimentally that other methods, as the convex hull formulation presented by Montagna et al. (2004) are better to account for selection decisions.

In this paper we develop a robust methodology to solve the design problem of a biotechnological multi-product batch plant in situations where equipment can be manufactured according to customer needs, as fermentors or tanks in general. To do that, we develop a MILP formulation which does not rely on the use of *Big-M* constraints and does not use a discrete range of equipment sizes. To do that we use four basic techniques (see Fig. 1): First, an extension of the non-linear (but convex) formulation proposed by Kocis and Grossmann (1988) is applied. Secondly, to deal with non-linear convex inequalities *a priori* we constructed linear outer (or inner) approximations of them which allow us to compute (*a posteriori*) true feasible solutions and lower (or upper) bounds.

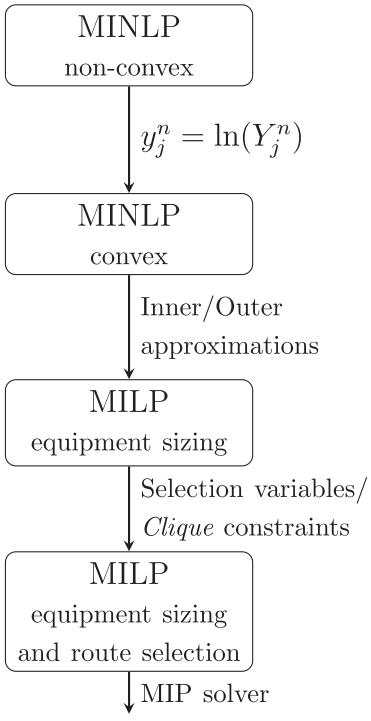


Fig. 1. Basic techniques used to model synthesis and design decisions considering continuous equipment sizes and discrete host selection.

Thirdly, to deal with integer variables, we used advanced reformulation techniques coming from the mixed-integer-programming literature (*clique* constraints). Finally, once the initial problem is transformed into a standard mixed-integer programming problem, it is possible to take advantage of mature commercial MIP solvers.

This approach, at least in our experiments, is more stable numerically, scalable, and faster to solve than current alternatives and can deal with the more general problem of jointly selecting equipment sizes and alternative production paths for multiple products. Using our approach, it is possible to quickly and accurately compute solutions at any desired precision level. In our extensive computational experiments (see Fig. 11) we found that current non-linear solvers only solved 43% of the instances generated for this study, while our approach was able to solve over 95% of the studied instances in a running time that, on average, was more than ten times faster than MINLP solvers in equivalent and standard MINLP formulations. To make these comparisons we introduce the performance profiles; a methodology borrowed from the optimization literature.

The rest of this paper is organized as follows. In Section 2 typical drawbacks found by a commonly used MINLP solver and the standard MINLP formulation is presented. In Section 3 classic and novel formulations for the design problem are described. Relevant information about the methodology used to benchmark different formulations and to avoid numerical instabilities is given in Section 4 and computational results are presented and discussed in Section 5. Finally, the conclusions are presented in Section 6.

2. Current limitations

Our main objective is finding a robust and scalable methodology for the design of biotechnological multi-product batch plant considering equipment sizing (design decisions) and selecting the downstream processing stages (synthesis decisions). Given the complexities that to date have been added to the original design problem we decided to go back to the problem studied by Iribarren et al. (2004) where only design and synthesis decisions are modeled. In their paper they designed a biotechnological batch plant for the production of four recombinant proteins, i, where each can be synthesized by two different hosts, h, having four microorganisms in total. In addition to that three of the fifteen processing stages, j, may be performed by two different unit operations, d. In their formulation they used constant size (S_{ijdh}) and time (T_{ijdh}) factors to model each stage; considered duplication of units in parallel in-phase, G_{jd}, and out-of-phase, M_{jd}, in order to diminish either the equipment sizes V_j or cycle times TL_i, respectively, and used *Big-M* constraints to account for the selection of hosts and equipment.

As a correctness test, we took the example presented in Iribarren et al. (2004) and splitted into 16 different instances that only allow equipment selection. Then we tested two different yet equivalent formulations based on their model but removing host selection[2].

In the first (C1) the selection of hosts was eliminated by limiting the set of available hosts, H_i, to just one per protein, and in the second (C2), by setting the values of the selection binary variables to 1 for the selected hosts and 0 for those non-selected. If the solution is being found by the solvers, we should observe two things:

(a) both model formulations (C1) and (C2) give the same solution, and
(b) the minimum of the separated instances is equivalent to the global minimum of the problem with host selection.

[2] To further isolate the results obtained from problems with non-standard local-settings, the GAMS modelling language was used and the experiments were run in the NEOS server (Gropp and Moré, 1997; Czyzyk et al., 1998; Dolan, 2001) available in http://www.neos-server.org.

Table 1
Comparison of the number of constraints and variables of some selected instances solved using models that account for selection with *Big-M* constraints – models (C1) and (C2) – and a classic formulation for design decisions only, (P1). All instances were solved using the DICOPT solver.

	Variables		Constraints		Status of solution
	Discrete	Continuous	Linear	Non-linear	
(C1)	480	185	303	10	Incorrect
(C2)	512	323	682	18	Incorrect
(P1)	360	71	179	9	Correct

Contrary to what we expected differences in the objective function value for both (C1) and (C2) formulations went from 1% to 78% in the 16 instances studied (data not shown), and even more striking, the solver finds a local minimum which is worse than those found for most instances without host selection.

These numerical instabilities seem to be aggravated with size since it is known that DICOPT works fine for small instances. Situation in accordance to the results obtained by Ponsich et al. (2007). In order to show these differences in sizes we built Table 1 to compare the number of constraints and variables involved in the smaller instances of the cases (C1) and (C2), that were incorrectly solved according to the aforementioned results, with the size of a smaller instance that was correctly solved by a classic formulation (P1) that solves an equipment sizing problem similar to that presented by Iribarren et al. (2004), but with no selection of hosts or equipments, and using DICOPT solver. This last formulation is presented en Section 3.1.1.

3. Problem formulation

Two major contributions are presented in this section. First, *clique* constraints are introduced to formulate the discrete part of the model allowing the selection of the production path without the use of *Big-M* constraints, in models (P2) and (P4). Second, a new approach, in Section 3.2, to handle non-linearities using standard reformulation techniques from the optimization field that permits the use of linear solvers leading to more reliable results and faster computing time. The relation among the four different models studied is shown in Fig. 2.

3.1. MINLP formulation

3.1.1. The equipment-sizing problem (P1)

In this section we present the most basic formulation for the design of biotechnological multi-product batch plants as only equipment sizing and duplication of units in parallel are considered.

The plant consists of a sequence of batch, semi-continuous and chromatographic stages used to manufacture different products i;

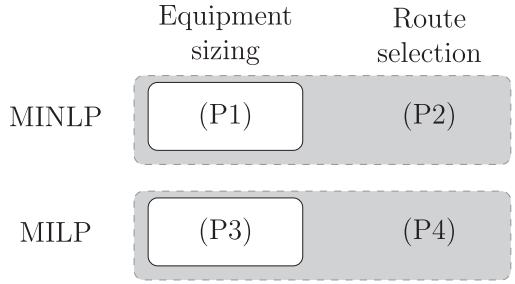


Fig. 2. Formulations compared in this article. Model (P1) is the most basic formulation that only includes design decisions. Model (P2) includes the selection of the downstream processes without the use of *Big-M* constraints. Models (P3) and (P4) are the transformed models of (P1) and (P2), respectively, using our proposed inner and outer approximations.

where semi-continuous as well as chromatographic stages are composed by the semi-continuous items plus feed and product tanks. At each stage j there are X_j^2 groups of units operating in parallel out-of-phase and each group is conformed by X_j^1 units operating in-phase. For semi-continuous or chromatographic stages feed and product tanks can only be duplicated out-of-phase. Single production campaigns are considered and batches are transferred from one stage to the next without delay (zero wait policy).

The objective is to minimize the investment costs of main equipments of the plant (see Eq. (1)) given fixed production targets, d_i, over a time horizon δ.

$$\min \text{cost} = \sum_{j\in\mathcal{E}^1} X_j^1 X_j^2 \left(c_j^1 {Y_j^1}^{\gamma_j^1}\right) + \sum_{j\in\mathcal{E}^2\cup\mathcal{E}^3} \left[X_j^2 \left(c_j^1 {Y_j^1}^{\gamma_j^1}\right) + X_j^2 \left(c_j^2 {Y_j^2}^{\gamma_j^2}\right) + X_j^1 X_j^2 \left(c_j^3 {Y_j^3}^{\gamma_j^3}\right)\right] + v\rho\delta \quad (1)$$

Variables $Y_j^{\cdot}$ represent the different equipment sizes. Parameters $c_j^{\cdot}$ and $\gamma_j^{\cdot}$ are cost coefficients distinctive for each kind of equipment and v is a slack variable included to assure feasibility (Montagna et al., 2004).

Making the change of variables introduced by Kocis and Grossmann (1988) we get the new objective function (2).

$$\min \text{cost} = \sum_{j\in\mathcal{E}^1} c_j^1 \exp\left(x_j^1 + x_j^2 + y_j^1\gamma_j^1\right) + \sum_{j\in\mathcal{E}^2\cup\mathcal{E}^3} \left[c_j^1 \exp\left(x_j^2 + y_j^1\gamma_j^1\right) + c_j^2 \exp\left(x_j^2 + y_j^2\gamma_j^2\right) + c_j^3 \exp\left(x_j^1 + x_j^2 + y_j^3\gamma_j^3\right)\right] + v\rho\delta \quad (2)$$

At each stage and for each product the size of the units must allow the processing of the incoming batch which can be splitted among X_j^1 units to not surpass the upper bound capacity of the equipment. In batch stages this constraint can be written as Eq. (3a); convexified in Eq. (3b).

$$Y_j^1 \geq \frac{S_{ij}^1 Y_i^4}{X_j^1} \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^1 \quad (3a)$$

$$y_j^1 + x_j^1 \geq s_{ij}^1 + y_i^4 \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^1 \quad (3b)$$

As in semi-continuous or chromatographic stages duplication is allowed just for semi-continuous items, feed and product tanks are sized using constraints (4) and (5).

$$y_j^1 \geq s_{ij}^1 + y_i^4 \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \cup \mathcal{E}^3 \quad (4)$$

$$y_j^2 \geq s_{ij}^2 + y_i^4 \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \cup \mathcal{E}^3 \quad (5)$$

Chromatographic columns have to process the incoming batch and both duplication in-phase and out-of-phase are allowed. Duplication in-phase is modeled in size constraint (6) since this permits smaller units and duplication out-of-phase is reflected in time constraints.

$$y_j^3 + x_j^1 \geq s_{ij}^3 + y_i^4 \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^3 \quad (6)$$

The cycle time for each product i, Y_i^5, is defined as the time elapsed between the production of two consecutive batches and is given by the larger operating time, T_{ij}, among the stages in the process. This time can be decreased if a duplication of units out-of-phase is used:

$$Y_i^5 \geq \frac{T_{ij}}{X_j^2} \quad \forall i \in \mathcal{I}, j \in \mathcal{E} \quad (7)$$

As batch stages operate for a fixed time, T_{ij}^0, cycle time constraint in its convex form is given by Eq. (8):

$$y_i^5 + x_j^2 \geq t_{ij}^0 \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^1 \quad (8)$$

Semi-continuous stages, on the other hand, operate during a time that depends on the final batch size, Y_i^4. For those stages the cycle time is constrained as in Eq. (9).

$$Y_i^5 \geq \frac{T_{ij}^1 \frac{Y_i^4}{X_j^1 Y_j^3}}{X_j^2} \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \quad (9a)$$

$$y_i^5 + x_j^2 \geq t_{ij}^1 + y_i^4 - x_j^1 - y_j^3 \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^2 \quad (9b)$$

Lastly, chromatographic stages are modeled considering both fixed and variable operation times leading to the highly non-linear constraint (10).

$$Y_i^5 \geq \frac{T_{ij}^0 + T_{ij}^1 \frac{Y_i^4}{X_j^1 Y_j^3}}{X_j^2} \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^3 \quad (10a)$$

$$y_i^5 + x_j^2 \geq \ln\left[\exp\left(t_{ij}^0\right) + \exp\left(t_{ij}^1 + y_i^4 - x_j^1 - y_j^3\right)\right] \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^3 \quad (10b)$$

Production targets for all products, d_i, must be satisfied within the time horizon δ.

$$\sum_{i\in\mathcal{I}} \frac{d_i Y_i^5}{Y_i^4} \leq \delta + v\delta \quad (11a)$$

$$\sum_{i\in\mathcal{I}} \frac{d_i}{\delta} \exp\left(y_i^5 - y_i^4\right) \leq 1 + v \quad (11b)$$

Finally, variables for duplication in-phase X_j^1 are restricted to integer values using constraints (12) and (13), where z_{jm}^3 are binary variables and $\mathcal{M}$ a set of available units to operate in parallel in-phase. The same is valid for variables for duplication out-of-phase, X_j^2.

$$x_j^1 = \sum_{m\in\mathcal{M}} z_{jm}^3 \ln(m) \quad \forall j \in \mathcal{E} \quad (12)$$

$$\sum_{m\in\mathcal{M}} z_{jm}^3 = 1 \quad \forall j \in \mathcal{E} \quad (13)$$

Appropriate upper and lower bounds are also considered for all of the variables.

3.1.2. The design problem with selection of routes and equipment sizing (P2)

More recent models (like Iribarren et al. (2004)) take into account the joint selection of the production processes including selection of hosts and equipment. Their formulation uses classical *Big-M* constraints. Since we know that these constraints are problematic (Bosch and Trick, 2005) in this work we propose a different way to formulate the integer part of the problem replacing the *Big-M* by *clique* constraints. With this formulation all constraints are ignored except the upper bound on the variables (Dietrich et al., 1993). Model (P2) includes the sizing of the equipment, the duplication of units in parallel working in-phase and out-of-phase and accounts for the selection of the global process selecting routes which are defined as the series of unit operations used to purify

a protein given a certain host that synthesizes it. In this way once the pair product-host, (i, h), is selected the set of stages conforming the process is fixed.

The objective function becomes:

$$\begin{aligned}\min \text{ cost} = &\sum_{j\in\mathcal{E}^1} z_j^2 c_j^1 \exp\left(x_j^1 + x_j^2 + y_j^1\gamma_j^1\right) \\ &+ \sum_{j\in\mathcal{E}^2\cup\mathcal{E}^3} z_j^2\left[c_j^1 \exp\left(x_j^2 + y_j^1\gamma_j^1\right) + c_j^2 \exp\left(x_j^2 + y_j^2\gamma_j^2\right)\right. \\ &\left. + c_j^3 \exp\left(x_j^1 + x_j^2 + y_j^3\gamma_j^3\right)\right] + \nu\rho\delta \end{aligned} \tag{14}$$

Since some stages can be unused and just one route per protein can be selected we introduced two binary variables: z_{ih}^1 and z_j^2. z_{ih}^1 is equal to 1 when for product i synthesis host h is selected and 0 otherwise and z_j^2 is 1 when stage j is used to process at least one of the products and 0 otherwise. Constraint (15) enforces to chose just one host h to produce the protein i and constraint (16) permits stage j to be used just in case at least one product needs it to be processed.

$$\sum_{(i,h)\in\mathcal{U}} z_{ih}^1 = 1 \tag{15}$$

$$z_j^2 \geq z_{ih}^1 \quad \forall(i,h,j)\in\mathcal{R}|(i,h)\in\mathcal{U} \tag{16}$$

For chromatographic stages constraints take the form of Eqs. (17)–(20) that are trivially satisfied if host h is not selected to produced protein i ($z_{ih}^1 = 0$). When the host h is selected for protein i ($z_{ih}^1 = 1$) and the stage j has to be performed to process product i then $z_j^2 = 1$ and constraints are the same as in previous formulation (Section 3.1.1).

$$y_j^1 z_j^2 \geq s_{ihj}^1 z_{ih}^1 + y_{ih}^4 z_{ih}^1 \qquad \forall(i,h,j)\in\mathcal{R}, j\in\mathcal{E}^3 \tag{17}$$

$$y_j^2 z_j^2 \geq s_{ihj}^2 z_{ih}^1 + y_{ih}^4 z_{ih}^1 \qquad \forall(i,h,j)\in\mathcal{R}, j\in\mathcal{E}^3 \tag{18}$$

$$y_j^3 z_j^2 + x_j^1 z_j^2 \geq s_{ihj}^3 z_{ih}^1 + y_{ih}^4 z_{ih}^1 \qquad \forall(i,h,j)\in\mathcal{R}, j\in\mathcal{E}^3 \tag{19}$$

$$y_{ih}^5 z_{ih}^1 + x_j^2 z_j^2 \geq \ln\left[\exp\left(t_{ihj}^0\right) + \exp\left(t_{ihj}^1 + y_{ih}^4 - x_j^1 - y_j^3\right)\right] z_{ih}^1 \quad \forall(i,h,j)\in\mathcal{R}, j\in\mathcal{E}^3 \tag{20}$$

If stage j is not necessary for the process ($z_j^2 = 0$); then equipment sizes are set to 0 with constraints as (21) and no unit is considered to conform that stage (constraint (22)):

$$y_j^{1,LO} z_j^2 \leq y_j^1 \leq y_j^{1,UP} z_j^2 \quad \forall j\in\mathcal{E} \tag{21}$$

$$\sum_{m\in\mathcal{M}} z_{jm}^3 = z_j^2 \quad \forall j\in\mathcal{E} \tag{22}$$

Finally, in the planning horizon constraint (23) only the terms associated to the selected host per protein are considered.

$$\sum_{(i,h)\in\mathcal{U}} \frac{d_{ih}}{\delta} z_{ih}^1 \exp\left(y_{ih}^5 - y_{ih}^4\right) \leq 1 + \nu \tag{23}$$

3.2. Mixed-integer linear formulations

To obtain more accurate solutions and, specially in larger instances, in a reasonable running time we present a MILP reformulation, which can be solved using any commercial MILP solver. These models are basically equal to their MINLP counterpart but replacing the non-linear objective and time constraints with sets of linear functions which give arbitrarily good lower or upper approximations of their respective original functions. The actual optimal solution is in between both approximations and the precision level is given by the number of cutting points selected to generate the set of linear functions to replace each non-linear function and the actual selection of the approximation points used for example equispaced or non-equispaced. In this way the accuracy of the solution can be as high as desired at the cost of longer computing time.

3.2.1. Inner and outer approximations

Given a convex function of one variable $g(x)\leq 0$ and a set of points $\{x_k\}_{k=1,\ldots,n}$ in the domain of g then, is easy to see that:

$$\left\{x|g(\hat{x}_k) + \nabla g(\hat{x}_k)(x-\hat{x}_k)\leq 0 \quad k=1,\ldots,n\right\} \supseteq \left\{x|g(x)\leq 0\right\} \tag{24}$$

and

$$\left\{x|g(\hat{x}_k) + \frac{g\left(\hat{x}_{k+1}\right) - g\left(\hat{x}_k\right)}{\hat{x}_{k+1}-\hat{x}_k}\left(x-\hat{x}_k\right)\leq 0 \quad k=1, ldots, n\right\} \subseteq \{x|g(x)\leq 0\}, \tag{25}$$

which allows for straightforward lower and upper approximations of g. Using this fact, it is easy to find inner and outer approximations of the problems (P1) and (P2).

In fact, for each non-linear constraints of the form $g_j(x)\leq 0$, and considering an arbitrary set of cutting points in the domain $\{x_k\}_{k=1,\ldots,n}$ the consideration of the set of constraints

$$g_j(\hat{x}_k) + \nabla g_j(\hat{x}_k)\left(x-\hat{x}_k\right)\leq 0 \quad k=1,\ldots,n \tag{26}$$

which leads to a larger feasible set, as can be seen in Fig. 3a. On the other hand, we consider the set of constraints

$$g_j(\hat{x}_k) + \frac{g_j\left(\hat{x}_{k+1}\right) - g_j\left(\hat{x}_k\right)}{\hat{x}_{k+1}-\hat{x}_k}\left(x-\hat{x}_k\right)\leq 0 \quad k=1,\ldots,n-1 \tag{27}$$

which leads to a smaller feasible set, as can be seen in Fig. 3b.

In the same way, the minimization of the cost objective function, $f(x)$, can be replaced by

$$\begin{aligned}&\min \quad \nu \\ &s.t. \qquad \nu \geq f(\hat{x}_k) + \nabla f(\hat{x}_k)(x-\hat{x}_k) \quad k=1,\ldots,n\end{aligned} \tag{28}$$

which leads, together with an outer approximation of constraints, to a lower bound of the true cost. The objective function can also be replaced by

$$\begin{aligned}&\min \quad \nu \\ &s.t. \qquad \nu \geq f(\hat{x}_k) + \frac{f\left(\hat{x}_{k+1}\right) - f\left(\hat{x}_k\right)}{\hat{x}_{k+1}-\hat{x}_k}(x-\hat{x}_k) \\ &\qquad\quad k=1,\ldots,n\end{aligned} \tag{29}$$

which leads, together with an inner approximation of constraints, to an upper bound of the true cost.

In what follows, $\nabla f(\hat{x}_k)$ or $\frac{f(\hat{x}_{k+1})-f(\hat{x}_k)}{\hat{x}_{k+1}-\hat{x}_k} = \alpha_k$ in the lower or upper approximation, respectively, $\hat{x}_k = b_k$ and $f(\hat{x}_k) = \beta_k$.

3.2.2. Reformulation for the equipment-sizing problem (P3)

The assumptions for this model are the same as those for the MINLP proposed in Section 3.1.2.

Objective function. Cost functions of Eq. (2) are individually linearized using the approximations given in Section 3.2.1 which leads to Eq. (30):

$$\min \text{ cost} = \sum_{j\in\mathcal{E}^1} v_j^1 + \sum_{j\in\mathcal{E}^2\cup\mathcal{E}^3}\left[v_j^1 + v_j^2 + v_j^3\right] \tag{30}$$

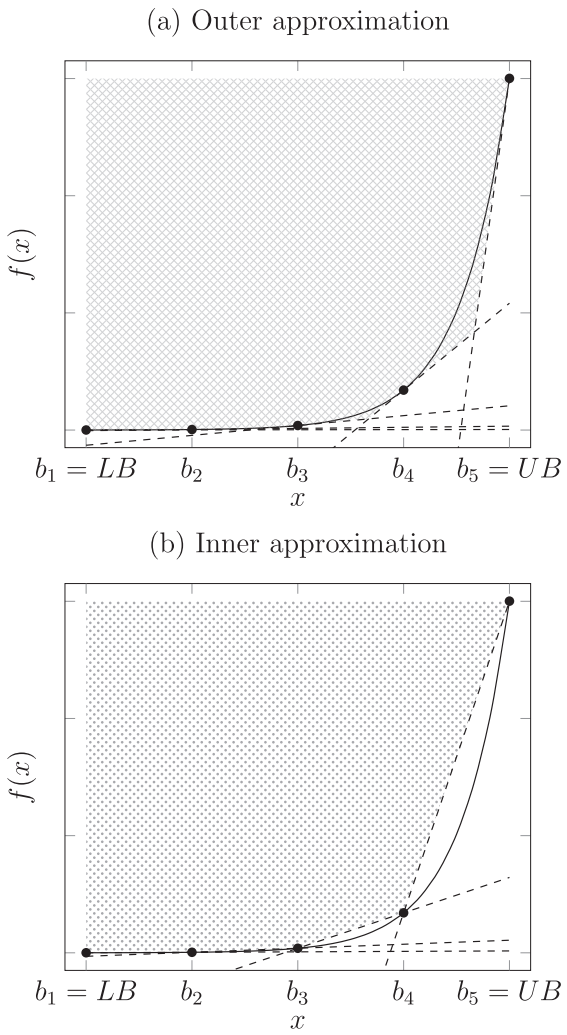


Fig. 3. Feasible region (patterned area) of (a) outer and (b) inner approximations (dashed lines) of an exponential function (solid line). Points b_i are the cutting points and LB and UB are the lower and upper bounds of x.

Constraints. Batch and semi-continuous stages and binary variables for duplication of units constraints in this MILP model are the same as those in the MINLP model shown in Section 3.1.1.

Chromatographic stages. Size constraints for feed and product tanks and column size constraint are the same as those in the MINLP model shown in Section 3.1.1.

Time constraint (31) is obtained from the linearization of Eq. (10):

$$y_i^5 + x_j^2 \geq \alpha_k^{6ij}\left(y_i^4 - x_j^1 - y_j^3 - b_k^{6ij}\right) + \beta_k^{6ij} \quad \forall i \in \mathcal{I}, j \in \mathcal{E}^3, k \in \mathcal{K}^6 \tag{31}$$

Planning horizon. From linearization of Eq. (11) constraint (32) is obtained:

$$\sum_{i \in \mathcal{I}} v_i^7 \leq 1 \tag{32}$$

Auxiliary variables. Cost functions in the objective function are linearized as shown in Eq. (33) and planning horizon constraint is linearized as shown in Eq. (34):

$$v_j^1 \geq \alpha_k^{1j}\left(x_j^1 + x_j^2 + \gamma_j^1 y_j^1 - b_k^{1j}\right) + \beta_k^{1j} \quad \forall j \in \mathcal{E}^1, k \in \mathcal{K}^1 \tag{33}$$

$$v_i^7 \geq \frac{d_i}{\delta}\alpha_k^{7i}\left(y_i^5 - y_i^4 - b_k^{7i}\right) + \frac{d_i}{\delta}\beta_k^{7i} \quad \forall i \in \mathcal{I}, k \in \mathcal{K}^7 \tag{34}$$

3.2.3. Reformulation for the design problem considering route selection and equipment sizing (P4)

Similar to the model presented in Section 3.2.2 this model was built based on its MINLP counterpart and most of constraints remain the same.

The objective function is the same as that from model (P3) and for stages design only differences are encountered for time constraints of chromatographic stages. In this way, applying the inner or outer approximations to Eq. (20) constraint (35) is obtained:

$$y_{ih}^5 + x_j^2 \geq \alpha_k^{6ihj}\left(y_{ih}^4 - x_j^1 - y_j^3\right) + \left(\beta_k^{6ihj} - \alpha_k^{6ihj} b_k^{6ihj}\right) z_{ih}^1 \quad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3, k \in \mathcal{K}^6 \tag{35}$$

This set of equations together with constraints of the type of (36) for variables $y_j^1, Y_j^2, y_{ih}^4, y_{ih}^5, X_j^1$ and X_j^2 will model the same situation as in (P2).

$$y_j^{3,LO} z_j^2 \leq y_j^3 \leq y_j^{3,UP} z_j^2 \quad \forall j \in \mathcal{E}^3 \tag{36}$$

If $z_j^2 = 0$ constraint (35) is trivially satisfied. On the other hand if $z_{ih}^1 = 0$ constraint (35) becomes:

$$x_j^2 \geq \alpha_k^{6ihj}\left(-x_j^1 - y_j^3\right) \quad \forall (i,h,j) \in \mathcal{R}, j \in \mathcal{E}^3, k \in \mathcal{K}^6 \tag{37}$$

Since α_k^{6ihj} is a positive parameter constraint (37) will be always satisfied only if $|x_j^1| > |y_j^3|$ or if both variables are bigger than or equal to 0. To assure this data preprocessing is necessary. As X_j^1 is always bigger than 0 we normalized variables Y_j^3 by their lower bound. More details in Section 4.3.3.

For the case of planning horizon constraint little difference is found between (32) and (38). The last one takes into account host selection:

$$\sum_{(i,h) \in \mathcal{U}} v_{ih}^7 \leq 1 \tag{38}$$

Finally, constraints for auxiliary variables v_j^1, v_j^2, v_j^3 and v_{ih}^7 are different from those for problem (P3) to account for route selection:

$$v_j^1 \geq \alpha_k^{1j}\left(x_j^1 + x_j^2 + \gamma_j^1 y_j^1\right) + \left(\beta_k^{1j} - \alpha_k^{1j} b_k^{1j}\right) z_j^2 \quad \forall j \in \mathcal{E}^1, k \in \mathcal{K}^1 \tag{39}$$

$$v_j^1 \leq z_j^2 v_j^{1,UP} \quad \forall j \in \mathcal{E}^1 \tag{40}$$

$$v_{ih}^7 \geq \frac{d_i}{\delta}\alpha_k^{7ih}\left(y_{ih}^5 - y_{ih}^4\right) + \frac{d_i}{\delta}\left(\beta_k^{7ih} - \alpha_k^{7ih} b_k^{7ih}\right) z_{ih}^1 \quad \forall (i,h) \in \mathcal{U}, k \in \mathcal{K}^7 \tag{41}$$

$$v_{ih}^7 \leq z_{ih}^1 v_{ih}^{7,UP} \quad \forall (i,h) \in \mathcal{U} \tag{42}$$

If $z_j^2 = 0$ constraints (39)–(42) are trivially satisfied and if $z_{ih}^1 = 0$ then constraints (41) and (42) are trivially satisfied. Finally, if $z_{ih}^1 = 1$ and $z_j^2 = 1$ then constraints (39)–(42) are the same as those in the MILP formulation without route selection.

4. Methods

4.1. Solvers and modelling language

For MINLP problems the open source BONMIN 1.5 and SCIP 3.0.1 solvers were studied. In our computational tests SCIP uses SoPlex

1.7.1 as the LP solver and BONMIN (with its default algorithm, B-Hyb) uses Cbc 2.7.1 as the MIP solver and Ipopt 3.10.0 with MUMPS as linear solver. For the case of BONMIN we tested 3 over 5 available algorithms: B-Hyb the default algorithm, B-Ecp a specific parameter setting of B-Hyb that can be faster in some cases (Bonami and Lee, 2013) and B-OA using CPLEX as the MILP solver that according to Mittelmann (2013) can be faster for convex instances. In preliminary studies solvers as KNITRO and COUENNE were also tested to solve our MINLP formulations, but their performance in our simplest instances were poorer than that for the selected solvers.

For MILP problems the commercial CPLEX solver in its version 12.4.0.0 was used as it is one of the top performer from the literature (Mittelmann, 2013).

All models were coded using the AMPL modelling language.

4.2. Execution environment

Each instance was executed using a single thread on a Intel(R) Xeon(R) CPU E5620@2.40GHz with a running time limit of 48 hours, an optimality relative gap of 0.1% for models (P3) and (P4) and 2% for (P1) and (P2), and a maximum memory usage of 6Gb of RAM.

The difference in the prescribed optimality gap for MILP and MINLP solvers is given by the fact that while MINLP problems are solved to find the actual minimum cost function within a defined optimality gap, and therefore an *a priori* optimality gap, MILP models find true upper and lower bounds for the actual cost function leading to an *a posteriori* optimality gap that is computed afterwards. As will be shown in Section 5.2, this difference ensure that our results are comparable.

4.3. Methodology

4.3.1. Instances

To compare different approaches two set of instances, with randomly generated data between given *reasonable* upper and lower bounds, were built: "sizing instances", to compare simpler models (P1) and (P3), and "routing instances" to compare more complex models (P2) and (P4). We considered a variety of different number of proteins to be produced (4–6), number of stages to conform the process (11–65), number of routes to synthesize the product (20–65) and different cost coefficients values (1–110% of nominal values).

4.3.2. Benchmarking

In order to compare the model-solver pairs studied in this work we introduce a new tool for process engineers that was introduced in the optimization field by Dolan and Moré (2002) to compare different optimization software: the performance profile.

As stated by Dolan and Moré (2002) the performance profile for a solver is the "cumulative distribution for a performance metric", for example computing time. In this way things like how many instances a solver is able to solve given some stop criteria like those shown in Section 4.2, or how fast it solves different instances of the same type of problem can be seen graphically.

As an example of how to read these plots, in Fig. 4a it can be seen that when using 17 cutting points 40% of the instances were solved to an *a posteriori* optimality gap up to 2% while when using 33 cutting points leads to an optimality gap under 0.5% for the same amount of instances.

4.3.3. Data pre-processing

It is known that zero-one problems of large-scale are hard combinatorial optimization problems (Crowder et al., 1983; Koch, 2004; Applegate et al., 2007) reason why in order to obtain reliable solutions preprocessing data is necessary. The use of tight bounds and

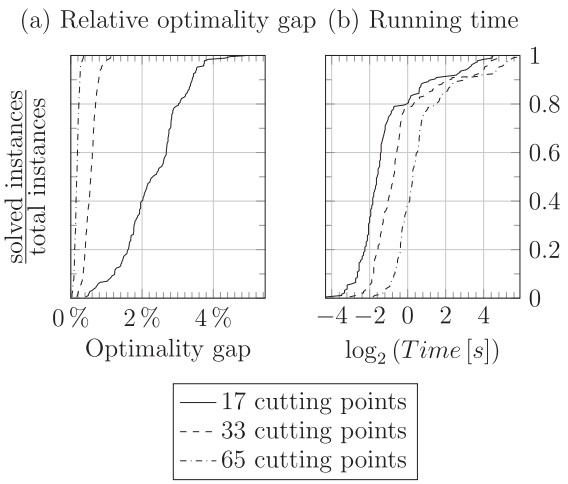


Fig. 4. Comparison of performance profiles of (a) relative optimality gap obtained *a posteriori* and (b) the logarithm of the running time of "sizing instances" solved with linear model (P3) using 17, 33 and 65 cutting points for lower and upper approximations with an optimality relative gap of 0.1%.

the normalization of the variables are necessary to decrease numerical errors.

Although not all of our instances are big enough to need data preprocessing all were subjected to the same treatment:

- All variable bounds and parameters associated to variables Y_j and Y_i were normalized by their respective lower bounds.
- Size and time factors were normalized and dimensionless considering the respective associated units. For example, as size factor for tanks have units of batch size divided by a volume this parameters are dimensionless by multiplying by the lower bound of the final batch size and dividing by the respective tank lower bound.
- Lower bounds for the cycle time were tightened using time constraints and upper bounds for final batch product were tightened using size constraints.

5. Results and discussion

In this section we show the robustness of our proposed MILP transformations and its superiority over classic MINLP formulations with *Big-M* constraints using performance profiles, a methodology borrowed from the optimization literature. Our approach is not only able to find correct solutions in realistic situations unlike MINLP formulations but also in a small fraction of the time required by those approaches. Major implications of these features are the exactness of the solutions that make this information reliable for decision-making; and as time reduction is significant numerous alternatives can be tested with the same formulation or with complexified models that may address the combination of different types of decisions.

This presentation is organized as follows: first, we describe the instances generated for comparison then discuss the selection of the cutting points for the proposed approach and finally, we compare MINLP and MILP formulations in terms of their performance solving the sets of instances using time as the metric.

5.1. Size of instances

To compare the most basic and easy to solve problems (P1) and (P3) a set of 186 instances ("sizing instances") were generated varying the number of proteins to be produced (2–6), the number of stages that conform each process (11–35) and the cost coefficient values (1–110% of nominal values). Sizes of these instances in terms

Table 2
Sizes of sample instances solved using non-linear model (P1).

	Variables		Constraints	
	Discrete	Continuous	Linear	Non-linear
Small	264	57	139	9
Medium1	840	157	407	5

Table 3
Sizes of sample instances solved using linear model (P3) with 33 cutting points for linear inner approximation.

	Variables		Constraints	
	Discrete	Continuous	Linear	Non-linear
Small	264	87	1357	–
Medium1	840	237	3096	–

of number of variables and constraints are shown in Tables 2 and 3, where "Small" corresponds to an example of one of the smaller instances solved with different models and "Medium1", to an example of the bigger instances solved for these two models. As it can be seen in both tables new auxiliary variables and the sets of linear functions generated to replace non-linear restrictions makes the problem from 7 to 12 times bigger in terms of linear constraints when 33 cutting points are used for linearization with an increase in about 50% of continuous variables. However, as we will see later, this increase in variables and constraints leads to smaller execution times and more accurate results.

To test models (P2) and (P4), as they were posed to solve more complex scenarios, a set of 249 new and bigger instances ("routing instances") were generated varying the number of proteins to be produced (4–6), the number of stages conforming the global process (18–65) and the number of routes available to produce the proteins (20–40). Sizes of these instances in terms of number of variables and constraints are shown in Tables 4 and 5, where "Medium2" corresponds to an example of one of the smaller "routing instances" solved with models (P2) and (P4) and "Large", to an example of the bigger instances solved in this work. As it can be seen in both tables, in comparison with Tables 2 and 3, the number of discrete variables increases by 5% with the addition of selection variables, z, and the number of linear constraints increases by 15% for (P4) and is around double for (P2). As we will see later, this addition permits the resolution of more complex scenarios, while at the same time not affecting execution time or optimality gap in comparison with the more basic formulation.

Table 4
Sizes of sample instances solved using non-linear model (P2).

	Variables		Constraints	
	Discrete	Continuous	Linear	Non-linear
Small	279	57	251	9
Medium1	881	157	719	5
Medium2	488	152	1262	22
Large	1794	613	15766	462

Table 5
Sizes of sample instances solved using linear model (P4) with 33 cutting points for linear inner approximation.

	Variables		Constraints	
	Discrete	Continuous	Linear	Non-linear
Small	279	87	1539	–
Medium1	881	237	3602	–
Medium2	488	229	4829	–
Large	1794	926	46489	–

5.2. Selection of cutting points

Contrary to MINLP problems (P1) and (P2) that are solved to an *a priori* optimality gap, models (P3) and (P4) give true upper and lower bounds for the actual cost function of each instance and therefore an optimality gap that is computed *a posteriori*. Fig. 4a shows the performance profiles of the gaps obtained *a posteriori* for the "sizing instances" solved with 17, 33 and 65 cutting points that generate 16, 32 and 64 linear functions for the inner approximations, respectively. Here we can see that our linear model is able to solve all instances with a maximum gap of 5% in less than 16 seconds when using 17 cutting points, and a gap of less than 0.5% in less than 64 seconds when using 65 cutting points. The running time profiles can be seen in Fig. 4b. If 65 cutting points had been selected, the optimality gap for non-linear solvers would have been around 0.5% as that is the worst gap obtained *a posteriori* with CPLEX (Fig. 4a). Given this, in our final experiments we use a set of 33 cutting points, since this option gives the largest improvement in gap versus the increase in execution time, and a slightly bigger optimality gap criteria of 2% was selected for non-linear solvers.

Once the number of points is selected, the specific values of these points must be chosen. The most obvious choice is equispaced points which, for the (relevant) exponential function e^x generates small errors for low values of x and large errors for high values. Another alternative is to use the expression (43), where N is the total number of cutting points including $-\infty$ as x_1 and $\bar{x}$ as the upper bound of x. This is a good approximation in order to minimize the *maximum value* of the error (see Fig. 5).

$$x_k = 2\ln\left(\frac{k-1}{N-1}\right) + \bar{x} \qquad \forall k \in 2...N \tag{43}$$

We can see in Fig. 5 that the choice of equispaced points leads to better approximations for low values of x, but much worse for high values. For our numerical experiments, equispaced points work better: while execution time remains the same for both approaches, *a posteriori* gaps were slightly worse for non-equispaced points (Fig. 6).

This leaves open important questions about the optimal point selection to improve the precision of upper and lower approximations. Our preliminary simulations seem to indicate that giving substantial attention to smaller values of x could significantly improve the results, but this is left for further research.

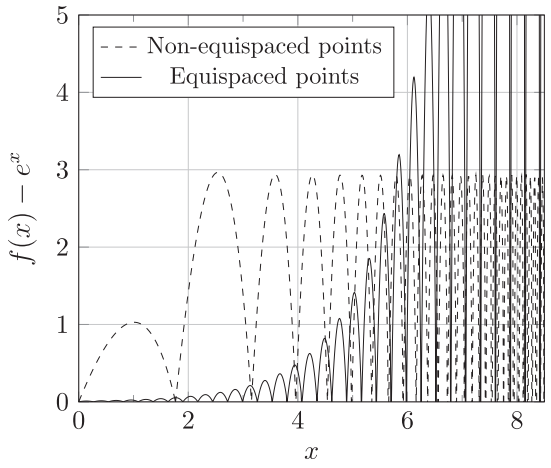


Fig. 5. Comparison of absolute errors using 2 different sets of 33 cutting points where $f(x)$ are the linear functions used to approximate the exponential function between 2 cutting points.

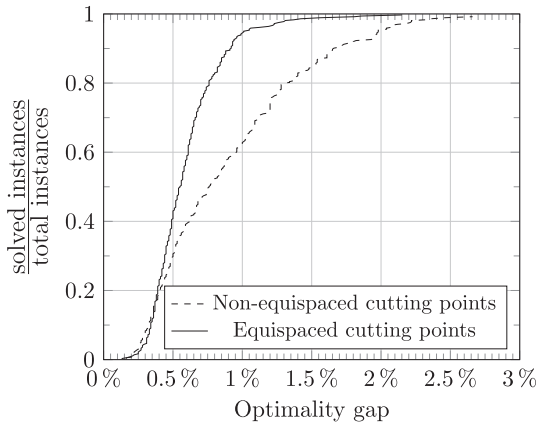


Fig. 6. Comparison of performance profiles of *a posteriori* gaps obtained using 33 cutting points to solve model (P4) where $f(x)$ are the linear functions used to approximate the exponential function between 2 cutting points. Time limit was set in 12 h.

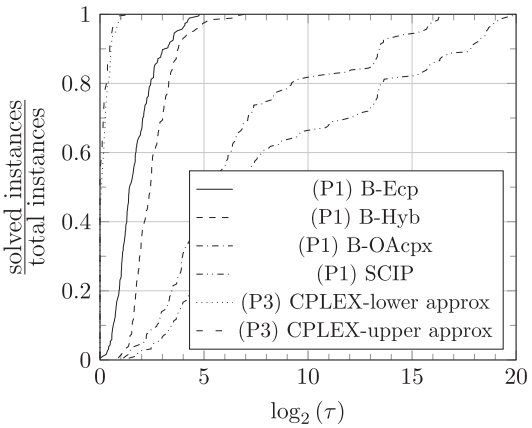


Fig. 8. Comparison of performance profiles of the logarithm of the ratio of the computing time of the pair model-solver versus the best time of the pairs model-solvers for "sizing instances" solved with models (P1) and (P3) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

5.3. Equipment sizing: comparison of problems (P1) and (P3)

We tested three different combinations of solvers-models: the linear model (P3) was solved using CPLEX as solver while the non-linear model, (P1), was solved using SCIP and 3 of the 5 algorithms that are available for using BONMIN which were chosen based on BONMIN users' manual and Mittelmann's benchmarking (Mittelmann, 2013) information. All of the instances were solved using the stopping criteria and conditions mentioned in Section 4.2.

Fig. 7 shows the performance profile of running time using BONMIN-Hyb, BONMIN-Ecp, BONMIN-OAcpx, SCIP and our CPLEX-based approach. From this, we can see that problem (P1) was solved faster using any BONMIN algorithm than using SCIP solver. Moreover, SCIP only worked well in about the 75% of the instances, while BONMIN is able to solve the 85% of the instances using the OA algorithm and the 100% of the studied instances using either the Ecp or the Hyb algorithm. On the other hand, model (P3) solved using CPLEX is able to solve all of the instances studied, as well as BONMIN, but taking much less time than the problem (P1). As B-Ecp and B-Hyb seems to be equally good for those instances that take longer to be solved we decided to use as the performance metric the ratio of the computing time of the model-solver versus the best time of all of the model-solvers, denoted by τ. Those performance profiles are plotted in Fig. 8 where we can see even more clear than the CPLEX-based approach is always better than all of the other options and that Ecp algorithm is always better than Hyb for this asked optimality gap.

5.4. Routes selection: comparison of problems (P2) and (P4)

As a test of correctness, models (P2) and (P4) were solved using the generated "sizing instances" where just one route was available to produce each product. Contrasting these results to those obtained by (P1) and (P3) it can be seen in Fig. 9 that both formulations, for equipment sizing and considering routes, are consistent solving the same amount of instances in virtually the same amount of time. Performance profiles of the relative difference between the value of the objective function obtained with simpler – (P1) and (P2) – and more complex formulations – (P3) and (P4) – are presented in Fig. 10 where it can be seen that the differences between MILP formulations, as well as for MINLP formulations solved using BONMIN, are at most the optimality gap asked for each solver. The case of SCIP is different because in almost the 10% of the instances the solver gives results with a difference in the objective function between models (P1) and (P2) greater than the optimality gap which shows that this solver is not reliable to solve this type of problems.

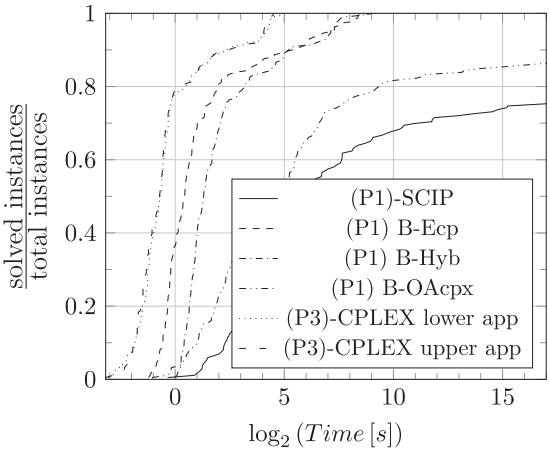


Fig. 7. Comparison of performance profiles of the logarithm of running time of "sizing instances" solved using models (P1) and (P3) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

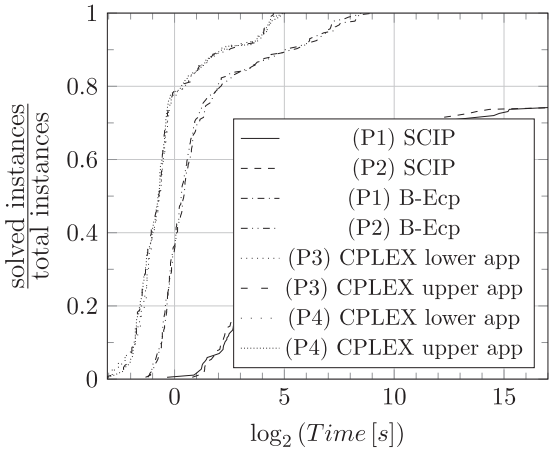


Fig. 9. Comparison of performance profiles of the logarithm of running time of "sizing instances" solved using models (P1), (P2), (P3) and (P4) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

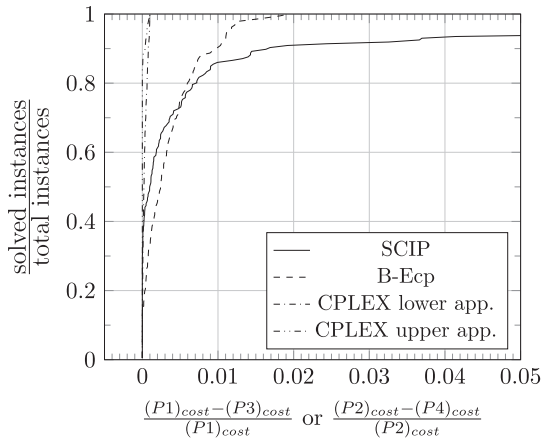


Fig. 10. Comparison of performance profiles of relative difference between simple and more complex formulation for "sizing instances". Models (P1) and (P2) were solved to an optimality gap of 2% and (P3) and (P4), to an optimality gap of 0.1%.

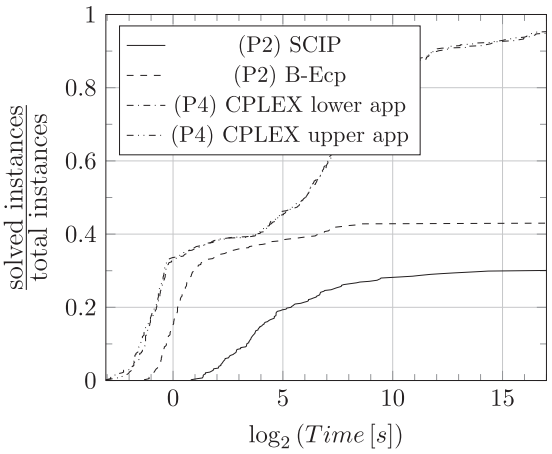


Fig. 11. Comparison of performance profiles of the logarithm of running time of "routing" and "sizing instances" solved using models (P2) and (P4) with an optimality relative gap of 0.1% for the linear solver and 2% for non-linear solvers.

As a final step in this work we compare in Fig. 11 the performance profiles of total running time obtained after solving all instances generated for this work (435 in total). Here we can see that for "routing instances" (P4) is much more robust than (P2), that was not able to solve any of those cases. In average (geometric average), the instances take about 40 s to be solved using model (P4), which is about the 3% of the time required by (P2)-BONMIN and less than the 1% of the time required by (P2)-SCIP. Nevertheless, for some punctual instances that represent the 4.6% of the instances studied, the time and/or memory usage were not enough to get the desired optimality gap. From this it can be stated that for the solution of more realistic instances or even to solve real problems considering continuous equipment sizes, the formulation proposed in this work is much more reliable and faster than the usual and widely studied standard MINLP formulation.

6. Conclusions

In this work we present a scalable approach to solve, within reasonable running times and quality assurance requirements, the problem of designing a biotechnological multi-product batch plant that support continuous equipment sizes and discrete host and/or process selection, up to sizes of real instances and that can be applicable to any kind of multi-product batch plant.

The proposed method was proved to be more *numerically stable* than other alternative approaches for the same problem giving true optimal solutions, and in general, faster than other tested approaches. Our method takes advantage of two facts: the continuous relaxation of the feasible region is convex and bounded (which allow us to build, up front, inner or outer approximations of the feasible space, and thus report true upper/lower bounds for each instance); and the fact that mixed-integer linear solvers are much more stable numerically and scalable in size than MINLP algorithms. Also, this approach relies on "off the shelve" optimization and modelling software, which makes it more amiable to practitioners.

To assert our claims, we borrow algorithm comparison tools from the optimization community, which are an interesting form to test the quality of competing algorithms to tackle the same class of problems.

Although, in a real scenario, semi-continuous units such as centrifuges and microfilters, among others, are available only in discrete sizes, unlike tanks that can be built according to customer needs, we feel that the proposed approach is robust enough to consider such issues, however, this was left as a next step in our research. Additionally, to increase the precision of our results for real cases, it is possible to explore a two step approach where after using the proposed method to obtain upper and lower approximations for the objective function we can refine different upper and lower variable bounds making them tighter and perform a re-optimization.

Finally, if true speed is the goal; we know that low-level implementations of dynamic inner/outer approximation can provide further time reductions, however, we feel that this is beyond the scope of this work.

Acknowledgements

This work was supported by a CONICYT scholarship for doctoral studies, FONDECYT Grant 1110024, Núcleo Milenio Información y Coordinación en Redes P10-024-F, and CONICYT for funding of Basal Centre, CeBiB, FB0001.

References

Applegate DL, Cook W, Dash S, Espinoza DG. Exact solutions to linear programming problems. Oper Res Lett 2007];35:693–9.
Barbosa-Póvoa AP. A critical review on the design and retrofit of batch plants. Comput Chem Eng 2007];31:833–55.
Bonami P, Lee J. BONMIN users's manual; 2013].
Bosch R, Trick M. Integer programming. In: Burke EK, Kendall G, editors. Search methodologies: introductory tutorials in optimization and decision support techniques. Springer; 2005]. p. 67–92.
Corsano G, Aguirre PA, Montagna JM. Multiperiod design and planning of multiproduct batch plants with mixedproduct campaigns. AIChE J 2009];55:2356–69.
Crowder H, Johnson EL, Padberg M. Solving large-scale zero-one linear programming problems. Oper Res 1983];31:803–34.
Czyzyk J, Mesnier MP, More JJ. The NEOS server. IEEE Comput Sci Eng 1998];5:68–75.
Dietrich BL, Escudero LF, Chance F. Efficient reformulation for 0-1 programs-methods and computational results. Discret Appl Math 1993];42:147–75.
Dolan ED, Moré JJ. Benchmarking optimization software with performance profiles. Math Program 2002];91:201–13.
Dolan ED. NEOS Server 4.0 administrative guide, Technical Report, Technical Memorandum ANL/MCS-TM-250. Mathematics and Computer Science Division, Argonne National Laboratory; 2001].
Durand GA, Mele FD, Bandoni JA. Determination of storage tanks location for optimal short-term scheduling in multipurpose/multiproduct batch-continuous plants under uncertainties. Ann Oper Res 2012];199:225–47.
Durand GA, Moreno MS, Mele FD, Montagna JM, Bandoni A. Comparing the performances of two techniques for the optimization under parametric uncertainty of the simultaneous design and planning of a multiproduct batch plant. Iberoam J Ind Eng 2014];5:42–54.
Floudas CA. Nonlinear and mixed-integer optimization: fundamentals and applications. Oxford University Press; 1995].
Fumero Y, Corsano G, Montagna JM. Detailed design of multiproduct batch plants considering production scheduling. Ind Eng Chem Res 2011];50:6146–60.

Fumero Y, Montagna JM, Corsano G. Simultaneous design and scheduling of a semi-continuous/batch plant for ethanol and derivatives production. Comput Chem Eng 2012a];36:342–57.
Fumero Y, Corsano G, Montagna JM. Planning and scheduling of multistage multiproduct batch plants operating under production campaigns. Ann Oper Res 2012b];199:249–68.
Geißler B, Martin A, Morsi A, Schewe L. Using piecewise linear functions for solving MINLPs. In: Mixed Integer Nonlinear Programming. Springer; 2012]. p. 287–314.
Goldberg D. What every computer scientist should know about floating-point arithmetic. ACM Comput Surv (CSUR) 1991];23:5–48.
Gropp W, Moré J. Optimization environments and the NEOS server. Approx Theory Opt 1997]:167–82.
Grossmann IE, Guillén-Gosálbez G. Scope for the application of mathematical programming techniques in the synthesis and planning of sustainable processes. Comput Chem Eng 2010];34:1365–76.
Grossmann IE, Caballero JA, Yeomans H. Advances in mathematical programming for the synthesis of process systems. Latin Am Appl Res 2000];30:263–84.
Gupta S, Karimi IA. An improved MILP formulation for scheduling multiproduct, multistage batch plants. Ind Eng Chem Res 2003];42:2365–80.
Iribarren OA, Montagna JM, Vecchietti AR, Andrews B, Asenjo JA, Pinto JM. Optimal process synthesis for the production of multiple recombinant proteins. Biotechnol Prog 2004];20:1032–43.
Koch T. The final NETLIB-LP results. Oper Res Lett 2004];32:138–42.
Kocis GR, Grossmann IE. Global Optimization of nonconvex mixed-integer nonlinear programming (MINLP) problems in process synthesis. Ind Eng Chem Res 1988];27:1407–21.
Li X, Chen Y, Barton PI. Nonconvex generalized Benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. Ind Eng Chem Res 2012];51:7287–99.
Margot F. Testing cut generators for mixed-integer linear programming. Math Program Comput 2009];1:69–95.
Mittelmann H. Decision Tree for Optimization Software; 2013].
Montagna JM, Vecchietti AR, Iribarren OA, Pinto JM, Asenjo JA. Optimal design of protein production plants with time and size factor process models. Biotechnol Prog 2000];16:228–37.
Montagna JM, Iribarren OA, Vecchietti AR. Synthesis of biotechnological processes using generalized disjunctive programming. Ind Eng Chem Res 2004];43:4220–32.
Moreno MS, Montagna JM. New alternatives in the design and planning of multiproduct batch plants in a multiperiod scenario. Ind Eng Chem Res 2007];46:5645–58.
Moreno MS, Montagna JM. Multiperiod production planning and design of batch plants under uncertainty. Comput Chem Eng 2012];40:181–90.
Moreno MS, Montagna JM, Iribarren OA. Multiperiod optimization for the design and planning of multiproduct batch plants. Comput Chem Eng 2007];31:1159–73.
Moreno MS, Iribarren OA, Montagna JM. Optimal design of multiproduct batch plants considering duplication of units in series. Chem Eng Res Des 2009];87:1497–508.
Moreno MS, Iribarren OA, Montagna JM. Design of multiproduct batch plants with units in series including process performance models. Ind Eng Chem Res 2009];48:2634–45.
Moreno-Benito M, Espuña A, Puigjaner L. Flexible batch process and plant design using mixed-logic dynamic optimization: single-product plants. Ind Eng Chem Res 2014];53:17182–99.
Nikolopoulou A, Ierapetritou MG. Optimal design of sustainable chemical processes and supply chains: a review. Comput Chem Eng 2012];44:94–103.
Nowak I. Overview of global optimization methods. In: Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming; 2005]. p. 121–8.
Pinto JM, Montagna JM, Vecchietti AR, Iribarren OA, Asenjo JA. Process performance models in the optimization of multiproduct protein production plants. Biotechnol Bioeng 2001];74:451–65.
Ponsich A, Azzaro-Pantel C, Domenech S, Pibouleau L. Mixed-integer nonlinear programming optimization strategies for batch plant design problems. Ind Eng Chem Res 2007];46:854–63.
Ravemark DE, Rippin DWT. Optimal design of a multi-product batch plant. Comput Chem Eng 1998];22:177–83.
Rebennack S, Kallrath J, Pardalos PM. Optimal storage design for a multi-product plant: a non-convex MINLP formulation. Comput Chem Eng 2011];35:255–71.
Reklaitis GV. Progress and issues in computer-aided batch process design. Found Comput Aided Process Des 1990]:241–75.
Rippin D. Batch process systems engineering: A retrospective and prospective review. Comput Chem Eng 1993];17:S1–13.
Robinson JD, Loonkar YR. Minimizing capital investment for multi-product batch-plants. Process Technol 1972];17:861.
Verderame PM, Elia JA, Li J, Floudas CA. Planning and scheduling under uncertainty: a review across multiple sectors. Ind Eng Chem Res 2010];49:3993–4017.
Vielma JP. Mixed integer linear programming formulation techniques; 2013].
Voudouris VT, Grossmann IE. Mixed-integer linear programming reformulations for batch process design with discrete equipment sizes. Ind Eng Chem Res 1992];31:1315–25.
Voudouris VT, Grossmann IE. Optimal synthesis of multiproduct batch plants with cyclic scheduling and inventory considerations. Ind Eng Chem Res 1993];32:1962–80.