

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
DIVISIÓN DE ESTUDIOS DE POSGRADO



Expansión de Capacidad de Plantas Tipo Batch

Por

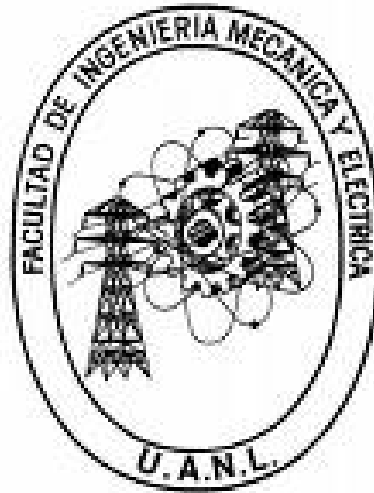
M.C. MARÍA GABRIELA GARCÍA AYALA

Como requisito parcial para obtener el Grado de

DOCTORA EN INGENIERÍA con especialidad en INGENIERÍA DE SISTEMAS

San Nicolás de los Garza, Nuevo León, Enero, 2010

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN  
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
DIVISIÓN DE ESTUDIOS DE POSGRADO



Expansión de Capacidad de Plantas Tipo Batch

Por

M.C. MARÍA GABRIELA GARCÍA AYALA

Como requisito parcial para obtener el Grado de

DOCTORA EN INGENIERÍA con especialidad en INGENIERÍA DE SISTEMAS

San Nicolás de los Garza, Nuevo León, Enero, 2010

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE INGENIERIA MECANICA Y ELÉCTRICA

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis « Expansión de Capacidad de Plantas Tipo Batch», realizada por la alumna María Gabriela García Ayala, con número de matrícula 1404152, sea aceptada para su defensa como opción al grado de Doctora en Ingeniería con orientación en Ingeniería de Sistemas.

El Comité de Tesis

Dr. Oscar Chacón Mondragón

Asesor

Dr. Roger Z. Ríos Mercado

Revisor

Dr. Igor Litvinchev

Revisor

Dra. Yasmín A. Ríos Solís

Revisor

Dr. Joaquín Acevedo Mascarúa

Revisor

Dr. Moises Hinojosa Rivera

Subdirector de Posgrado

San Nicolás de los Garza, Nuevo León, Enero, 2010

*Para Juan Carlos, por siempre brindarme unos brazos donde caer rendida, por ser capaz de robarme una sonrisa o devolverme la paz en un instante, por ser el amor de mi vida...*

# Índice General

## ***Capítulo 1. Introducción***

1.1 Motivación.....	1
1.2 Definición del Problema.....	3
1.3 Objetivos de Investigación.....	4
1.4 Resumen de Tesis.....	5

## ***Capítulo 2. Antecedentes***

2.1 Introducción a Procesos Batch.....	7
2.2 Estrategias Computacionales.....	12
2.2.1 Programación Multiperíodo Entero Mixta.....	17
2.2.2 Programación Disyunta.....	21
2.3 Aplicación Industrial.....	23
2.3.1 Breve introducción al proceso cervecero.....	23
2.3.2 Aplicación Industrial.....	24

## ***Capítulo 3. Modelo Multiperíodo de Diseño Retrofit para Plantas Batch***

3.1 Introducción.....	31
3.2 Modelo General Multiperíodo de Van den Heever (2000).....	32
3.2.1 Modelo MINLP Multiperíodo General.....	33
3.2.2 Modelo Multiperíodo Disjunto.....	37
3.2.3 Algoritmo de Aproximación Externa (OA) Disjunto.....	39
3.3 Aplicación a Problema de Interés.....	44
3.4 Casos de Estudio.....	51
3.5 Conclusiones.....	59

## ***Capítulo 4. Planeación de la Producción y Horizonte Móvil Incorporados al Modelo Retrofit***

4.1 Introducción.....	62
4.2. Horizonte Móvil.....	63

4.3 Planeación de la Producción.....	71
4.4 Algoritmo Corregido para Diseño Retrofit de Plantas Batch.....	74
4.5 Casos de Estudio.....	78
4.6 Conclusiones.....	83
<b>Capítulo 5. Conclusiones y Direcciones Para Trabajo Futuro</b>	
5.1 Recopilación.....	86
5.2 Resumen de Tesis.....	87
5.3 Conclusiones Generales.....	90
5.4 Contribuciones de investigación.....	93
5.5 Direcciones para Trabajo Futuro.....	94
5.5.1 Programación disyunta.....	94
5.5.2 Integración de plantación y diseño.....	95
5.5.3. Manejo de incertidumbre.....	95
5.5.4. Incluir gastos de operación.....	97
5.5.5. Optimización de Superestructura.....	97
<b>Bibliografía</b> .....	99
<b>Apéndices</b>	111
Apéndice A: Derivación para el envolvente convexo del problema (DMILP).....	111
Apéndice B: Transformación Exponencial para Convexificación del modelo (Vaselenak, 1987).....	117
Apéndice C: Datos para ejemplos.....	123
Apéndice D: Listado .....	125
<b>Resumen Bibliográfico</b> .....	132

## Índice de Tablas

<b>Tabla 2.1.</b> Comparación entre Procesos Continuos vs. Batch.....	8
<b>Tabla 2.2.</b> Diseño Retrofit multiproducto (Barbosa-Póvoa, 2007) .....	16
<b>Tabla 3.1:</b> Casos de estudio Capítulo 3 .....	52
<b>Tabla 3.2:</b> Resultados para el ejemplo 1 .....	55
<b>Tabla 3.3:</b> Resultados para el ejemplo 2.....	56
<b>Tabla 3.4.</b> Problema resuelto por diferentes modelos.....	58
<b>Tabla 4.1.</b> Estadísticas del modelo y solución de Ejemplo 1 para 10 períodos.....	68
<b>Tabla 4.2.</b> Estadísticas del modelo y solución de Ejemplo 2 para 20 períodos.....	70
<b>Tabla 4.3.</b> Casos Capítulo 4.....	79
<b>Tabla 4.4.</b> Resultados del algoritmo para diseño retrofit de plantas batch con corrección de planeación y horizontes móviles.....	80
<b>Tabla 4.4.</b> Resultados para tres puntos de demanda.....	82

## Índice de Figuras

<b>Figura 1.1.</b> Superestructura del planta de cerveza.....	4
<b>Figura 2.1.</b> Plantas Flowshop y Jobshop (Biegler, 1997).....	9
<b>Figura 2.2.</b> Programación de campañas de un solo producto y campañas mixtas (Biegler, 1997).....	10
<b>Figura 2.3.</b> Estructura de Bloque-Diagonal con restricciones y variables ligadas.....	18
<b>Figura 2.4.</b> Proceso general de elaboración de cerveza.....	24
<b>Figura 2.5.</b> Etapas del modelo. ....	28
<b>Figura 3.1.</b> Algoritmo disjunto de aproximación externa.....	40
<b>Figura 3.2.</b> Estructura para diseño retrofit de plantas batch multiproducto (Vaselenak, 1987).....	44
<b>Figura 3.3.</b> Solución al ejemplo 1 para cuatro períodos.....	54
<b>Figura 3.4.</b> Expansiones de planta para ejemplo 2 con 10 períodos.....	56
<b>Figura 3.5.</b> Tiempo de cómputo (seg. CPU) para algoritmo disjunto.....	57
<b>Figura 3.6.</b> Opción de utilizar un equipo para dos etapas.....	59
<b>Figura 4.1.</b> Algoritmo de horizonte móvil, Erdirik-Dogan (2007).....	65
<b>Figure 4.2.</b> Bloque detallado en horizonte de 20 años.....	67
<b>Figura 4.3.</b> Algoritmo para diseño retrofit de plantas batch con corrección de planeación.....	75
<b>Figura 4.4.</b> Algoritmo para diseño retrofit de plantas batch con corrección de planeación y horizonte móvil.....	77
<b>Figura 4.5.</b> Configuraciones de planta para diferentes demandas.....	81



## Agradecimientos

Gracias a mi familia, que siempre me apoya en cada uno de mis proyectos. A mis papás, por enseñarme quien soy! Gracias JC por siempre dejarme elaborar alguna aventura nueva para crecer. Gracias a Juan Pablo por darle sentido a tratar de ser mejor y trabajar incansablemente. Gracias JÓse por enseñarme como es la felicidad cada día. Gracias Pato por existir y ser mi gran regalo de Dios.

Agradezco a FIME, la oportunidad de tenerme como estudiante, en especial a todo el profesorado y alumnado de PISIS que me acompañó durante mis estudios doctorales.

Gracias al Dr. Chacón y a todos los miembros del comité de tesis, por tomarse el tiempo de aportar a mi trabajo, de manera que siempre se fue mejorando a lo largo de estos años. Gracias en especial al Dr. Igor Litvinchev por sus incontables contribuciones no solo para mi trabajo sino para mi persona y manera de pensar.

Gracias Ignacio por siempre estar dispuesto a contestar mis preguntas, platicar conmigo y compartir tus enormes recursos intelectuales y materiales conmigo.

Agradezco además el apoyo económico brindado por CONACYT y PAICYT de la UANL para la elaboración de esta tesis.

## Resumen

María Gabriela García Ayala.

Candidato para el Grado de Doctora en Ingeniería con Especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica

Título del estudio:

### Expansión de Capacidad de Plantas Tipo Batch

Número de páginas: 132

Objetivos y métodos de estudio:

- Desarrollar un modelo multiperíodo para optimizar la expansión de plantas tipo batch en donde se puedan alocar diferentes tipos de unidades para cada etapa dada del proceso.
- Aplicar modelo y proponer algoritmo de solución para un problema real de gran escala.
- Investigar ventajas/desventajas de utilizar programación disyunta.

Contribuciones y Conclusiones:

- ✓ Se propone un modelo multiperíodo que representa adecuadamente el problema de expansión de capacidad de plantas batch incorporando la posibilidad de manejar diferentes opciones de equipo para una etapa.
- ✓ Se utiliza programación disyunta como una herramienta útil para disminuir los tiempos de cómputo del problema maestro (DLP) al solucionar el modelo (MINLP).

Firma del Asesor:

---

Dr. Oscar Chacón Mondragón



## **1. Introducción**

### **1.1 Motivación**

Esta tesis doctoral se enfoca en el desarrollo de un nuevo modelo matemático para el diseño multiperíodo de plantas tipo batch y su respectivo algoritmo de solución. El objetivo final de este trabajo es encontrar un calendario de inversiones, en donde se defina el tiempo y los requerimientos de expansión de una planta batch de manera óptima, durante un horizonte de largo plazo.

El diseño de plantas batch multi-producto es un problema importante en la industria farmacéutica, alimentaria y de químicos especializados, y ha recibido atención considerable durante las pasadas décadas (Montagna, 2003; Pinto, Barbosa-Póvoa y Novais, 2005). A pesar de los modelos mejorados, los algoritmos especializados y recientes mejoras en equipo computacional y paquetes de optimización, el diseño de plantas batch multiproducto sigue siendo un problema complejo de modelar y de resolver. Típicamente se estudian casos para

la industria farmacéutica, en donde la flexibilidad de los equipos es valiosa, dada la corta vida que pueden tener los productos. Los modelos de expansión a largo plazo son inexistentes en ese tipo de industria. Por tal motivo, los esfuerzos de investigación en este tema han sido limitados y explorados sólo en dos trabajos que se basan en modelos de diseño de plantas batch multiperíodo: Van den Heever y Grossman (1999) y Moreno, Montagna e Iribarren (2007).

Con todo esto en consideración se escogió para este trabajo una industria en la cual este tipo de problema pudiera ser explorado en un entorno real: el aumento de capacidad para una planta cervecera. Contrario a lo que está reportado en la literatura, nuestro problema presenta dos diferencias importantes y nuevas en la literatura científica: 1) el equipo de una planta cervecera realiza las mismas operaciones por décadas, y 2) el volumen de producción es enorme, si comparamos lo que se produce de cerveza en un año contra lo que se puede producir de algún medicamento. Esto atribuye características intrínsecas a esta industria en particular y por ende al tipo de problema a resolver desde el punto de vista de optimización.

El modelar y solucionar un problema de optimización con estas características da origen a modelos complejos de optimización discreta/continua, cuyos requerimientos computacionales tienden a incrementarse exponencialmente con el tamaño del problema. El objetivo de este trabajo es poder abordar este problema y proponer un modelo con un algoritmo de solución con bases de programación disyunta y heurísticas.

Este capítulo resalta los objetivos de investigación y la definición de problema que es planteado para su solución; se concluye además, con un bosquejo por capítulo del resto de la tesis.

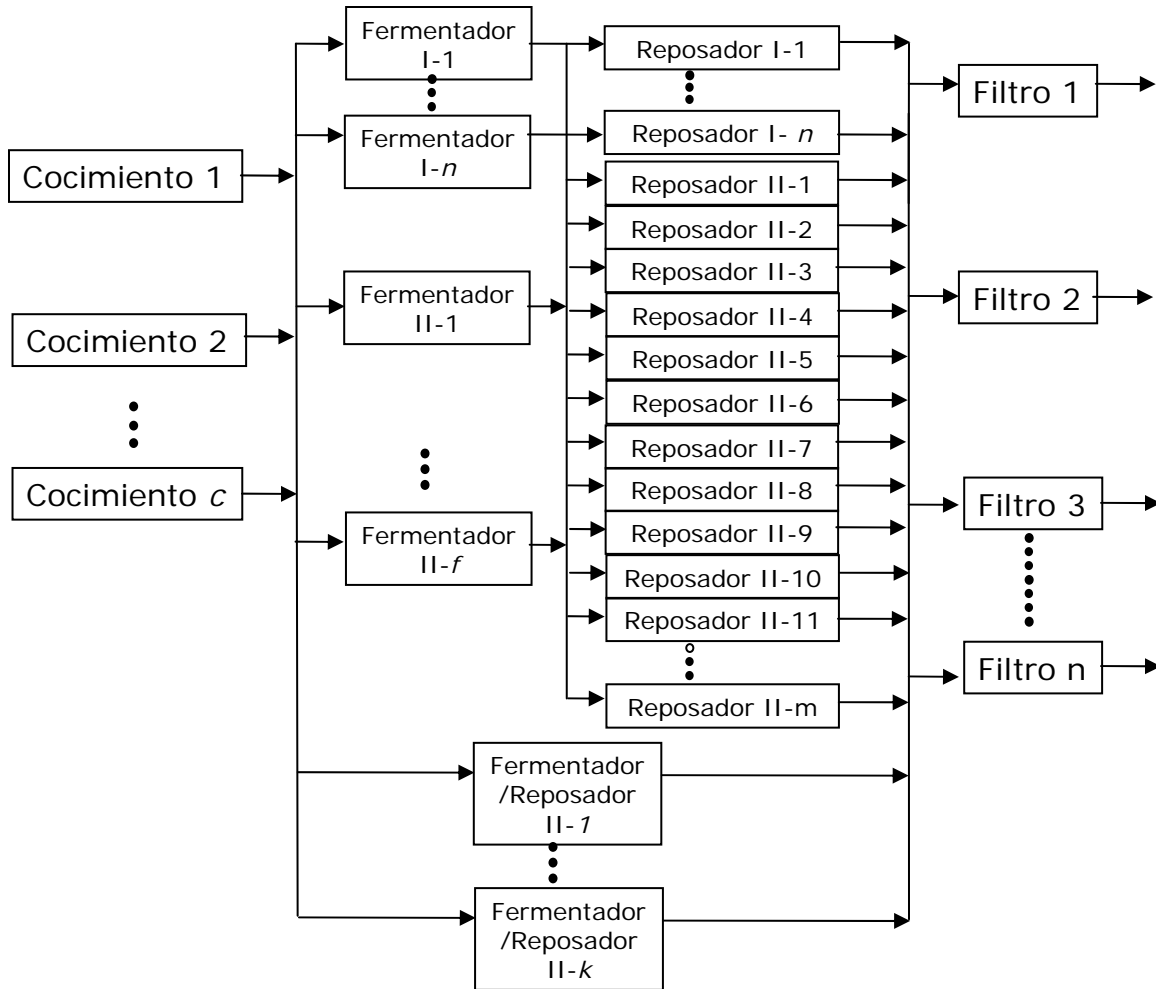
## 1.2 Definición del Problema

Dada una planta batch con una serie de equipos y productos con demanda creciente, se desea encontrar un programa de expansiones modulares que permita cumplir la demanda de los productos. El valor a optimizar es el costo de inversión por compras de equipo para cada período del horizonte de planeación.

La industria cervecera presenta problemas interesantes desde el punto de vista de modelación y de optimización ya que al ser un sistema de producción de tipo batch, cuenta con una serie de discontinuidades implícitas al proceso, las cuales se pueden representar por modelos del tipo entero mixtos.

Un caso específico dentro del diseño de plantas batch, es el de expansión de plantas existentes o “retrofit”. Es un problema de optimización cuyo objetivo es obtener una estructura de planta óptima nueva, a partir del equipo existente, de tal forma que se maximizan los beneficios, sujetos a una nueva demanda.

En general, la solución es un mapa de configuración de planta, donde equipo que no se usa se vende y equipo nuevo es adquirido y ajustado para trabajar con el existente. En la Figura 1.1 se presenta una superestructura de planta de cuatro etapas. Observar como el modelo de optimización tiene posibilidad de expandirse cuantas veces sea necesaria en cada una de ellas. Nótese además que la etapa 2 de Fermentadores y la 3 de Reposadores pueden ser elaboradas en un mismo equipo. El trabajo de esta tesis cae dentro de esta clasificación: diseño retrofit de plantas batch. Además, también es un problema multiproducto y multiperíodo.



**Figura 1.1** Superestructura de planta de cerveza.

El modelo matemático que describe al problema es un modelo de programación entero mixto no lineal y se presenta en el Capítulo 3.

### 1.3 Objetivos de Investigación

Para el problema de la planta cervecera, se consideran los siguientes objetivos particulares de la investigación.

1. Desarrollar un modelo multiperíodo para optimizar la expansión de plantas tipo batch.
2. Desarrollar un mecanismo para poder asignar diferentes tipos de unidades para una etapa dada del proceso.

3. Investigar ventajas y desventajas de utilizar programación disyunta.
4. Desarrollar un algoritmo de solución para el modelo matemático presentado.
5. Aplicar el modelo y el algoritmo de solución un problema real de gran escala.
6. Entregar a la empresa en cuestión un modelo para usarse como herramienta, que genere resultados útiles en su toma de decisiones.

## 1.4 Resumen de Tesis

En el Capítulo 1, se describe el objetivo de esta tesis, el problema a abordar, así como sus metas de investigación y su posicionamiento y contribuciones al área. Para el Capítulo 2, se incluye una descripción general sobre plantas batch y su diseño. Se introdujo el área de diseño conocido como retrofit, al cual este trabajo pertenece. Se explican las dos técnicas matemáticas involucradas en la tesis, recordando los problemas de programación entera mixtas no lineal (MINLP) y los conceptos básicos de programación disyunta. Por último, se explica el problema industrial que da origen a esta tesis.

En el Capítulo 3 proponemos un modelo multiperíodo para el retrofit de plantas batch basado en el modelo general de Van den Heever y Grossmann (1999). Aquí describimos el modelo matemático y las suposiciones bajo las cuales trabaja. Después de describir el modelo se demuestra como puede ser convertido en un modelo de programación disyunta y este, a su vez, por manipulaciones matemáticas y lógicas, se puede representar como un MINLP para aprovechar las herramientas que ofrece semejante manipulación. La efectividad de este modelo se demuestra con experimentación numérica en instancias reales y simuladas.

En el Capítulo 4 nos enfocamos sobre el modelo de retrofit disyunto presentado en el Capítulo 3. Se le hace una desagregación para tomar en cuenta la demanda catorcenal de los productos. Además, dado que el modelo del Capítulo 3 presenta complicaciones con períodos



considerables de tiempo, se propone el uso de un método de horizonte móvil a fin de poder cubrir el horizonte de tiempo deseado por el problema. Los resultados muestran un error de optimalidad pequeño y soluciones fuertes para el problema.

El Capítulo 5 presenta las conclusiones sobre los diferentes aspectos del problema de diseño de retrofit para plantas batch que estudiamos en esta tesis de doctorado; así como los problemas que surgieron del modelo en uso, y la manera concreta en que se resolvió cada problema. Por último, se hace una recopilación de la tesis, de los resultados de la investigación y de sus aportaciones al área de ingeniería de sistemas de proceso y se discuten posibles áreas para trabajo futuro.

## **Capítulo 2. Antecedentes**

### **2.1 Introducción a procesos batch**

Los procesos químicos pueden ser clasificados como “batch” o por lotes, continuos o semi-continuos. Este trabajo trata con procesos tipo batch. En los procesos batch, la alimentación se hace al equipo al principio del proceso, y el contenido es removido en algún tiempo posterior. No hay flujos que crucen los límites del sistema entre el tiempo de alimentación y el de descarga (Felder y Rosseau, 2000).

Los procesos batch son utilizados en la manufactura de químicos especializados, productos farmacéuticos, alimenticios y ciertos tipos de polímeros. De hecho, los procesos

batch se adjudican el 45% de todos los procesos químicos (Reeve, 1995). La Tabla 2.1 muestra una comparación entre los procesos continuos y los procesos batch.

**Tabla 2.1.** Comparación entre procesos Continuos vs. Batch.

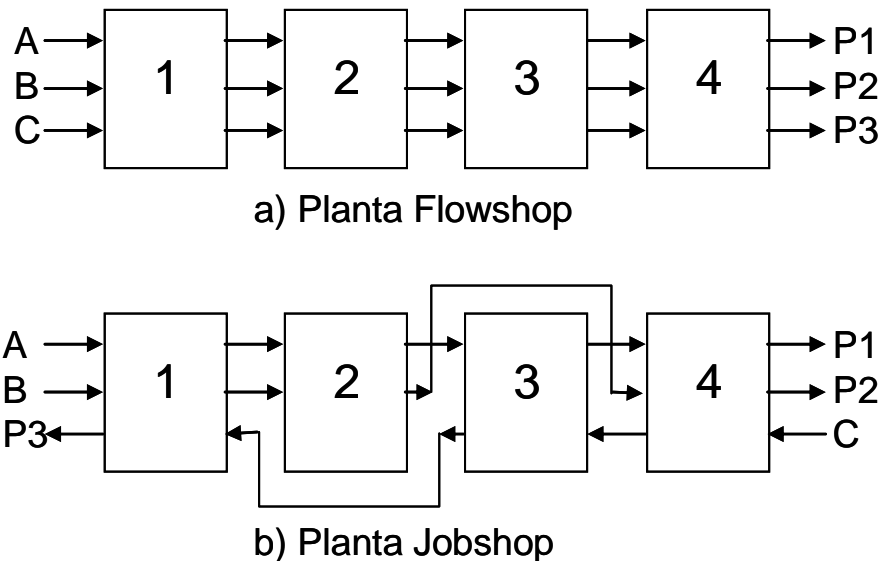
Procesos Continuos	Procesos Batch
<ul style="list-style-type: none"> <li>• Condiciones de operación estacionarias</li> <li>• Gran escala de producción</li> <li>• Cada equipo realiza una operación o función específica</li> <li>• Alta Automatización</li> <li>• Calidad del producto constante</li> <li>• Velocidad de producción constante</li> </ul>	<ul style="list-style-type: none"> <li>• Funcionamiento intermitente</li> <li>• Existe un ciclo de operación</li> <li>• Pequeña escala de producción</li> <li>• Plantas flexibles multiproducto y multipropósito</li> <li>• Grandes tiempos de procesamientos o residencia</li> <li>• Reacciones lentas flujos pequeños</li> <li>• Calidad del producto variable</li> <li>• Productos de alto valor y calidad</li> <li>• Condiciones de control estrictas</li> <li>• Velocidad de producción variable</li> <li>• Intensivo en mano de obra</li> <li>• Procedimientos de síntesis complejos</li> </ul>

Las plantas batch generalmente son plantas multi-producto, lo cual se traduce a que todos o varios de los productos comparten equipo. Esto, a su vez, requiere que la producción sea asignada a un calendario, dicha asignación puede tener un gran impacto económico.

En la mayoría de las ocasiones, la manufactura de un producto en una planta batch se lleva a cabo siguiendo una receta especificada por un conjunto de tareas a realizarse con condiciones de operación y tiempos de producción fijos. Las recetas también son comunes

debido a que, en la industria de alimentos y farmacéutica, son requeridas por regulaciones gubernamentales.

Cuando las plantas batch son multi-producto, éstas se clasifican en dos tipos: “flowshop”, en donde todos los productos siguen todas las etapas en el mismo orden y “jobshop”, donde no todos los productos requieren de todas las etapas o siguen la misma secuencia, ver Figura 2.1. Se trata de una planta con tres materiales: A, B y C, que requieren de cuatro etapas de proceso: 1, 2, 3 y 4, para producir tres productos P1, P2 y P3. En el caso a) todos los productos siguen las etapas en el mismo orden, en el caso b) las etapas se pueden llevar a cabo en una secuencia diferente entre ellas.

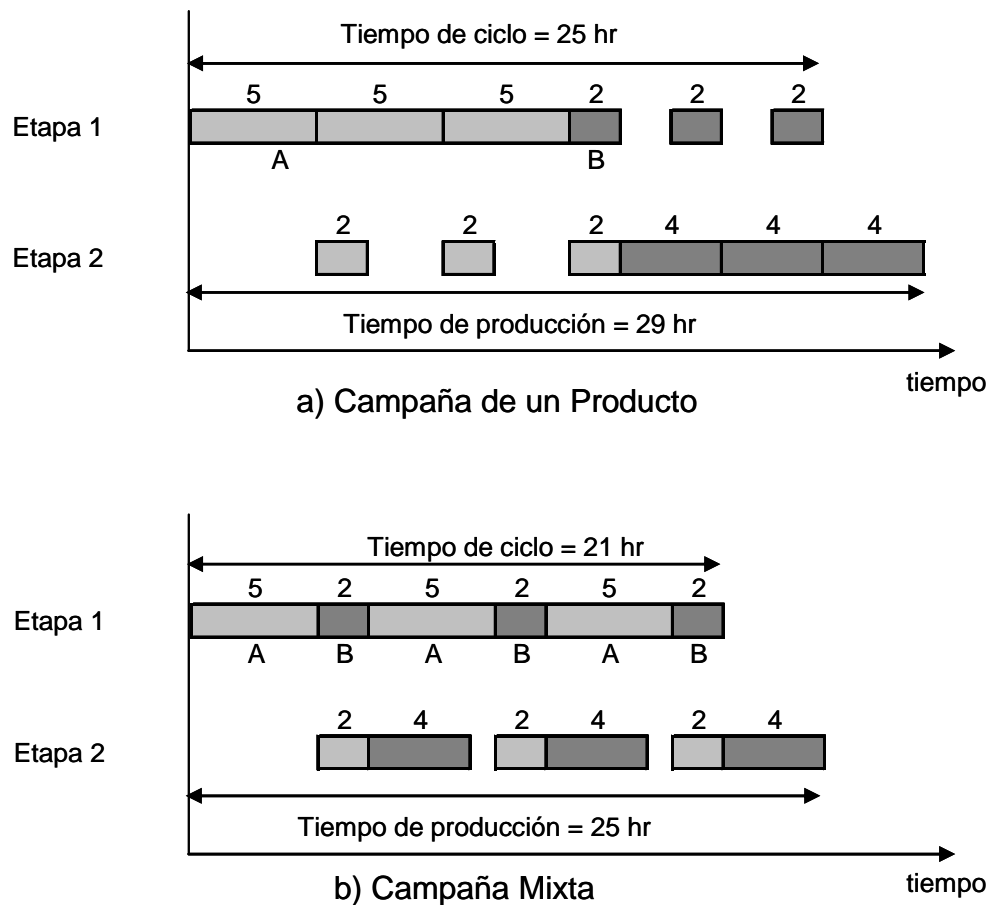


**Figura 2.1.** Plantas Flowshop y Jobshop (Biegler, Westerberg y Grossmann, 1997)

Cabe mencionar que las plantas flowshop con frecuencia son referidas como plantas multi-producto, mientras las jobshop son referidas como plantas multi-propósito (Biegler, Westerberg y Grossmann, 1997).

Un aspecto importante en las plantas flowshop es el tipo de campaña de producción que se utilice para producir un número determinado de batches para los diferentes productos. Una opción es utilizar campañas de un solo producto SPC (por sus siglas en inglés: “Single Product Campaign”), en donde todos los batches de un producto son elaborados juntos, antes de cambiar a otro producto. La otra opción es utilizar campañas mixtas (MPC, por sus siglas en inglés: “Mixed Product Campaign”), en donde los batches se producen según una secuencia pre-establecida permitiendo cambiar de producto entre un batch y el siguiente.

Para claridad del punto considérese el siguiente ejemplo: se desean producir tres batches de dos productos, A y B, en una planta con dos etapas.



**Figura 2.2.** Programación de campañas de un solo producto y campañas mixtas (Biegler, Westerberg y Grossmann, 1997)

El tiempo de procesamiento del producto A es de 5 horas en la primera etapa y 2 horas en la segunda. El tiempo de procesamiento del producto B es de 2 horas en la primera etapa y 4 horas en la segunda. En la Figura 2.2 se puede observar la diferencia entre utilizar las campañas de un solo producto y las mixtas.

Adicionalmente, para un modelo de planta batch se necesita definir que tipo de almacenamiento se va a utilizar. Estas políticas afectarán el diseño y operación de cualquier modelo batch. En el pequeño ejemplo de la Figura 2.2 la política utilizada fue la de “sin espera” (ZW, por sus siglas en inglés “zero wait”) lo cual establece que al terminar una tarea de procesarse en una etapa, debe inmediatamente empezar a ser procesada en la etapa siguiente.

El almacenamiento del producto en etapas intermedias es considerado como parte de descripción del proceso batch, entre los cuales se presentan a continuación cinco tipos esquema de almacenamientos (Shah, 1998; Voudouris y Grossmann, 1996):

- Sin espera (ZW, *Zero-Wait*), El material no es estable y se debe procesar inmediatamente después de ser producido.
- Almacenaje intermedio ilimitado (UIS, *Unlimited Intermediate Storage*). El material es estable y existen uno o más contenedores con capacidad ilimitada.
- Almacenaje intermedio finito (FIS, *Finite Intermediate Storage*). El material es estable y existe uno o más contenedores con capacidad finita.
- Almacenaje intermedio compartido (SIS, *Shared Intermediate Storage*). El material es estable y puede almacenarse en uno o más contenedores que pueden ser compartidos con otros productos, más no simultáneamente.
- Sin almacenaje intermedio (NIS, *No Intermediate Storage*). El material es estable pero no hay contenedores disponibles; sin embargo el material puede quedarse temporalmente en el equipo que lo produjo.

La regla general de desarrollo de modelos batch indica que antes de empezar un modelo se debe definir si se permite almacenar productos intermediarios entre las etapas y bajo que tipo de almacenaje se va a operar. Estas opciones permiten incrementar la eficiencia del uso de los equipos. Un último aspecto cuando se trata de diseño de plantas batch, es si se permite que las unidades trabajen de forma desfasada, o si sólo pueden trabajar de forma paralela.

### **2.1.1 Diseño de Plantas Batch**

En su forma más general, la producción batch involucra una localidad general en donde una variedad de productos son generados compartiendo los recursos disponibles como lo son mano de obra, equipos, servicios, etc. Diferentes secuencias que elaboran el mismo producto pueden ser realizadas en diferentes equipos, en diferentes tiempos.

En 1989, Reklaitis hizo una revisión sobre el tema de diseño de plantas batch, reportando 50 referencias con fechas entre 1972 y 1989. A partir de este trabajo, el problema de diseño de plantas batch es analizado desde tres principales perspectivas: el diseño determinístico (Georgiadis, Rotstein y Macchietto, 1997; Dedieu et al., 2003; Dietz et al., 2008), el diseño bajo incertidumbre (Pistikopoulos, Thomaidis e Ierapetritou, 1996; Epperly, Ierapetritou y Pistikopoulos, 1997; Petkov y Maranas, 1998) y la integración general del proceso (Lin y Floudas, 2001; Macchietto, 2005; Pournali, Amidpour y Rashtchian, 2006; Puigjaner y Guillen-Gosálbez, 2008; Cavin, Fischer y Hungerbuhler 2008).

Además de las perspectivas de diseño de plantas batch, Reklaitis (1989) separa el problema de diseño en cuatro niveles de decisión:

- determinación de la estructura del proceso que involucra la definición de la receta del producto y las particiones de las tareas para cada producto,

- selección de la mejor estrategia de operación en donde se toma la decisión del modo de operación a utilizarse en los diferentes productos involucrados,
- asignación de equipo a actividades o almacenamiento de materiales,
- determinación del tamaño de los equipos, incluyendo unidades de proceso, unidades intermedias e interconexiones entre la planta.

Bajo estas consideraciones de clasificación, el problema abordado en esta tesis se modela en forma determinista cubriendo los cuatro niveles de decisión.

En 2007, Barbosa-Póvoa realiza una revisión más actual del tema, cubriendo 117 referencias, tomando como punto de partida la revisión de Reklaitis y cubriendo hasta el año 2006. Barbosa-Póvoa define al problema de diseño de plantas batch de forma general como sigue. Dadas:

- las recetas de los productos describiendo la producción de uno o más productos,
- los posibles equipos a instalar en la planta y sus opciones para realizar diferentes actividades,
- el tiempo del horizonte de planeación,
- los recursos disponibles a lo largo del horizonte de planeación (materias primas, transporte, dinero, almacén, etc.),
- los inventarios disponibles y sus restricciones,
- demandas y entregas a lo largo del horizonte de planeación,
- políticas de almacenamiento,
- datos de costo de operación e inversión de capital relacionados con las inversiones y/o instalaciones de la planta,

determinar:

- la configuración óptima de la planta (i.e., número y tipo de equipos en cada etapa),
- un calendario del proceso utilizando recursos para cumplir con la producción requerida. (i.e., tamaño y número de batches),

de tal manera que se optimiza el objetivo de la planta, cualquiera que éste pudiera ser.



El diseño de plantas batch multi-producto ha sido estudiado por diversos autores con diferentes objetivos (Sargent y Grossmann, 1979; Karimi y Modi, 1989; Vecchiotti y Montagna, 1998; Montagna, 2003). Típicamente, el análisis de procesos batch va enfocado a productos especializados donde la vida de los productos es corta, un par de años, y los volúmenes manejados son pequeños. Con tal consideración en mente, se procura que las plantas sean lo más flexible posibles, de modo que un equipo pueda ser utilizado en diferentes líneas de producción y de diversas formas. Existen varias publicaciones sobre flexibilidad (Straub y Grossmann, 1992; Pistikopoulos, Thomaidis e Ierapetritou, 1996; Ravemark y Rippin, 1998; Rotstein, Lavie y Lewin, 1996; Epperly, Ierapetritou y Pistikopoulos, 1997; Bernal-Haro et al., 2002).

El caso de expansión de capacidades se le conoce como el problema de retrofit (Yoo et al., 1999) y no ha sido tan estudiado. El diseño retrofit puede ser definido como el rediseño de una planta actual a una de mejor configuración de tal manera que se mejore la economía del proceso. Dicho rediseño se puede lograr aumentando la producción, reduciendo los costos de operación o incrementando la flexibilidad de los procesos.

El problema de retrofit es un problema de optimización cuyo objetivo es obtener una estructura de planta óptima a partir de una configuración actual, de tal manera que se maximicen los beneficios sujetos a una demanda. En general, el problema de retrofit obtiene una configuración de planta nueva donde el equipo que no se usa es vendido y unidades nuevas son adicionadas y ajustadas para trabajar con las existentes, para configurar el nuevo proceso (Montagna, 2003).

Plantas batch multi-producto producen un conjunto de productos usando los mismos equipos, operando en la misma secuencia. Como los productos son diferentes entre sí, cada

unidad es compartida por todos los productos, pero no utilizan el total de su capacidad para todos ellos. Cada unidad trabaja a su máxima capacidad solamente cuando está procesando los productos para los que esa etapa sea su limitante volumétrica. De la misma manera, las etapas no trabajan todo el tiempo para todos los productos. Una etapa trabaja todo el tiempo (cargando un batch nuevo tras otro) solamente cuando se encuentra procesando el producto para el cual esta etapa es limitante o es definido cuello de botella, en términos de tiempo, definiendo así el tiempo de ciclo para ese producto. En un modelo de optimización del diseño de plantas batch lo que se desea es optimizar el ritmo de producción, el cual depende del tamaño del batch y del tiempo de ciclo. Existe bibliografía extensa para los dos casos. Por ejemplo, Ravermark y Rippin (1998) introducen equipo en paralelo de forma que reduce el tiempo de ciclo. Karimi y Modi (1989) introducen etapas de almacenamiento que permiten incrementar la utilización de los equipos aumentando el tamaño del batch.

La primera publicación sobre diseño retrofit (Vaselenak, Grossmann y Westerberg, 1987) consideraba la adición de unidades batch a una planta existente para satisfacer incrementos en la demanda. Fletcher, Hall y Johns (1991) modificaron el modelo de Vaselenak et al. permitiendo que las unidades nuevas operaran tanto de forma paralela como de forma desfasada.

Yoo et al. (1999) presentaron un modelo más completo, donde se deja la libertad de disponer de equipo no usado y utilizar nuevas unidades en diferentes esquemas de producción.

Montagna (2003) incluyó además, la opción de utilizar tanques de almacenamiento intermedio, lo cual da soluciones más eficientes y parecidas a la realidad; sin embargo, aumentó considerablemente la complejidad del modelo.

Sólo dos referencias de expansiones de capacidad de plantas batch considerando una serie de períodos de tiempo han sido reportadas.

La primera es un modelo de programación disyunta presentado por Van den Heever y Grossmann (1999), dónde presentan un modelo general para redes de proceso y aplican una serie de algoritmos para su resolución. Éste modelo general puede abarcar al problema de diseño de plantas batch, y por ende al retrofit del mismo.

La segunda, publicada por Moreno, Montagna e Iribarren (2007), es un modelo más enfocado a planeación, dónde el diseño ocurre en el período uno, y en los demás períodos se obtienen las variables de planeación como inventarios, consumos de materias primas, etc.

**Tabla 2.2.** Diseño retrofit multi-producto (Barbosa-Póvoa, 2007)

<b>Bibliografía</b>	<b>Características del Modelo</b>	<b>Tipo de modelo/métodos de solución</b>
Vaselenak et al. (1987)	Productos con mimos modo de operación	MINLP
España y Puigjaner (1989)	Tiempos de proceso dependientes del tamaño del batch. Campañas de un solo producto. Produce tanto desfasado como en-fase. Sin almacenamiento.	Heurístico
Fletcher et al. (1991)	Estrategia de operación independiente para cada producto. Sin almacenamiento.	MILP
Petkov y Maranas (1998)	Demanda de productos con incertidumbre. Campañas de un solo producto. Expansiones de capacidad por etapas.	Modelo de dos niveles: nivel interno que define operación; y nivel externo definiendo diseño/capacidad
Yoo et al. (1999)	Permite modificar el modo de operación de unidades existentes. Sin almacenamiento.	MINLP resuelto por DICOPT ++
Van den Heever y Grossmann (1999)	Simultáneamente diseño y expansión así como diseño de operación, expansión y planeación.	Modelo de optimización disyunto, multiperíodo no-lineal resuelto por algoritmo lógico de descomposición a dos niveles
Montagna (2003)	Incluye unidades de procesamiento y almacenaje.	MINLP resuelto por DICOPT ++
Goel et al. (2004)	Confiabilidad de equipo y su mantenimiento.	MINLP

La Tabla 2.2 muestra la bibliografía existente sobre el problema de retrofit de plantas batch, la cual es tomada de Barbosa-Póvoa (2007).

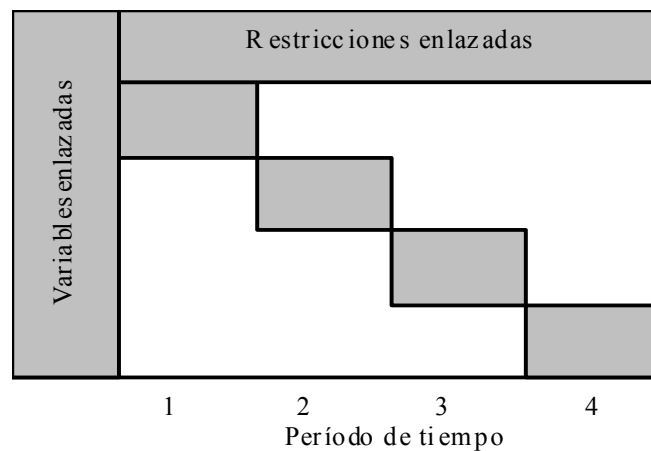
El trabajo de esta tesis se puede definir como un modelo multiperíodo de retrofit, obteniendo simultáneamente cotas de operación. Es un MINLP y tiene la peculiaridad de permitir la introducción diferentes tipos de tecnología en cada etapa para cada equipo nuevo.

## **2.2 Estrategias Computacionales**

### **2.2.1 Programación Multiperíodo Entera-Mixta**

Los modelos para optimización multiperíodo típicamente tienen un objetivo, como minimizar costo, sujeto a restricciones en la forma de ecuaciones y usualmente involucran variables tanto discretas como continuas. Las variables continuas pueden ser variables de estado, representando condiciones de operación tales como flujos y temperaturas dependientes del tiempo, o variables de diseño representando tamaños de equipo. Las variables discretas son, por lo general, binarias (0,1) y representan decisiones discretas, por ejemplo el invertir en una unidad o el operar un equipo dado en un período determinado. Las ecuaciones que representan estos modelos pueden ser lineales, por ejemplo los balances de masa, o no lineales como ecuaciones de eficiencia del proceso. Las restricciones pueden ser válidas para todos los períodos, o para alguno en particular, y algunas variables y/o restricciones pueden enlazar períodos de tiempo (Figura 2.3), evitando una posible solución por descomposición donde los períodos de tiempo fueran resueltos independientemente. Modelos involucrando variables binarias y continuas con ecuaciones no lineales son clasificados como Modelos de Programación Entera Mixta No Lineal (MINLP, por sus siglas

en inglés “Mixed Integer Non Linear Programming”). Para un compendio de modelos MINLP ver Grossmann y Kravanja (1997). Los modelos MINLP son notables por ser NP-completos (Bussieck y Pruessner, 2003), significando que cualquier algoritmo que pretende encontrar una solución óptima al problema requiere de un tiempo de ejecución que crece exponencialmente con el tamaño del problema para el peor caso. Para modelos MINLP multiperíodos, los tiempos de solución rápidamente se vuelven intratables, particularmente si los modelos involucran un incremento en el número de variables binarias para cada período adicional. Por ello, existe una necesidad de desarrollar algoritmos y modelos más eficientes, capaces de resolver problemas de tamaño considerable.



**Figura 2.3.** Estructura de bloque diagonal con restricciones y variables enlazadas

DICOPT++, el algoritmo de solución de MINLP utilizado en el paquete comercial de optimización GAMS (Brooke, Kendrick y Meeraus, 1992), es la base para solución de los modelos de esta tesis. Utiliza el algoritmo de aproximación externa (Outer-Aproximation, OA) de Durán y Grossmann (1986) y es una extensión del trabajo de penalidad aumentada igualdad relajada de Viswanathan y Grossmann (1990). El algoritmo requiere iterar entre un sub-problema de Programación No Lineal (NLP), donde las variables binarias son fijadas, y

un problema maestro entero-mixto lineal donde el modelo es linealizado en el punto de solución del NLP. Una técnica de descomposición similar para problemas entero mixtos es la descomposición de Benders (Generalized Benders Decomposition, GBD) (Geoffrion, 1972) que involucra la solución de sub-problemas NLP y problemas maestros pseudos-enteros. Lo anterior corresponde a la proyección del espacio completo entero-mixto del OA al espacio de únicamente las variables binarias (Quesada y Grossmann, 1992), y es por esta proyección que el problema maestro de OA predice cotas más fuertes que las del problema maestro de GDB.

Para el caso de problemas considerados en esta tesis, problemas multiperíodos retrofit para plantas batch, donde un gran número de variables discretas se adicionan con cada período adicional, el cuello de botella en la solución del problema por el algoritmo OA (o GBD) es el problema maestro MILP, debido al gran número de variables discretas. Por tal motivo se pretende abordar el reto de reducir el tiempo de solución del MILP.

Sahinidis et al. (1989) presentan modelo MILP multiperíodo para planeación a largo plazo en la industria química que solucionaron con técnicas que ayudan a ramificar y acotar, planos cortantes, GBD y heurísticos. Otros métodos usados para solucionar MILPs incluyen generación de restricciones y proyecciones utilizadas en conjunto con un algoritmo de planos cortantes (Sahinidis y Liu, 1996), o una estrategia de descomposición a dos niveles para la síntesis y planeación de redes de procesos (Iyer y Grossmann, 1998). Dedopoulos y Shah (1996) presentaron formulaciones MILP y MINLP para planeación de mantenimientos utilizando el enfoque de agregación para reducir el tamaño del problema. Varvarezos et al. (1992) propusieron un método de descomposición para el algoritmo OA para MINLPs convexos, donde al sub-problema NLP es descompuesto aun más en un problema maestro

linealizado y en un problema de NLP primal con penalidades para asegurar factibilidad. Algoritmos basados en GBD también han sido empleados para solucionar MINLP multiperíodos en problemas de síntesis y planeación (Papalexandri y Pistikopoulos, 1994). Paules y Floudas (1992) propusieron un procedimiento de soluciones anidadas combinando los algoritmos GBD y el OA, aplicándolo a secuencias de destilación con integración de calor.

Van den Heever y Grossmann (1999) han sido prácticamente los únicos que, con su algoritmo, específicamente tratan de reducir el esfuerzo computacional del problema maestro MILP en problemas multiperíodos. En este algoritmo, se presenta un modelo general de programación disyunta para el problema de diseño de procesos químicos, siendo el problema multiperíodo retrofit para plantas batch una aplicación del mismo. Utilizan lógica y programación disyunta para considerar todas las alternativas posibles obteniendo muy buenos resultados para problemas grandes.

En la siguiente Sección 2.2.2, se discuten algunas de las estrategias para ayudar con el problema de tamaño del MINLP son discutidas. Las siguientes estrategias han surgido para resolver problemas de optimización discreta/continua con técnicas basadas en lógica: Programación Disyunta Generalizada (GDP, por sus siglas en inglés General Disjunctive Programming) (Raman y Grossmann, 1994), Programación Mixta Lógica Lineal (MLLP por sus siglas en inglés Mixed Logic Linear Programming) (Hooker y Osorio, 1999), y Programación de Restricciones (CP, por sus siglas en inglés Constraint Programming) (Hentenryck, 1989; Cardoso, Salcedo y Feyo de Azevedo, 1997).

Las motivaciones para estas tres estrategias con bases de lógica han sido facilitar la modelación, reducir el esfuerzo de la búsqueda combinatoria y mejorar el uso de las no-

linealidades. Una revisión general de optimización basada en lógica se puede encontrar en Hooker (2000). En este trabajo nos concentraremos en GDP, principalmente porque MILP y CP se enfocan a modelos lineales.

### **2.2.2 Programación Disyunta**

Programación disyunta es una representación alternativa a la programación entera mixta. Una disyunción es un conjunto de restricciones de las cuales por lo menos una debe ser válida (Raman y Grossmann, 1994). Un programa entero mixto (Mixed Integer Problem, MIP) puede ser derivado de una disyunción tomando la formulación del envolvente convexo para cada una de las disyunciones (Balas, 1985) o usando la formulación de la gran M (Lee y Grossmann, 2003). La idea básica de ambos métodos es expresar restricciones (igualdades y desigualdades) en términos de restricciones globales que siempre se mantienen, disyunciones que corresponden a restricciones condicionales en el espacio continuo y proposiciones lógicas en el espacio discreto. Todas estas restricciones son expresadas en función de variables continuas y booleanas que son seleccionadas para optimizar una función objetivo dada. Türkay y Grossmann (1996) propusieron un algoritmo lógico de aproximación externa (OA) para la solución de problemas GDP no lineales para optimizar redes de proceso. Este algoritmo resuelve un número de sub-problemas NLPs con una cantidad fija de opciones discretas, y luego inicializa el problema maestro MILP con la linealización de las restricciones no lineales en la disyunción.



Sea la ecuación siguiente una disyunción representada por el uso de los operadores lógicos OR ( $\vee$ ) y NOT ( $\neg$ ):

$$\left[ \begin{array}{c} w_{jt} \\ h_{jt}(x) \leq 0 \\ c_{jt} = \alpha_{jt} \end{array} \right] \vee \left[ \begin{array}{c} \neg w_{jt} \\ B^{jt} x = 0 \\ c_{jt} = 0 \end{array} \right] \quad \forall j, t.$$

La ecuación anterior puede ser interpretada como sigue: si la unidad  $j$  es operada en el período  $t$  ( $w_{jt} = \text{verdadero}$ ), entonces aplica la ecuación  $h_{jt}(x)$  describiendo la unidad y aplica un costo fijo  $\alpha_{jt}$ . Si la unidad no es operada ( $w_{jt} = \text{falso}$ ), el costo fijo y un subconjunto de variables son fijadas en cero por la matriz  $B^{jt}$ .

Las relaciones lógicas entre las variables discretas que describen las conexiones e interacciones entre las unidades, son dadas por las proposiciones lógicas consistiendo de variables y los operadores lógicos (tales como OR ( $\vee$ ), AND ( $\wedge$ ), NOT ( $\neg$ ) e IMPLY ( $\Rightarrow$ )).

Por ejemplo,  $y_1 \Rightarrow y_2 \wedge y_3$

significa que la selección de la unidad 1 ( $y_1=1$ ) implica la selección de la unidad 2 ( $y_2=1$ ) y la selección de la unidad 3 ( $y_3=1$ ). De manera similar, si  $z_{jt}$  representa la expansión de la unidad  $j$  en tiempo  $t$ , entonces:

$$y_j \Rightarrow \vee_t z_{jt}$$

significa que la selección de unidad  $j$ ,  $y_j$ , implica la expansión de esa unidad  $j$ , en al menos uno de los períodos.

Para el caso de modelos no lineales, Lee y Grossmann (2000, 2001) propusieron un algoritmo de optimización para resolver programas generales de GDP involucrando múltiples términos en las disyunciones. Este algoritmo está basado en el envolvente convexo de las disyunciones relajadas. Las reformulaciones llevan a problemas MINLP con cotas fuertes,

mientras los algoritmos por lo general involucran métodos de ramificación y acotamiento donde la ramificación es realizada en las disyunciones. De manera similar a los modelos MINLP, hay dificultades prácticas y teóricas cuando igualdades no lineales están presentes en las restricciones.

Para problemas MINLP no convexos, que tienen restricciones cuadráticas en forma de igualdades, Adjiman, Androulakis y Floudas (2000), Tawarmalani y Sahinidis (2002) y Kesavan y Barton (2000) han propuesto algoritmos de optimización global, basados en el método de ramificar y acotar de forma espacial. Smith y Pantelides (1999) propusieron técnicas automáticas de reformulación para convertir MINLP algebraicos a problemas MINLPs con términos cuadráticos, fraccionales y cóncavos separables.

## **2.3 Aplicación Industrial**

### **2.3.1 Breve introducción al proceso cervecero (McCabe, 1999)**

La cerveza es un producto fermentado que se vende para consumo humano. Esta se elabora a partir de agua, lúpulo, malta, adjuntos y levadura. El proceso de su elaboración a nivel industrial puede ser dividido en tres etapas principales: cocimientos, fermentación/reposo y envasado, (ver Figura 2.4).

La etapa de cocimientos, es el proceso mediante el cual los componentes insolubles contenidos en las materias primas son convertidos en productos solubles y particularmente en azúcares fermentables (mosto), los cuales serán aprovechados por la levadura durante la etapa de fermentación para formar alcohol y gas carbónico. Al final de lo que llamamos etapa

de cocimiento, el mosto es enfriado y es inoculado con levadura para empezar la etapa de fermentación.

Durante la etapa de fermentación, la levadura, como parte de su metabolismo, convierte los azúcares del mosto en alcohol etílico y dióxido de carbono principalmente, además de formar varios cientos de diversos compuestos que contribuyen al olor y sabor final de la cerveza. Típicamente, la fermentación se lleva al cabo entre 7 y 21 días. Al final de la etapa de fermentación, la cerveza es enfriada y transferida a una etapa de reposo para darle las características finales del producto. Esta parte del proceso es equivalente al añejamiento de un vino y en el caso de cerveza esta etapa comprende entre 3 y 28 días.

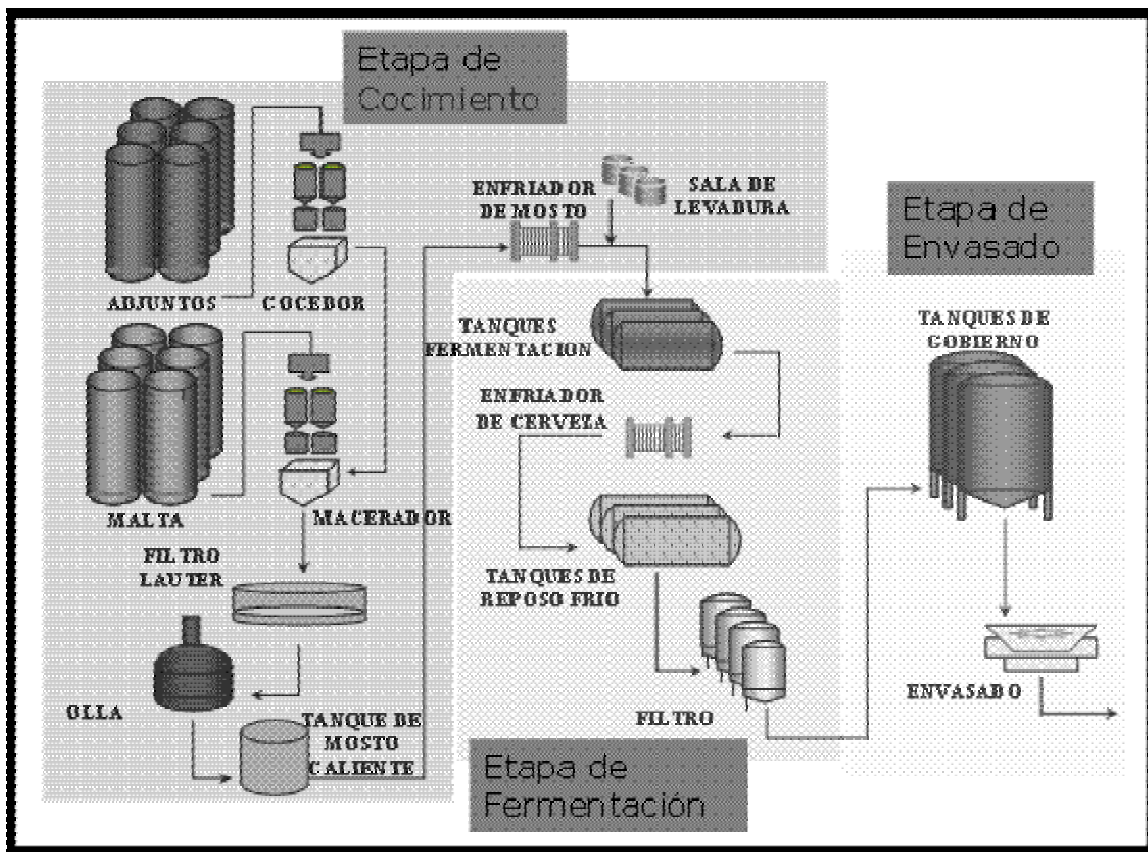


Figura 2.4. Proceso general de elaboración de cerveza (Cabada, 2004)

La última etapa es la de envasado, en donde la cerveza terminada es empacada en las diferentes presentaciones para consumo final.

### **2.3.2 Aplicación Industrial**

Dado que la demanda de cerveza sigue una curva suave y ascendente, siempre hay necesidad de ir ajustando la capacidad de la planta para cumplir con la nueva demanda. Esto no es nada nuevo, dicho ajuste de capacidad se ha hecho por décadas, la cervecería con la cual estamos trabajando tiene más de cien años y siempre ha escalado su capacidad agregando tanques a juicio del encargado de las operaciones en turno. La idea de éste trabajo es otorgarles una herramienta metodológica donde se obtiene un calendario de inversiones de tanques.

Actualmente, la manera en la que se hace este escalamiento es una metodología sencilla aunque algo laboriosa. Básicamente el escalamiento tendría los siguientes pasos:

1. Se asume un solo producto y se maneja el total del volumen de todos los productos como la demanda a satisfacer.
2. Se calcula etapa por etapa la capacidad volumétrica, sumando el volumen de todos los tanques.
3. Se obtiene una velocidad promedio de producción de los productos en cada una de las etapas basada en datos históricos.
4. Se identifica el cuello de botella.
5. Se decide por cuanto expandir la etapa limitante o cuello de botella y se expande.

Este proceso se repite cada vez que la capacidad de la planta se encuentre topada por la demanda. El algoritmo heurístico tiene las ventajas siguientes:

- Es sencillo de comprender, por lo cual no se necesita un experto para hacerlo ni para explicarlo o entenderlo.
- No necesita de algún paquete computacional específico, en cualquier hoja de cálculo se puede realizar, o en el peor de los casos, a mano.
- Al deshacer un cuello de botella con una expansión, el análisis ya está listo y se tiene identificada cual será el siguiente cuello de botella de manera automática antes de que éste se presente.

A pesar de sus ventajas, esta metodología utilizada por la industria tiene problemas serios.

A continuación se presenta un análisis exhaustivo de los problemas que se presentan al hacer expansiones de capacidad de esta forma, con la heurística. Complementamos cada punto con nuestra propuesta para su solución.

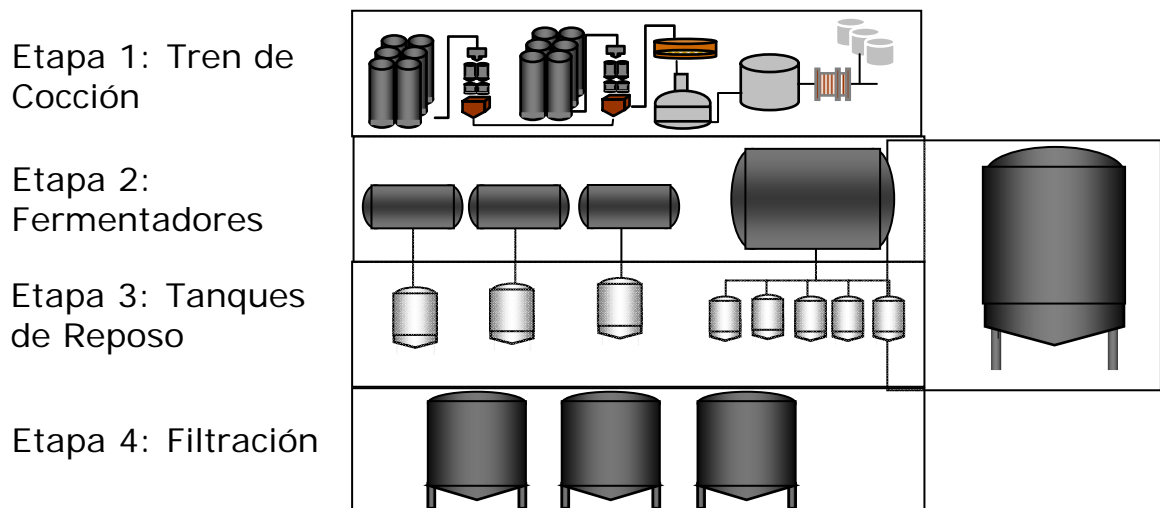
En primer lugar, la industria cervecera estudiada, no aprovecha la ventaja de toda la información disponible.

- La empresa no tiene un punto de pronóstico de demanda, tiene una curva de muchos años, con lo cual planea sus inversiones, metas de producción, utilidades, etc. ¿Porqué entonces sólo utilizar un punto de demanda para las expansiones? Este punto está directamente ligado con lo complejo que un sistema batch de este tipo es en su funcionamiento y diseño, el entendimiento no es trivial. Básicamente no utilizan más puntos porque no saben cómo hacerlo. Esto deja de fuera posibles soluciones que resultarán en mejor interés para la compañía. Por ejemplo ¿porqué expandir dos veces consecutivas una misma etapa? sería más económico colocar un equipo grande desde la primera expansión. Este tipo de resultados no es posible con su modelo.
  - Para hacer uso de toda la información de la demanda con que cuenta la empresa se propone un modelo multiperíodo de largo plazo.

- Al suponer un sólo producto con motivos de análisis, lo que en realidad se está calculando es una cota inferior del escalamiento necesario. Al desagrupar los productos, tal capacidad será siempre insuficiente.
  - El modelo propuesto es un modelo multi-producto. Cabe mencionar que a pesar de ser multi-producto, no se toman en cuenta todos los productos de la compañía. Se agruparon los productos en tres familias: productos de alta calidad para mercados pequeños, productos *light*, y todos los demás. La razón de las tres familias es la siguiente: los productos especializados son más caros de hacer, y también tienen un volumen de producción considerablemente menor a las demás familias; los productos *light* tienen en su receta diferencias significativas convirtiéndolos en una familia; y finalmente la tercera categoría, contienen a los productos realizados masivamente con un tipo de receta similar.
  
- Se suma el volumen de todos los equipos de la etapa para determinar una capacidad por etapa. Aquí intrínsecamente se está suponiendo que todos los equipos trabajan de forma paralela. El suponer que no se pueden desfazar equipos dentro de una etapa castiga mucho a la modelación, es un escenario de peor caso, con poca de flexibilidad y muy alejado de la realidad. Si bien el problema a abordar no es la calendarización de las actividades de cada equipo, hay simplificaciones menos drásticas.
  - El modelo propuesto permite que un equipo nuevo sea operado de forma paralela con otro equipo existente; o bien de manera desfasada.
  
- El utilizar un tiempo promedio de producción por producto por etapa la empresa, simplifica los cálculos y el entendimiento del problema; pero se pierde información de la receta, y no es posible determinar si la producción podría hacerse de una mejor forma.

- El utilizar la receta de cada producto, permitiría poder mejorar el proceso. Recuérdese que en el diseño de plantas batch lo que se optimiza es el ritmo de producción, que depende del tamaño de batch y su tiempo de ciclo.
- Por último, al decidir cuánto expandir el cuello de botella, el algoritmo heurístico no proporciona información alguna. Quizá exista un heurístico adicional para determinar el tamaño de la expansión.
  - Nuestro modelo concluye qué etapa expandir, cuándo hacerlo y de qué tamaño debe ser la expansión, de tal manera que se cumpla un objetivo de optimización, específicamente en nuestro caso minimizar la inversión total.

Para traducir el problema a un modelo matemático se empieza por agrupar etapas del proceso. La Figura 2.5 presenta el agrupamiento que seguimos para esta tesis; dicha estructura se mantiene en todos los ejemplos visitados.



**Figura 2.5** Etapas de modelo

**Etapa 1: Cocimiento.** Todo proceso hasta antes de los tanques de fermentación está en una etapa que llamamos *Cocimiento*, y es la primera etapa. Para los ejemplos específicos que abordamos, consideramos la etapa de *Cocimiento* como ilimitada.

Debido a los costos de instalación y escalamiento de estos procesos, se instala una casa de cocimientos, generalmente una de las principales inversiones en términos de costo. Posteriormente se van agregando tanques en diferentes períodos de tiempo, hasta que finalmente la capacidad de los tanques iguala la de la casa de cocimiento.

Podría incluirse el expandir esta primera etapa consiguiendo los parámetros adecuados de costos de instalación, y reduciendo su capacidad inicial, la cual es ilimitada para la escala de los ejemplos actuales; sin embargo, el verdadero problema planteado por la cervecería es precisamente el escalamiento de la tanquería hasta la capacidad de la casa de cocimientos.

En la etapa de Cocimiento, nos encontramos con las primeras dificultades de los procesos batch, dependiendo de la capacidad de la casa de cocimientos, se pueden necesitar de uno hasta doce cocimientos para llenar un fermentador.

**Etapa 2: Fermentación.** En la segunda etapa, están todos los tanques donde se realiza la fermentación. En la Figura 2.5 se puede ver cómo hay tres tipos de fermentadores: i) Los de escala 1:1 con la etapa siguiente, que vacían su producto a un único equipo de la etapa siguiente; ii) los de gran escala, que vacían su producto a una serie de equipos de la etapa siguiente y iii) los equipos que pueden llevar a cabo las dos etapas de proceso (fermentación y reposo) en un mismo tanque. La etapa de fermentación dura de siete a catorce días dependiendo del producto.



El poder manejar tres tipos de fermentadores, es una de las aportaciones de esta tesis a la literatura sobre el tema.

**Etapa 3: Reposo.** Una vez removida la levadura y finalizada la fermentación, se procede a la etapa de reposo. La etapa de reposo dura de tres a nueve días. En realidad, en la empresa se pueden usar tanques de fermentación para el reposo. Los tanques de reposo llevan menos equipamiento que los de fermentación, y por ende son más baratos, dichos tanques no pueden ser utilizados para la fermentación.

**Etapa 4: Filtración.** En la etapa de filtración se agrupó todo lo que ocurre después del reposo y antes del envasado, específicamente filtración y tanques de gobierno. La filtración es un proceso semi-continuo, y los tanques de gobierno actúan como buffer, para poder alimentar de manera continua el envasado. Así como en la etapa 1, se le ha considerado como una caja negra con capacidad disponible.

El problema se centra en la expansión de capacidad de las etapas de fermentación y reposo por medio de la adición de tanques a lo largo de un período de tiempo.

Además de agrupar procesos, también se agruparon productos en tres familias como se explicó anteriormente.

## **3. Modelo Multiperíodo de Diseño Retrofit para Plantas Batch**

### **3.1 Introducción**

Los modelos MINLP multiperíodo, para diseño ó incluso para cualquier otra aplicación, rápidamente se convierten en combinatoriamente explosivos cuando se requiere tratar a un problema realista. La literatura está repleta de algoritmos y modelos, pero rara vez utilizan un problema de tamaño considerable, como los que se presentan en la vida real. En adición a ello, no existe una forma general de problema. La mayoría de los casos son de problemas específicos y se resuelven con modelos específicos. Por eso el modelo de Van den Heever y Grossmann (1999), el cual presenta un modelo general de problemas multiperíodos, fue utilizado como base para desarrollarlo y se adaptó en este trabajo para que funcionase con el problema descrito en el Capítulo 1.

Partiendo del modelo general, este capítulo delinea el procedimiento y los pasos para transformarlo en un modelo de programación disyunta, y este a su vez, como puede ser representado por un MINLP. Se presenta el algoritmo del modelo de aproximación externa

disyunto y se explican las ventajas que tiene. Este algoritmo es una extensión del de un sólo período de Turkay y Grossmann (1996).

Posteriormente, se hacen las manipulaciones para adecuar el modelo general a nuestro problema, basado en el modelo de retrofit de plantas batch propuesto por Fletcher, Hall y Johns (1999). Se comenta a detalle el significado de cada ecuación, así como su manipulación para convexificar al modelo.

La efectividad del modelo propuesto se demuestra a través de varios ejemplos resueltos para horizontes de diferentes tamaños. El primer ejemplo es una versión simplificada donde solo se manejan dos productos. Con los resultados de este ejemplo, se ve la necesidad de ayudar al modelo para realizar sus operaciones de manera más efectiva. Se llevaron a cabo algunas estrategias conocidas para agilizar la solución del modelo.

Primero, colocar cotas superiores a variables continuas para ayudar a la convergencia. Explotando el hecho de que el método utiliza una superestructura, entonces sabemos que ciertas variables no pueden sobrepasar la superestructura de la red, esto permite acotar el tamaño y número de batches, dos variables importantes en el modelo. La segunda herramienta consiste en agregar variables de holgura a ecuaciones “difíciles” para el modelo, agilizando así su convergencia.

El segundo ejemplo ya hace uso de estas estrategias mejorando sus tiempos de solución y pudiendo encontrar respuestas para problemas de tamaño considerable.

### **3.2 Modelo General Multiperíodo de Van den Heever y Grossmann (1999)**

En dicho trabajo se inicia con una base establecida de una super-estructura de red de proceso, operando en  $T$  períodos de tiempo. Estos períodos pueden ser meses, estaciones, años, períodos de problemas estocásticos o múltiples escenarios operando en condiciones diferentes. Las demandas, costos, alimentaciones y parámetros técnicos varían de período a período. Los datos deterministas para esos parámetros son datos iniciales para el modelo. El

objetivo de este algoritmo es el de optimizar el proceso sobre todos los períodos de tiempo minimizando el costo o maximizando las utilidades sujeto a una serie de restricciones. Las restricciones incluyen los balances de materia, de calor, ecuaciones de desempeño de procesos, ecuaciones que ligan períodos, relaciones lógicas entre procesos y variables de decisión.

Además, se consideran tres decisiones importantes en este modelo de optimización:

- a) la selección de la estructura de la red,
- b) la operación o apagado de equipo en un período  $t$ , y
- c) la expansión de un proceso en un período.

La selección de la estructura de la red se realiza siendo una decisión en un tiempo involucrando costos de inversión, mientras las otras dos decisiones son válidas para cada período de tiempo, y pueden involucrar tanto costos fijos, como variables de operación y de expansión. Las restricciones del modelo general son tanto lineales como no lineales, mientras que las variables son discretas (de decisión) y continuas (flujos, tamaño de equipo, etc.). Por tanto, el modelo cae dentro de una categoría general para diseño, operación/expansión y planeación acorde para el caso específico que se estudia en esta tesis.

### 3.2.1 Modelo MINLP Multiperíodo General

El modelo considera optimizar una súper-estructura de red de proceso sobre los períodos  $t = \{1 \dots T\}$ . El modelo general para el diseño y operación/expansión es el siguiente. Primero definimos los conjuntos, parámetros y variables para la red del proceso:

*Conjuntos:*

- |     |                                |
|-----|--------------------------------|
| $I$ | conjunto de corrientes         |
| $J$ | conjunto de procesos           |
| $T$ | conjunto de períodos de tiempo |

*Índices:*

- $i$  corriente del conjunto  $I$   
 $j$  proceso en el conjunto  $J$   
 $t$  tiempo en el conjunto  $T$

*Parámetros:*

- $\alpha_{jt}$  costo de expansión variable para la unidad  $j$  en el período  $t$   
 $\beta_{jt}$  costo de expansión fijo para la unidad  $j$  en el período  $t$   
 $\gamma_{jt}$  costo de operación fijo para la unidad  $j$  en el período  $t$   
 $c_{it}$  costo asociado con la corriente  $i$  en el período  $t$   
 $u$  cota superior válida correspondiente para cada variable

*Variables:*

*Variables Binarias*

- $y_j$  selección de invertir (1) o no (0) en la unidad  $j$ , 0 de otro modo  
 $w_{jt}$  selección de operar (1) o no (0) la unidad  $j$  en el período  $t$   
 $z_{jt}$  selección de expandir (1) o no (0) la capacidad unidad  $j$  en período  $t$

*Variables continuas*

- $Q_{jt}$  capacidad de la unidad  $j$  en el período  $t$   
 $QE_{jt}$  expansión de capacidad de unidad  $j$  en el período  $t$   
 $X_t$  Valor de la variable de estado (flujo, temperatura, etc.) en el período  $t$   
 $X_{it}$  subconjunto de variables de estado para la corriente  $i$  en el período  $t$   
 $CO_{jt}$  costo de operación para la unidad  $j$  en el período  $t$   
 $CE_{jt}$  costo de expansión para unidad  $j$  en el período  $t$

Basado en las definiciones anteriores el modelo general es el siguiente:

$$(P) \quad \min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} X_{it} \quad (3.1)$$

suejto a

$$CO_{jt} = \gamma_{jt} w_{jt} \quad \forall j, t \quad (3.2)$$

$$CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} \quad \forall j, t \quad (3.3)$$

$$g_t(X_t, X_{t-1}) \leq 0 \quad \forall t \in T \setminus \{1\} \quad (3.4)$$

$$h_{jt}(Q_{jt}, X_t, X_{t-1}) \leq 0 \quad \forall j, t \in T \setminus \{1\} \quad (3.5)$$

$$Q_{jt} = Q_{j,t-1} + QE_{jt} \quad \forall j, t \quad (3.6)$$

$$X_t - U w_{jt} \leq 0 \quad \forall j, t \quad (3.7)$$

$$QE_{jt} - U z_{jt} \leq 0 \quad \forall j, t \quad (3.8)$$

$$y_j \leq \sum_t w_{jt} \quad \forall j \quad (3.9a)$$

$$w_{jt} \leq y_j \quad \forall j, t \quad (3.9b)$$

$$w_{jt} \leq \sum_{\tau=1}^t z_{j\tau} \quad \forall j, t \quad (3.9c)$$

$$z_{jt} \leq w_{jt} \quad \forall j, t \quad (3.9d)$$

$$CO_{jt}, CE_{jt}, Q_{jt}, QE_{jt}, X_t \geq 0 \quad (3.10)$$

$$y_j, w_{jt}, z_{jt} \in \{0,1\} \quad (3.11)$$

$$j \in J \quad i \in I \quad t \in T$$

La ecuación (3.1) es la función objetivo minimizando el costo total, mientras las ecuaciones (3.2) y (3.3) representan costo de operación y expansión, respectivamente. La ecuación (3.2) representa el costo de operación fijo, el costo de operación variable es representado por la suma de los costos de una corriente o flujo determinado,  $c_{it}X_{it}$ , en (3.1). Las restricciones globales validas para un período en particular, como balance de masa, son representadas en (3.4). Las restricciones representadas en (3.5) son válidas para una unidad dada  $j$  en un período en particular, por ejemplo relaciones de entrada-salida. Tanto (3.4) como (3.5) pueden, por generalidad, involucrar variables que pasan información de un período anterior,  $X_{t-1}$ , en cuyo caso dan lugar a variables de ligadura. Solo un subconjunto de ecuaciones pequeño involucra variables de ligadura, si es que éstas están presentes.

Otras variables de ligadura se encuentran en (3.6) y dictan que la capacidad en el presente período es igual a la capacidad del período anterior más la expansión de capacidad.

Las restricciones (3.7) establecen que todas las variables de estado asociadas con la unidad  $j$  son cero si ésta no es operada en el período  $t$ , mientras que las restricciones (3.8) fijan la expansión de capacidad de la unidad  $j$  a cero si es que no se expande en el período  $t$ . Las restricciones (3.9a)-(3.9d) representan relaciones lógicas entre las variables binarias. La restricción (3.9a) establece que una unidad es operada en por lo menos un período si es seleccionada, mientras (3.9b) establece que una unidad debe ser seleccionada si ha de ser operada en cualquier período. La restricción (3.9c) indica que una unidad solamente puede ser operada si ha sido expandida a una capacidad más grande que cero, y la restricción (3.9d) establece que una unidad solamente puede ser expandida en cierto período si es operada. En este caso se asume que la unidad  $j$  es expandida en el período  $t$  si necesita ser operada en ese período. Esta es una suposición válida, dado que no habría necesidad de expandir la unidad  $j$  si no va a ser utilizada. La construcción de una expansión tiene que ser terminada antes de que pueda utilizarse un equipo, esto se maneja descontando el costo de expansión de manera apropiada. La restricción (3.10) representa las variables que por su naturaleza deben ser positivas, y la (3.11) aquellas que son binarias.

En el caso de que no haya expansión ( $z_{jt}=0$ ), la ecuación (3.1) se convierte en  $\min \sum_t \sum_j CO_{jt} + \sum_t \sum_i c_{it} X_{it}$ , ya que el costo de expansión,  $CE_{jt}$  en la ecuación (3.3) sería cero.

En todas las formulaciones siguientes, los dominios  $j \in J$ ,  $i \in I$  y  $t \in T$  aplican, pero se han omitido por conveniencia.

Cabe aclarar que mientras las variables binarias  $y_j$  relacionadas a la selección de una unidad son independientes del período de tiempo, las variables binarias  $w_{jt}$  y  $z_{jt}$  por operación y expansión son definidas para cada período de tiempo. Por tanto, el número de estas variables aumenta conforme se incrementa el número de períodos.

### 3.2.2 Modelo Multiperíodo disyunto

El utilizar la programación disyunta reduce la dimensionalidad del problema y evita singularidades a causa de flujos con valor de cero, y elimina no-convexidades de los procesos inexistentes.

En la siguiente derivación se convierte al modelo (P) paso por paso a su forma disyunta. Primero se considera solamente la expansión y las ecuaciones y variables asociadas:

$$\left[ \begin{array}{c} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt}QE_{jt} + \beta_{jt} \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array} \right] \quad \forall j, t \quad (3.12)$$

La restricción (3.12) implica que si la unidad  $j$  es expandida en el período  $t$ , las ecuaciones (3.3) y (3.6) se aplican, de lo contrario la capacidad simplemente es igual a la capacidad del período anterior y ningún costo ocurre. Ahora consideremos el siguiente nivel de decisión, la planeación:

$$\left[ \begin{array}{c} w_{jt} \\ h_{jt}(Q_{jt}, x_t, x_{t-1}) \leq d \\ CO_{jt} = \gamma_{jt} \\ \text{ecuaciones de expansión (3.12)} \end{array} \right] \vee \left[ \begin{array}{c} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array} \right] \quad \forall j, t \quad (3.13)$$

La restricción (3.13) implica que si la unidad  $j$  es operada en el período  $t$ , las restricciones apropiadas (3.2) y (3.5) aplican, de lo contrario, un subconjunto de variables asociados con esa unidad y sus costos de operación se fijan a cero. Notar que la disyunción (3.12) esta dentro de la disyunción (3.13), tomando en cuenta que una expansión será considerada solamente si la unidad es operada en ese período de tiempo ( $w_{jt} = \text{verdadero}$ ). La construcción de la expansión es empezada algún tiempo antes de que pueda empezar a operar, y se maneja para descontárselo apropiadamente al costo de expansión.

Finalmente, se involucra el nivel de decisión de diseño. La restricción (3.13) implica que si se ha decidido instalar la unidad  $j$  ( $y_j = 1$ ), las ecuaciones pertinentes aplican, de lo



contrario, un subconjunto de variables asociadas a esa unidad se fijan a cero para todos los períodos y no se aplica costo alguno.

$$\left[ \begin{array}{c} y_j \\ \text{ecuaciones de operación} \\ CI_j = \beta_j + \alpha_j Q_j \end{array} \right] \quad (3.13) \quad \vee \quad \left[ \begin{array}{c} \neg y_j \\ B^j X_t = 0 \\ CI_j = 0 \\ \forall t \end{array} \right] \quad \forall j \quad (3.14)$$

Las disyunciones (3.12) y (3.13) están dentro de (3.14), viendo que un plan de operación y expansión solo es necesario si se ha decidido invertir en el proceso. Mas aún, si la expansión no ha de ser incluida en el problema, la disyunción (3.12) es simplemente remplazada por la función de costo de inversión. Sustituyendo las ecuaciones aplicables en (3.14), sumando las ecuaciones restantes en (P), y expresando las restricciones (2.9a)-(2.10) en su forma de lógica propocional (Raman y Grossmann, 1994) se obtiene la formulación disyunta final del modelo multiperíodo (P).

$$(PD): \quad \min \quad Z = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} X_{it}$$

sujeto a:

$$\left[ \begin{array}{c} y_j \\ \left[ \begin{array}{c} w_{jt} \\ h_{jt}(Q_{jt}, X_t, X_{t-1}) \leq d \\ CO_{jt} = \gamma_{jt} \end{array} \right] \\ \left[ \begin{array}{c} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array} \right] \\ \left[ \begin{array}{c} \neg w_{jt} \\ B^j X_t = 0 \\ CO_{jt} = 0 \end{array} \right] \vee \left[ \begin{array}{c} \neg y_j \\ B^j X_t = 0 \\ CI_j = 0 \\ \forall t \end{array} \right] \end{array} \right] \quad \forall j$$

$$\begin{aligned} y_j &\Rightarrow \bigvee_{t=1}^T w_{jt} \quad \forall j, t, & w_{jt} &\Rightarrow y_j \quad \forall j, t \\ w_{jt} &\Rightarrow \bigvee_{\tau=1}^t z_{j\tau} \quad \forall j, t, & z_{jt} &\Rightarrow w_{jt} \quad \forall j, t \end{aligned}$$

$$CE_{jt}, CO_{jt}, Q_{jt}, QE_{jt}, X_t \geq 0 \quad y_j, w_{jt}, z_{jt} \in \{0,1\}$$

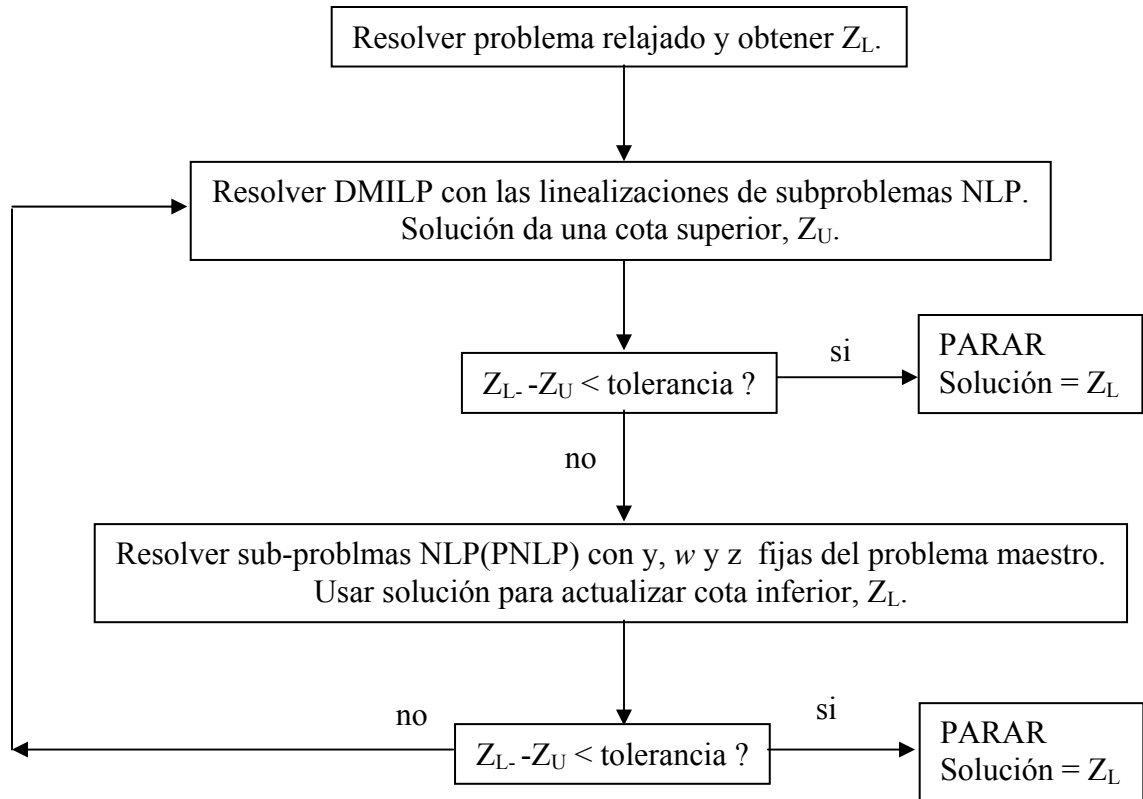
El modelo (PD) es una representación general incorporando tres niveles de decisión: diseño, planeación de la operación y planeación de la expansión. Estas decisiones son representadas por las variables discretas,  $y_j$ ,  $w_{jt}$  y  $z_{jt}$ , respectivamente. La formulación fácilmente se puede convertir a modelos más específicos ignorando los términos y variables apropiados. Esto nos lleva a poder utilizar este modelo y aplicarlo al problema de estudio; o de manera general, al retrofit de plantas batch.

### 3.2.3 Algoritmo de Aproximación Externa (OA) Disyunto

En el algoritmo disyunto de aproximación externa propuesto primero por Turkay y Grossmann (1996a), un problema de cobertura inicial se resuelve para determinar el menor número posible de configuraciones,  $N$ , para cubrir todas las unidades. Estas  $N$  configuraciones son usadas para resolver  $N$  sub-problemas NLP para generar linealizaciones iniciales para todas las restricciones.

Alternativamente, el problema relajado del MINLP puede ser utilizado como punto de partida si no aplica el problema de cobertura; éste es nuestro caso, dado que la configuración inicial de la planta es dada. Los sub-problemas NLP solamente incluyen ecuaciones para equipos existentes (por ejemplo ecuaciones de las disyunciones con valor “verdadero”).

El algoritmo que se presenta en la Figura 3.1 es una extensión a problemas multiperíodo del método de Turkay y Grossmann, que se restringía a un solo período. Al aplicar este algoritmo se descompone el modelo (PD) en un sub-problema NLP y un problema maestro MILP, como se presenta posteriormente. Ambos problemas son convertidos a forma entero-mixta por formulación del envolvente convexo (Balas, 1985, Turkay y Grossmann, 1996a).



**Figura 3.1:** Algoritmo Disyunto de Aproximación Externa

$Z_L$  es la solución del problema relajado, por lo que es una cota inferior de la solución óptima del problema general.  $Z_U$  es la solución de un problema agrandado, por lo que es una cota superior. Este algoritmo trata de igualar las dos cotas, o acercarlas dentro de una tolerancia aceptable.

La ventaja de esta formulación es que se reduce la dimensionalidad del problema al solamente considerar disyunciones con valor “verdadero” (ver modelo PNLP abajo), evita singularidades a causa de flujos con valor de cero, y elimina no-convexidades de los procesos inexistentes.

El sub-problema NLP es como sigue:

$$(PNLP) \quad \min \quad Z_U = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} X_{it}$$

sujeto a

$$\begin{array}{l}
 g_t(X_t, X_{t-1}) \leq 0 \\
 \left. \begin{array}{l} Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array} \right\} z_{jt} = \text{verdadero} \\
 \left. \begin{array}{l} Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array} \right\} z_{jt} = \text{falso} \\
 \left. \begin{array}{l} h_{jt}(Q_{jt}, X_t, X_{t-1}) \leq d \\ CO_{jt} = \gamma_{jt} \end{array} \right\} w_{jt} = \text{verdadero} \\
 \left. \begin{array}{l} B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array} \right\} w_{jt} = \text{falso} \\
 \left. \begin{array}{l} B^{jt} X_t = 0 \end{array} \right\} y_j = \text{falso} \\
 CE_{jt}, CO_{jt}, Q_{jt}, QE_{jt}, X_t \geq 0, \quad y_j, w_{jt}, z_{jt} \in \{0, 1\}
 \end{array}
 \left. \right\} y_j = \text{verdadero}$$

Para formular el problema maestro MILP la variable  $X_t$  es particionada en dos conjuntos de variables,  $X_t^k$ , cuyos valores son de cero si la expresión booleana asociada con esa disyunción es “falsa”, y  $X_t^{k'}$ , que puede tomar cualquier valor positivo independientemente del valor de la variable booleana. Además, las ecuaciones  $h_{jt}(Q_{jt}, X_t, X_{t-1})$  y  $g_t(X_t, X_{t-1})$  son particionadas en un conjunto de ecuaciones no lineales,  $h_{jt}^{nl}(X_t) \leq 0$  y  $g_t^{nl}(X_t) \leq 0$ , y un conjunto de ecuaciones lineales  $h_{jt}^l(Q_{jt}, X_t, X_{t-1}) = AX_t^{k'} + BX_t^k + CQ_{jt} \leq b$  y  $g_t^l(X_t) = DX_t \leq d$ .

Dadas  $L$  iteraciones mayores, el conjunto linealizado de términos activos de las disyunciones para operar es definido como  $K^{jt} = \{k \mid w_{jt} = \text{verdadero}, k = 1, \dots, L \text{ y } t = 1, \dots, T\}$ .

La formulación del problema maestro disyunto se muestra a continuación:

$$(DLP) \quad \min \quad Z_L = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} X_{it}$$

sujeto a

$$g_t^{nl}(X_t^k) + \nabla_{x_t} g_t^{nl}(X_t^k)^T (X_t - X_t^k) \leq 0 \quad \forall t$$

$$DX_t \leq d \quad \forall t$$

$$\left[ \begin{array}{c} y_j \\ \\ \\ w_{jt} \\ h_{jt}^{nl}(X_t^k) + \nabla_{x_{t,s'}} h_{jt}^{nl}(X_t^k)^T (X_{t,s'} - X_{t,s'}^k) + \\ \nabla_{x_{t,s}} h_{jt}^{nl}(X_t^k)^T (X_{t,s} - X_{t,s}^k) \leq a \\ AX_{t,s'} + BX_{t,s} + CQ_{jt} \leq b \\ k \in K^j \\ CO_{jt} = \gamma_{jt} \\ \\ z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array} \right] \vee \left[ \begin{array}{c} -w_{jt} \\ B^{jt} X_t = 0 \\ CO_{jt} = 0 \end{array} \right] \forall t \vee \left[ \begin{array}{c} -y_j \\ B^{jt} X_t = 0 \\ \forall t \end{array} \right] \forall j$$

$$y_j \Rightarrow \bigvee_{t=1}^T w_{jt} \quad \forall j, t, \quad w_{jt} \Rightarrow y_j \quad \forall j, t$$

$$w_{jt} \Rightarrow \bigvee_{\tau=1}^t z_{j\tau} \quad \forall j, t, \quad z_{jt} \Rightarrow w_{jt} \quad \forall j, t$$

$$\Omega(y) = \{y_j \mid y_j = 1\}$$

$$CE_{jt}, CO_{jt}, Q_{jt}, QE_{jt}, X_t \geq 0, \quad y_j, w_{jt}, z_{jt} \in \{0, 1\}$$

Después de aplicar el envolvente convexo a cada disyunción, la formulación final es obtenida, DMILP (ver Apéndice A para derivación completa).

$$(DMINLP) \quad \min \quad Z_L = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} X_{it}$$

sujeto a

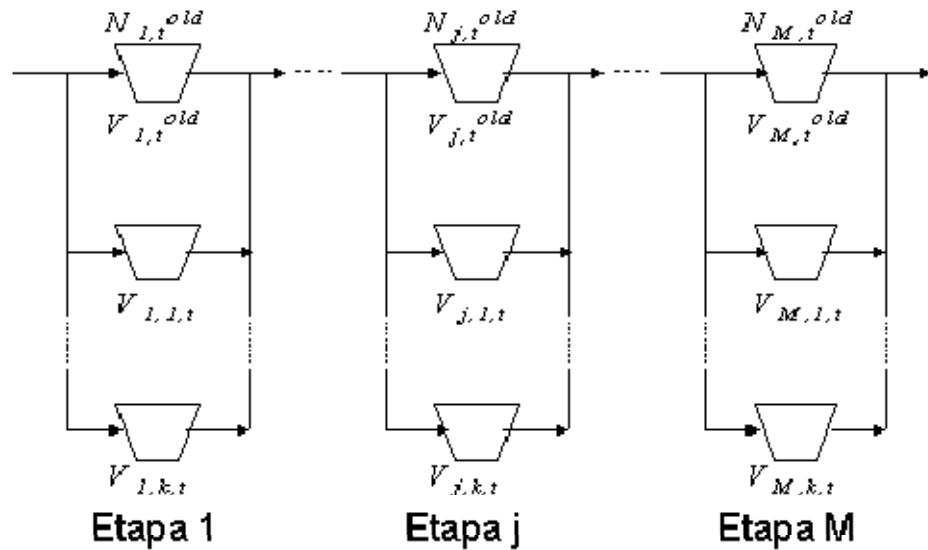
$$\begin{aligned}
 \nabla_{x_t} g_t^{nl}(x_t^k)^T x_t &\leq -g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T x_t^k && \forall t \\
 Dx_t &\leq d && \forall t \\
 \nabla_{x_{t,s}} h_{jt}^{nl}(x_t^k)^T x_{t,s}^1 + \nabla_{x_s} h_{jt}^{nl}(x_t^k)^T x_{t,s} &\leq \\
 -\left[ h_{jt}^{nl}(x_t^k) - \nabla_{x_{t,s}} h_{jt}^{nl}(x_t^k)^T x_{t,s}^k - \nabla_{x_s} h_{jt}^{nl}(x_t^k)^T x_{t,s}^k - a \right] w_{jt} &&& \forall j, t \\
 &&& k \in K^j \\
 Ax_{t,s}^1 + Bx_{t,s} + CQ_{jt}^1 &\leq bw_{jt} && \forall j, t \\
 x_{t,s} &= x_{t,s}^1 + x_{t,s}^2 && \forall t \\
 x_{t,s}^1 &\leq U w_{jt} && \forall j, t \\
 x_{t,s}^2 &\leq U(1 - w_{jt}) && \forall j, t \\
 x_{t,s} &\leq U w_{jt} && \forall j, t \\
 Q_{jt} &= Q_{jt}^1 + Q_{jt}^2 && \forall j, t \\
 Q_{jt}^1 &\leq U w_{jt} && \forall j, t \\
 Q_{jt}^2 &\leq U(1 - w_{jt}) && \forall j, t \\
 CE_{jt} &= \alpha_{jt} Q_{jt} + \beta_{jt} z_{jt} && \forall j, t \\
 CO_{jt} &= \gamma_{jt} w_{jt} && \forall j, t \\
 Q_{jt} &= Q_{j,t-1} + QE_{jt} && \forall j, t \\
 QE_{jt} - Uz_{jt} &\leq 0 && \forall j, t \\
 y_j &\leq \sum_{t=1}^T w_{jt} && \forall j \\
 w_{jt} &\leq y_j && \forall j, t \\
 w_{jt} &\leq \sum_{\tau=1}^t z_{j\tau} && \forall j, t \\
 z_{jt} &\leq w_{jt} && \forall j, t \\
 CE_{jt}, CO_{jt}, Q_{jt}, QE_{jt}, X_t &\geq 0, && y_j, x_{t,s}, z_{jt} \in \{0,1\}
 \end{aligned}$$

Sobre el modelo podemos decir que la convergencia a la solución óptima es garantizada en un número finito de iteraciones si el modelo es convexo (Turkay y Grossmann, 1996). Como se discutirá más adelante, también se puede hacer uso de prioridades de ramificar en la búsqueda y de ramificar y acotar en el problema maestro, explotando la estructura jerárquica ramificando primero en  $y_j$ , después en  $w_{jt}$  y finalmente en  $z_{jt}$ . En el modelo DLP se puede visualizar este punto. Si  $y_j$  es cero, entonces el problema se simplificará dado que la unidad  $j$  no será tomada en cuenta, y las restricciones asociadas a

ella serán solamente ecuaciones de paso. Por añadidura, si  $y_j$  es cero,  $w_{jt}$  y  $z_{jt}$ , también serán cero puesto que no se puede operar ni expandir una unidad que no sea tomada en cuenta para la planta.

### 3.3 Aplicación al Problema de Interés

Considérese una planta batch multi-producto para manufacturar  $N$  productos y consistiendo de  $M$  etapas en secuencia con equipo paralelo en cada etapa. Ver Figura 3.2, dónde  $V_{jkt}$  es el volumen la unidad  $k$  en la etapa  $j$  para algún período dado  $t$ .



**Figura 3.2:** Estructura para diseño retrofit de plantas batch multiproducto (Vaselenak, Grossmann y Westerberg, 1987)

En el diseño retrofit de estas plantas batch, el problema cae en decidir la adición de equipo nuevo a una planta existente para la cual se han especificado nuevas metas de producción y no pueden ser cumplidos con la configuración de la planta existente (Fletcher, Hall y Johns, 1991).

Las unidades nuevas pueden ser añadidas a operar en fase con las unidades existentes de tal manera que incrementen su capacidad (opción A), o en secuencia con unidades existentes de manera que se reduzca el tiempo de ciclo (opción B).

Para el problema de retrofit multiperíodo, pueden ser añadidas nuevas unidades en cada período de tiempo, dependiendo de las fluctuaciones del mercado pronosticadas sobre un horizonte de planeación. Debe determinarse para cada período de tiempo una configuración de planta revisada, con estrategia de operación, tamaño de equipos y parámetros de proceso batch para los cuales el beneficio es óptimo.

La formulación MINLP de este problema para un solo período de tiempo es dada por Fletcher, Hall y Johns (1991), quien, en contraste con Vaselenak, Grossmann y Westerberg (1987), permite a las unidades operar en diferentes modos para diferentes productos.

Las siguientes suposiciones son necesarias para el funcionamiento de este modelo:

- se produce bajo campañas de un mismo producto (Single Product Campaigns),
- no hay almacenamiento entre etapas,
- no se puede partir o mezclar batches,
- no hay tiempos de transporte (llenado y vaciado espontáneos),
- el tiempo de procesamiento  $t_{ij}$  del producto  $i$  en la etapa  $j$  es independiente del tamaño del batch,
- los factores de tamaño  $S_{ij}$  son constantes,
- existen recursos ilimitados.

Con respecto al primer punto, sobre producir bajo campañas de un mismo producto, se tratará de reducir su impacto con el método de corrección de planeación, que posteriormente se plantea en el Capítulo 4.



El no haber disponibilidad de almacenamiento entre etapas y el no poder partir batches, son factores reales de la producción de cerveza, por lo que no afecta el supuesto al resultado.

Los tiempos de procesamiento y transporte son tomados en cuenta en las recetas de cada producto, y en la planta en efecto son independientes del tamaño del batch.

Utilizar los factores de tamaño,  $S_{ij}$ , para caracterizar y modelar un sistema batch ha sido adoptado desde su primera publicación (Sargent y Grossmann, 1979) por todas las referencias encontradas.

Por último se suponen recursos ilimitados puesto que se asume que la compañía quiere cumplir con la demanda siempre.

Recordar como se explicó en el Capítulo 1, en el diseño de plantas batch, la idea general es optimizar el ritmo de producción de cada producto en la planta. El ritmo de producción es función del tamaño del batch y del tiempo de ciclo, y cada producto tiene su propio ritmo de producción.

En esta tesis se ajusta el modelo para incluir diferentes tipos de tecnología en dos etapas de la planta. Se usa la misma nomenclatura que Fletcher, Hall y Johns (1991), con la diferencia que algunos parámetros y variables son indexados en el tiempo debido a la formulación multiperiodica. Nótese como las expansiones ocurren solamente una vez en todo el horizonte de planeación, y son el equivalente a instalar una unidad nueva. También se utiliza aquí una formulación convexificada de las restricciones. Además, el modelo disyunto tiene una estructura similar al problema (PD), con la excepción de que disyunciones adicionales son necesarias para modelar los dos modelos de operación de las unidades (opción B u opción C).

Nomenclatura:

*Conjuntos:*

- $I$  conjunto de productos
- $J$  conjunto de etapas
- $K$  conjunto de unidades por etapa
- $T$  conjunto de períodos de tiempo

*Índices:*

- $i$  producto del conjunto  $I$
- $j$  etapa del conjunto  $J$
- $k$  unidad  $k$  del conjunto  $K$  de unidades añadidas
- $t$  tiempo en el conjunto  $T$
- $m$  conjunto de unidades existentes en la planta

*Parámetros:*

- $M$  número de etapas en la planta,  $*J* = M$
- $N_j^{old}$  número de unidades que existen en la etapa  $j$
- $V_{jm}^{old}$  volumen de la unidad  $m$  en etapa  $j$
- $T_{ij}$  tiempo de proceso del producto  $i$  en etapa  $j$
- $H_t$  horizonte de tiempo para cumplir demanda en el período  $t$
- $S_{ij}$  factor de tamaño que requiere el producto  $i$  de la etapa  $j$
- $K_{jt}$  costo fijo anualizado de instalar una unidad nueva en etapa  $j$  en período  $t$
- $C_{jt}$  coeficiente de costo anualizado, por unidad de volumen de unidad nueva instalada en etapa  $j$  en período  $t$
- $Q_{it}$  demanda requerida del producto  $i$  en período  $t$
- $V_j^L$  volumen mínimo de unidades nuevas en etapa  $j$

- $V_j^U$  volumen máximo de unidades nuevas en etapa  $j$
- $Z_j$  máximo numero de unidades que se pueden añadir a etapa  $j$
- $Z^U$  máximo numero de unidades que se pueden añadir a la planta

*Variables Binarias:*

- $y_{jk}$  1 si se selecciona invertir en la unidad  $k$  en la etapa  $j$ , 0 de otra forma
- $w_{jkt}$  1 si se opera una unidad  $k$  en etapa  $j$  en el período  $t$ , 0 de otra forma
- $w_{ijkmt}^B$  1 si se opera una nueva unidad  $k$  sincronizada con la unidad existente  $m$  para el producto  $i$  en la etapa  $j$  en período  $t$ , 0 de otra forma
- $w_{ijkt}^C$  1 si se opera una nueva unidad  $k$  en secuencia con unidad existente para el producto  $i$  en etapa  $j$  en período  $t$ , 0 de otra forma
- $z_{jkt}$  1 si se instala una unidad nueva  $k$  en la etapa  $j$  para período  $t$ , 0 de otra forma

*Variables Continuas:*

- $N_{it}$  numero de batches del producto  $i$  en el período  $t$
- $B_{it}$  Tamaño del batch del producto  $i$  en el período  $t$
- $T_{Lit}$  Tiempo de ciclo del producto  $i$  en el período  $t$
- $V_{jkt}$  volumen de unidad nueva  $k$  en etapa  $j$  en el período  $t$
- $E_{jkt}$  volumen de expansión de unidad nueva  $k$  en etapa  $j$  en el período  $t$
- $V_{ijkmt}^B$  volumen requerido en unidad nueva  $k$  en etapa  $j$  para producto  $i$  para usarse sincronizada con unidad existente  $m$  en período  $t$
- $V_{ijkt}^C$  volumen requerido de unidad nueva  $k$  en etapa  $j$  para producto  $i$  a usarse en secuencia con unidades existentes en período  $t$
- $CE_{jkt}$  costo de expansión/instalación para unidad nueva  $k$  en etapa  $j$  en período  $t$

$CA_{tk}$  costo de ajuste por escoger unidad nueva  $k$  en período  $t$

Se definen las siguientes variables para aplicar transformación exponencial y convexificar el modelo (Vaselenak, Grossmann y Westerberg, 1987, ver Apéndice B para derivación):  $x^{(1)}_{it} = \ln N_{it}$ ,  $x^{(2)}_{it} = \ln B_{it}$ ,  $x^{(3)}_{it} = \ln T_{Lit}$ . La formulación multiperíodo, obtenida aplicando el modelo general de programación disyunto (PD) se presenta a continuación. A este modelo lo llamamos PDR (Programa Disyunto Retrofit)

i) Función Objetivo:

$$\min \sum_t \sum_j \sum_k CE_{tjk} - \sum_t \sum_k CA_{tk}$$

ii) Meta de producción:

$$N_{it} B_{it} \geq Q_{it} \quad \forall i, t \text{ con la transformación queda,}$$

$$x^{(1)}_{it} + x^{(2)}_{it} \geq \ln Q_{it} \quad \forall i, t$$

iii) Tiempo de ciclo limitante del producto  $i$ :

$$\frac{T_{ij}}{T_{Lit}} - \sum_k w_{ijkt}^C \leq N_j^{old} \quad \forall i, j, t ; \text{ después de la transformación queda,}$$

$$N_j^{old} + \sum_k w_{ijkt}^C \geq T_{ij} \exp(-x^{(3)}_{it}) \quad \forall i, j, t$$

iv) Tiempo de producción en el período  $t$  anual:

$$\sum_i N_{it} T_{Lit} \leq H_t \quad \forall t ; \text{ ya transformado}$$

$$\sum_i \exp(x^{(1)}_{it} + x^{(3)}_{it}) \leq H_t \quad \forall t$$

v) Cota en el numero total de unidades nuevas:

$$\sum_j \sum_k y_{jk} \leq Z^U$$

vi) Restricciones de capacidad opción B:

$$\sum_k V_{ijkmt}^B + (V_j^{old})_m \geq S_{ij} B_{it} \quad \forall i, j, m, t$$

vii) Asignación ordenada de unidades nuevas:

$$y_{jk} \geq y_{j,k+1} \quad \forall j, k = 1 \dots Z_j - 1$$

viii) Costo de ajuste por escoger una expansión del mismo volumen en dos etapas juntándolas en un equipo

$$CA_{kt} = c_{2t} \quad \forall \{k, t \mid E_{2kt} = E_{3kt}\}$$

ix) Disyunción de cada unidad  $k$  añadida a la etapa  $j$ :

$$\left[ \begin{array}{c} y_{jk} \\ V_{jkt} \leq V_j^U \\ V_{jkt} = V_{jk,t-1} + E_{jkt} \\ w_{jkt} \\ V_{jkt} \geq V_j^L \\ \left[ \begin{array}{c} w_{ijk1t}^B \\ V_{ijk1t}^B \leq V_{jkt} \\ V_{ijk1t}^B \leq V_j^U \end{array} \right] \vee \dots \vee \left[ \begin{array}{c} w_{ijkmt}^B \\ V_{ijkmt}^B \leq V_{jkt} \\ V_{ijkmt}^B \leq V_j^U \end{array} \right] \vee \left[ \begin{array}{c} w_{ijk}^C \\ V_{ijk}^C \leq V_{jkt} \\ V_{ijk}^C \leq V_j^U \\ V_{ijk}^C \geq B_{it} S_{ij} \end{array} \right] \forall i \\ \left[ \begin{array}{c} z_{jkt} \\ CE_{jkt} = K_{jt} + c_{jt} E_{jkt} \\ V_j^L \leq E_{jkt} \leq V_j^U \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jkt} \\ CE_{jkt} = 0 \\ E_{jkt} = 0 \end{array} \right] \\ \vee \left[ \begin{array}{c} \neg w_{jkt} \\ V_{jkt} \geq 0 \end{array} \right] \forall t \\ \vee \left[ \begin{array}{c} \neg y_{jk} \\ V_{jkt} = 0 \end{array} \right] \forall j, k \end{array} \right]$$

x) Ecuaciones Lógicas:

$$\sum_m w_{ijkmt}^B + w_{ijkmt}^C = w_{jkt} \quad \forall i, j, k, t \quad \text{operar en un solo modo}$$

$$\sum_t z_{jkt} = y_{jk} \quad \forall j, k \quad \text{si se escoge una unidad es porque se ha expandido}$$

$$y_{jk} \geq \sum_t w_{jkt} \quad \forall j, k \quad \text{unidad se expande solamente una vez}$$

$$w_{jkt} \leq y_{jk} \quad \forall j, k, t \quad \text{si se escoge una unidad debe operar}$$

$$w_{jkt} \leq \sum_{\tau=1}^t z_{jk\tau} \quad \forall j, k, t$$

$$z_{jkt} \leq w_{jkt} \quad \forall j, k, t$$

xi) Variables:

$$N_{it}, B_{it}, T_{Lit}, V_{jk}, E_{jkt}, V_{jkt}, V_{ijkmt}^B, V_{ijkmt}^C \geq 0 \quad y_{jk}, w_{jkt}, w_{ijkmt}^B, w_{ijkmt}^C, z_{jkt} = \{0,1\}$$

$$i \in I, \quad j \in J, \quad t \in T, \quad k \in Z, \quad m \in N_j^{old}$$

### 3.4 Casos de Estudio

Todos los modelos de esta tesis son resueltos en el paquete comercial GAMS (Brook, Kendrick y Meeraus, 1992) versión 22.5, utilizando DICOPT (GAMS Development Corporation, 2008) con Cplex (GAMS Development Corporation, 2008) como motor de solución para los programas lineales entero-mixtos, y CONOPT (GAMS Development Corporation, 2008) para los programas no lineales. GAMS se ejecuta en una máquina Dell DXP051, procesador 3192 Mhz y 2GB de RAM.

Los datos de los parámetros de los ejemplos se encuentran en los apéndices (apéndice C), ellos incluyen la configuración inicial de la planta, junto con el volumen inicial de cada tanque existente, los tiempos de proceso y la demanda de cada producto, las horas de producción al año, así como los costos fijos y variables de instalar un equipo nuevo.

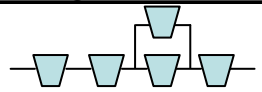
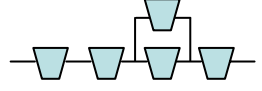
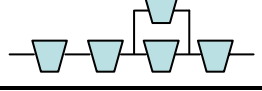
Se resuelven tres casos, ver Tabla 3.1, todos tienen cuatro etapas de proceso y la misma configuración de planta inicial. Los primeros dos casos manejan además la misma demanda para el número de períodos dado, y los mismos costos de expansión.

El Caso 1 maneja dos productos y se resuelve para tres horizontes diferentes. Es en realidad un ejemplo pequeño, con el cual poder empezar a hacer ajustes al modelo para instancias más grandes, que son las de interés.

El Caso 2 tiene tres productos y se resuelve para cuatro diferentes horizontes de tiempo, tres productos es lo que se maneja para el caso industrial.

El Caso 3, se realizó para demostrar la posibilidad de adicionar un equipo en el que se puedan llevar acabo dos etapas de proceso, la de fermentación y reposo. Se aumentó la demanda de manera brusca, de modo que la capacidad instalada es notablemente insuficiente, además se elevaron los costos de instalación, a manera de forzar al modelo a dar el resultado buscado.

**Tabla 3.1:** Casos de estudio Capítulo 3.

	configuración inicial	número de etapas	número de productos	número de periodos
Caso 1		4	2	4 8 10
Caso 2		4	3	5 8 10
Caso 3		4	3	3

Al intentar resolver para tres familias de productos, se encontró que el método no pudo resolver el problema, agotando algún recurso ya sea de iteraciones, de tiempo, de iteraciones o de memoria.

Se utilizaron tres estrategias para tratar de eficientizar al modelo, de manera que pudiese resolver instancias más grandes. Primero se le adicionan variables de holgura  $P_1$  y  $P_2$  a las ecuaciones ii) y iv) respectivamente, del modelo PDR:

$$ii) x_{it}^{(1)} + x_{it}^{(2)} \geq \ln Q_{it} - P_1 \quad \forall i, t$$

$$\text{iv) } \sum_i \exp(x_i^{(1)} + x_i^{(3)}) \leq H_t + P_2 \quad \forall t$$

con su respectiva corrección en la función objetivo:

$$\min \sum_t \sum_j \sum_k CE_{tjk} + M(P_1 + P_2) \quad \text{donde } M \text{ es un número muy grande y } P_1, P_2 \geq 0.$$

En la ecuación ii) lo que hace es que se le permite al modelo tener una demanda inferior a la real,  $Q_{it}$ . En la ecuación iv) se le permite tener más tiempo de producción por año,  $H_t + P_2$ . Esto permite al motor de solución poder agilizar su búsqueda ya que resulta más sencillo eliminar una respuesta como *mala* (si  $P_1$  y  $P_2$  tomaran valor alguno diferente de cero), que desecharla como no factible, que sería lo que tendría que hacer sin las variables de holgura.

En segundo lugar, aprovechando que el modelo maneja una super-estructura de red y además los volúmenes de los equipos que se pueden añadir en cada etapa, tienen una cota tanto superior como inferior. Por tanto, si no se agrega equipo alguno, utilizando los tiempos de proceso del producto  $i$  en etapa  $j$ ,  $T_{ij}$ , se puede calcular el tiempo de ciclo máximo posible.

De la misma manera, al agregar todos los equipos posibles a la red, con los  $t_{ij}$  se puede calcular el tiempo de ciclo mínimo posible para cada producto. En forma similar, con el tiempo de proceso del producto  $i$  en etapa  $j$   $T_{ij}$ , el horizonte de diseño,  $H_t$ , y la demanda  $Q_{it}$ , se puede calcular el número mínimo de batches necesarios. Asimismo si en lugar de utilizar la demanda  $Q_{it}$ , se utiliza el número máximo de unidades a añadir y el volumen máximo permitido, podemos obtener una cota inferior para el número de batches fortaleciendo al modelo.

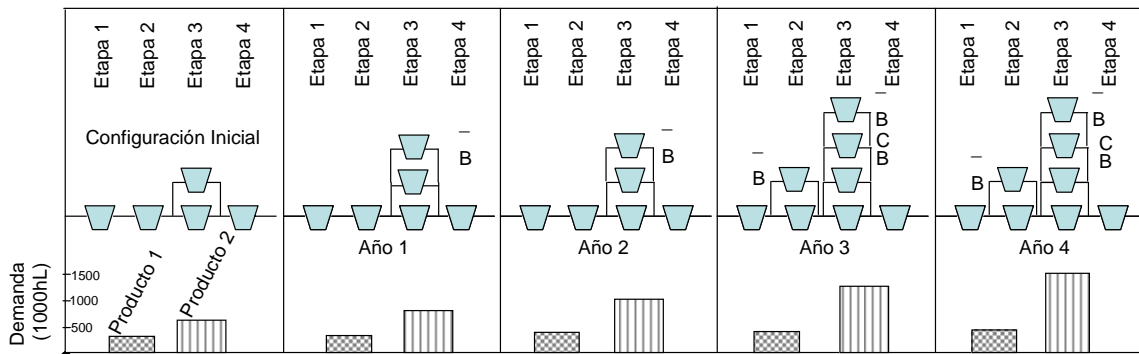
Adicionalmente, se habilitó al motor de solución la opción de hacer uso de prioridades de ramificar en la búsqueda, y de ramificar y acotar en el problema maestro,



explotando la estructura jerárquica ramificando primero en  $y_j$ , después en  $w_{jt}$  y finalmente en  $z_{jt}$ ; observar estructura en ecuación ix) del modelo (PDR).

La implementación de estas estrategias ayuda al modelo de manera considerable. Por ejemplo, antes de hacer las modificaciones, no se obtenían resultados para más de ocho períodos. En el caso específico del Caso 2, el modelo de cinco períodos, el tiempo de cómputo para encontrar la solución óptima disminuyó de 84,790 segundos a 130 segundos. Esto es cercano a un día, reducido a tan sólo a dos minutos. Ambos modelos encontraron un valor de \$535 de inversión.

El primer caso para dos productos con cuatro periodos de tiempo se muestra en la Figura 3.3, donde se ve que una unidad nueva es añadida en la etapa tres en el año uno y también en la etapa tres en el año tres, dando como resultado un costo de inversión de \$281.8, expandiendo un total de tres unidades; una unidad de la etapa 2 y dos unidades de la etapa 3.



**Figura 3.3:** Solución al Caso 1 para cuatro períodos.

Este es un ejemplo simplificado, donde se manejan solo dos productos a manera de ilustrar el modelo. Se resuelve esta instancia para 4, 8 y 10 períodos de tiempo. Los resultados para el Caso 1 se encuentran en la Tabla 3.2

**Tabla 3.2:** Resultados para el Caso 1.

Número de Períodos de tiempo	Variables Discretas	Variables Continuas	Restricciones	Tiempo de solución (seg. CPU ) OA disyunto	Costo de Inversión (\$1000)
4	396	939	1354	4	281.80
8	780	1,859	2,670	79	408.84
10	1,620	3,807	5,392	745	580.51

Con estos cambios ya fue posible encontrar resultados para modelos de hasta 10 períodos. En las tablas de resultados, Tabla 3.2 y 3.3., los renglones son las diferentes instancias resuletas, en este caso diferente número de períodos. Acomodados por columna encontramos el tamaño del problema, descrito por su número de variables discretas, continuas y de restricciones; y finalmente los resultados, una columna del tiempo que tardó en encontrar la solución óptima y el valor objetivo de la misma en la última columna.

Para el Caso 2 de tres familias de productos, con la misma configuración inicial que en el caso anterior se puede observar el comportamiento explosivo del modelo.

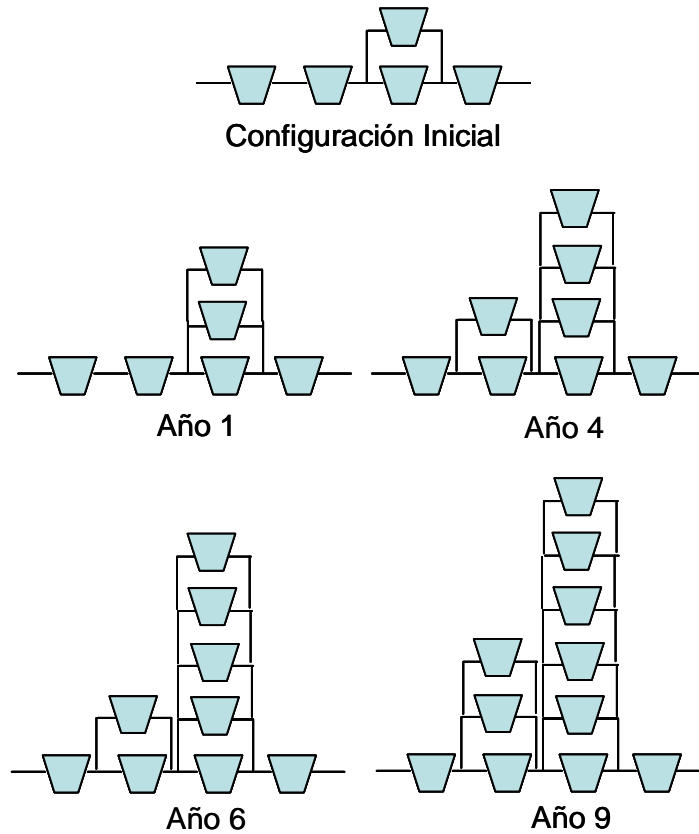
En la Figura 3.3 se ve como aumentan los tiempos de solución en un orden de magnitud al cambiar de cinco a ocho períodos de tiempo. Lo mismo pasa al cambiar de ocho a diez períodos de tiempo. Para diez períodos de tiempo el modelo ya está tomando 41,000 segundos, esto es 11.4 horas. Recordando que nuestro modelo es de diseño con un horizonte de planeación de veinte o más años, esto no es aceptable.

**Tabla 3.3:** Resultados para el Caso 2.

Número de Períodos de tiempo	Variables Discretas	Variables Continuas	Número Restricciones	Tiempo de solución (seg. CPU) OA Disyunto	Costo de Inversión (\$1000)
5	2240	5253	7038	130	535.80
8	3,560	8379	11,190	4019	752.52
10	4,440	10,463	13,958	41226	950.36
20	8840	20,883	27,798	***	---

\*\*\* No encontró solución después de 20 iteraciones ó 5 días (432,000 seg.)

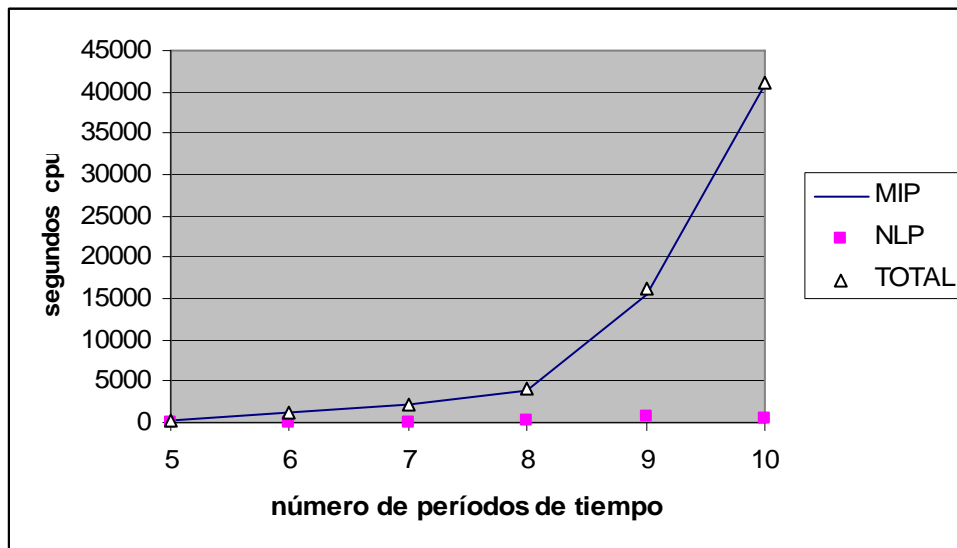
En la Figura 3.4 se ve que existe una tendencia en las expansiones, manteniendo una relación de la Etapa 2 (fermentadores) con la Etapa 3 (tanques de reposo) entre  $\frac{1}{3}$  y  $\frac{1}{2}$ .



**Figura 3.4.** Expansiones de planta para ejemplo 2 con 10 períodos

Los tiempos calculados para el problema de 20 períodos no cumplen con la expectativa. Esto motiva el desarrollo de un algoritmo que ayude a mejorar sus tiempos de cómputo, antes de poder presentar el modelo; para ello se explora una adecuación práctica en el siguiente capítulo.

El tiempo total de solución muestra las mismas tendencias que los tiempos de solución de los MILPs. Los tiempos de solución de los NLPs son insignificantes en comparación con los de los MILPs del problema maestro. El beneficio de utilizar la formulación disyunta es insuficiente ya que para problemas grandes no siempre se encuentra resultado. Ver tendencias en la Figura 3.5 para el Caso 2.



**Figura 3.5:** Tiempo de cómputo (seg. CPU) para algoritmo disyunto

Nótese como el tiempo del algoritmo esta prácticamente asociado al tiempo que toma resolver los problemas maestros, motivo por el cual escogimos el algoritmo disyunto, a pesar de que parece ser insuficiente. Para poder hacer una comparación efectiva, y demostrar las

ventajas de la programación disjunta, modelamos al problema de la manera tradicional. La Tabla 3.3 presenta los datos del problema resuelto bajo diferentes circunstancias.

**Tabla 3.4.** Problema resuelto por diferentes modelos.

Modelo	Variables Discretas	Variables Continuas	Número de Restricciones	Tiempo de solución (seg. CPU)	Costo de Inversión (\$1000)
Modelo disyunto no-convexo	2,220	5,267	7,122	***	***
Modelo Convexificado*	2,220	4,097	5,932	623452	**1024.06
Modelo PDR	2,220	5322	8,958	41226	950.36
Modelo PDR con cotas, variables de holgura y ramificación con prioridad.	2,220	5,297	7,152	6328	950.36

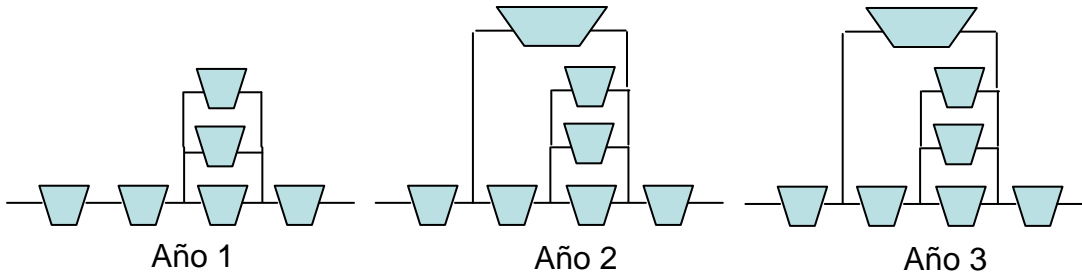
\* Convexificación propuesta por Vaselenak, Grossmann y Westerberg, 1987.

\*\* Mejor solución encontrada al agotar recursos de memoria.

\*\*\* El modelo se reporta como infactible.

El primer renglón muestra como no fue posible encontrar solución alguna al modelo no convexo. El modelo convexificado en su forma natural, encuentra solución, pero no es la óptima, y cabe notar el tiempo que tardó en encontrarlo, 623,452 segundos. El modelo disyunto (PDR) ya encuentra la solución óptima, tardando 41,226 segundos. Finalmente, el modelo (PDR), con las herramientas implementadas, encuentra la misma solución óptima en tan solo 6328 segundos.

Los Caso 1 y Caso 2 no utilizan la opción de hacer uso de un solo equipo para las etapas dos (fermentador) y tres (reposador). Esta opción se incluyó al método y es una de las aportaciones de esta tesis; por lo que es importante mostrar su funcionamiento.



**Figura 3.6:** Caso 3, Opción de utilizar un equipo para dos etapas.

Para el Caso 3, elevamos los costos de instalación, y la demanda en el primer período. Ésto aumenta la necesidad de una capacidad considerablemente mayor a la presente, y la opción de escoger un solo equipo para dos etapas es una solución más económica. Modificando los parámetros, para un modelo pequeño de tres períodos de tiempo, con la misma configuración inicial y una demanda más alta que los ejemplos uno y dos, se obtiene la configuración de planta de la Figura 3.6.

Habría que revisar los costos de operación contra los de instalación; en este trabajo no consideramos costos de operación, solamente los de inversión. Esto nos lleva a eliminar la posibilidad de analizar si el comprar una tecnología más cara, pero con menores costos de operación, pudiera ser una mejor inversión.

### 3.5 Conclusiones

El método algebraico para resolver MINLPs multiperíodo que involucran variables discretas para la decisión de la topología de la planta, la expansión y la operación en cada período de tiempo es combinatoriamente explosivo (Floudas, 2004), como hemos visto en este capítulo. Para atacar este problema de manera efectiva, se utiliza un modelo general de programación disyunta para diseño y planeación de redes de procesos industriales. Dicho modelo ha sido modificado para poder resolver nuestro problema específico.

El problema de diseño retrofit de plantas batch ha sido resuelto por medio del modelo propuesto con satisfacción para períodos cortos de tiempo.

Al utilizar programación disyunta atacamos el problema específico de reducir el tiempo computacional de los problemas maestros en cada paso de la iteración. Esto permite encontrar la solución óptima del modelo.

Además, utilizamos tres estrategias para ayudar a su convergencia. La primera fue colocar cotas más estrictas en dos variables decisivas para el modelo: el número y el tamaño de batch ( $N_{it}$  y  $B_{it}$ ). Esto se logra mediante manipulaciones algebraicas con el uso de la superestructura y de los límites de volúmenes permitidos.

La segunda adecuación fue el colocar variables de holgura en dos ecuaciones que se sabe de antemano son las “difíciles” del modelo. Esto permite que el método de solución pueda eliminar soluciones no factibles de manera más eficiente, mejorando considerablemente nuestros tiempos de solución. La tercera fue explotar la estructura del modelo disyunto con prioridades en la ramificación del algoritmo de solución. Estas tres estrategias ayudaron enormemente al modelo, mejorando su tiempo de cómputo en un orden de magnitud.

A pesar de utilizar la programación disyunta y las herramientas de ayuda, los tiempos para problemas de más de 10 períodos ya son excesivos. Valdría la pena explorar algún método para tratar de ayudar en la búsqueda de la configuración de planta. Recientemente algunos algoritmos específicos para diseño retrofit de plantas batch se han publicado en la literatura (Goel, 2004; Moreno, 2007) que se pudiesen probar para comparar resultados.

Se elabora un ejemplo pequeño para mostrar la posibilidad que maneja el modelo de juntar dos etapas del proceso en un solo equipo. Bajo las circunstancias del Caso 3, el modelo escoge agregar un tanque fermentador/reposador, donde las dos etapas de producción se llevan a cabo en una misma unidad. En la literatura, no se ha encontrado ningún trabajo que permita juntar dos etapas de proceso en un equipo.

La curva de demanda utilizada para los ejemplos es una curva suave, si llegara a presentar saltos grandes de demanda, la opción de utilizar un equipo para dos etapas sería

más atractiva para el modelo. Esto es debido principalmente a que el aumento en la demanda del producto sigue al crecimiento poblacional, que es una curva suave.



## **4. Planeación de la Producción y Horizonte Móvil Incorporados al Modelo Retrofit**

### **4.1 Introducción**

El propósito de este capítulo es elaborar, a partir del modelo de diseño retrofit del Capítulo 3, un modelo mas robusto con capacidad de dar resultados para muchos períodos de tiempo.

Al modelo del Capítulo 3, se le detectaron dos problemas importantes. Primero, el modelo no encuentra soluciones para problemas grandes. Un modelo de diez períodos tarda más de once horas en encontrar solución. Un modelo de 20 períodos no puede encontrar solución al problema. Recordemos que el modelo es planteado para diseño de plantas batch a largo plazo. Lo que el cliente busca es un calendario de expansiones e inversiones, un algoritmo que sólo pueda resolver un par de períodos no es útil. Para corregir este problema se propone utilizar un algoritmo aproximado de horizonte móvil, que han sido ampliamente estudiados particularmente en el área de control y manejo de inventarios Chand, Ning y Sethi (2004) hacen una recopilación al respecto.

El segundo problema es que, en la industria, la demanda se tiene que cumplir de forma catorcenal, nuestro modelo es de campañas de un solo producto y la demanda se cumple de manera anual. Esto se traduce en que nuestros resultados no solo son optimistas, sino que en realidad son infactibles al escalar la capacidad (o demanda de ella) a catorcenas. Para corregir este error, se propone un modelo de desagregación, en donde una vez que se tiene fija la configuración de la planta, se corre un modelo NLP para obtener los valores de las variables continuas como volúmenes, tamaño de batch, etc.

## **4.2 Horizonte Móvil**

Los modelos presentados en las secciones anteriores pueden resolverse directamente con métodos de “ramificar y acotar”. Para problemas grandes y complejos, involucrando horizontes de tiempo a largo plazo, el esfuerzo computacional puede ser intenso y de tiempo considerable. Por tanto consideramos la alternativa de incluir un modelo de horizontes móviles para agilizar el tiempo de solución y poder incluir ejemplos de períodos de tiempo considerables.

En esta sección se describe el algoritmo de horizontes móviles, el cual es una aproximación utilizada para reducir el esfuerzo computacional de problemas multiperíodos grandes. En lugar de utilizar el horizonte completo de diseño, se descompone el problema en una secuencia de sub-problemas que se resuelven de manera recursiva (Beraldi et al., 2008). Todos los algoritmos de horizontes móviles proporcionan aproximaciones de las soluciones detalladas con una reducción significativa en sus requerimientos computacionales.

En el área de sistemas de proceso encontramos algunas referencias que utilizan algoritmos de horizonte móvil. Shah y Pantelides (1991) proponen un algoritmo de horizonte

móvil inverso para el secuenciamiento a corto plazo de la operación de homogenización de mezclas de lubricantes. Dimitriadis, Shah y Pantelides (1997) presentaron un algoritmo de horizonte móvil para plantas multi-propósito donde hacen uso del algoritmo de Wilkinson et al. (1996).

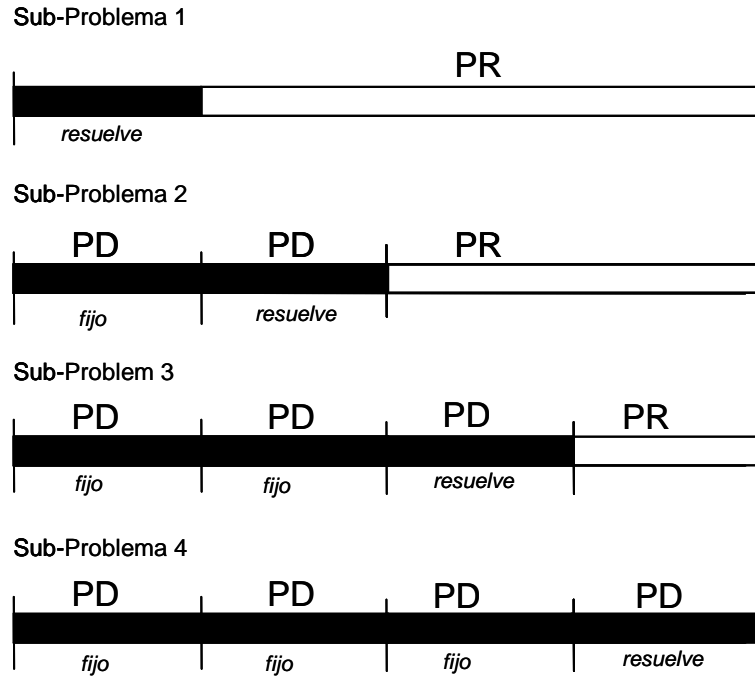
Para propósitos de esta tesis, escogimos un algoritmo de horizontes móviles novedoso. La novedad radica en que para cada sub-problema, solamente la parte inicial del horizonte es modelada con el problema detallado (PD), el resto del horizonte es modelado con el modelo relajado de diseño (PR), para un horizonte de  $H$  períodos de tiempo y tomando  $r$  períodos a la vez. La solución de esta parte detallada, se fija a los valores obtenidos, y se prosigue con el siguiente sub-problema.

#### Algoritmo general de Horizontes Móviles

1. Contador  $f=r$
2. Relajar variables binarias para el períodos en rango  $[f+1,H]$
3. Resolver modelo
4. Fijar variables binarias obtenidas en la solución para el rango  $[0,f]$
5. Si  $f = H$  parar;
6. Ajustar contador,  $f=f+r$  si  $f+r \leq H$   
 $f=H$  si  $f+r > H$
7. Volver a 2.

Dogan y Grossmann (2007) es la única referencia publicada dónde se utiliza la relajación de las variables discretas para el horizonte en cuestión. Esto ayuda al modelo puesto que

sigue teniendo la información de todo el horizonte de diseño en cada iteración del horizonte móvil.



**Figura 4.1.** Algoritmo de horizonte móvil (Erdirik-Dogan y Grossmann, 2007).

Como la Figura 4.1 indica, el horizonte de modelación se parte en tres secciones, la primera que tiene los valores de las variables discretas fijas en alguna iteración anterior, la segunda en donde se está resolviendo el problema detallado y la última en donde las variables binarias están relajadas. Con cada sub-problema los períodos cubiertos por el algoritmo detallado aumentan, mientras los períodos cubiertos por el modelo relajado decrecen.

Nótese que la complejidad computacional de dicho algoritmo es similar a la del PDR ya que en cada iteración del Algoritmo de Horizontes Móviles se resuelve PDR. Además el número de iteraciones es relativamente bajo a pesar de los crecientes bloques del problema

detallado debido a que se están fijando algunas variables a sus valores óptimos obtenidos en sub-problemas previos. Este esquema recurrente continúa hasta que el horizonte completo ha sido solucionado con PD.

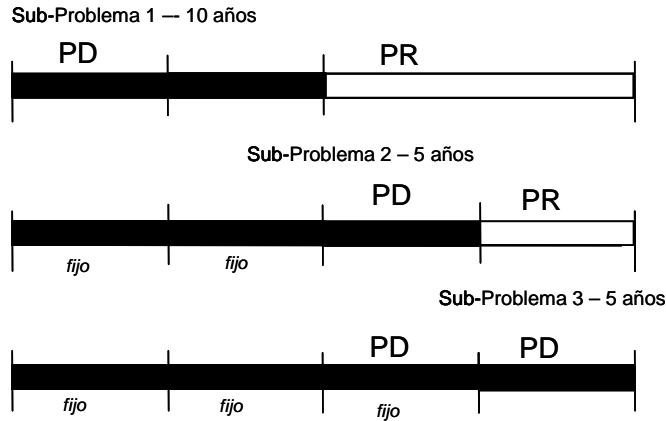
Aunque es posible fijar todas las variables a sus valores óptimos obtenidos en los sub-problemas previos, escogimos fijar solamente las variables binarias, es decir la configuración de la planta. Las variables continuas (volúmenes, tamaño de batch, etc.) se dejan libres para reducir posibles complicaciones por infactibilidad.

Otra decisión concierne a la longitud del horizonte que va a ser solucionado por el modelo detallado en cada sub-problema. Dependiendo del esfuerzo computacional requerido, se puede ajustar el número de períodos considerados. Para nuestro problema se ejecutan diez períodos de tiempo en el problema detallado, seguido de dos sub-problemas que adicionan períodos de cinco años. Ésto es debido principalmente, a que sabemos que podemos encontrar solución para el problema de diez períodos.

En la Figura 4.2, observamos que el algoritmo necesita de tres iteraciones. La primera en donde se encuentra la solución de los primeros diez períodos de tiempo, y se relajan las variables binarias de los últimos diez períodos.

En la segunda iteración, se resuelve el problema detallado para los primeros quince períodos, con las variables binarias de los primeros diez períodos fijas a los valores de la primera iteración, y los últimos cinco períodos están relajados.

Para la tercera iteración, se fijan las binarias del problema detallado de los primeros quince períodos con los resultados de las iteraciones previas, y se corre el algoritmo detallado para el horizonte completo de veinte períodos.



**Figura 4.2.** Bloque detallado en horizonte de 20 años

#### 4.2.1. Ejemplos Algoritmo Horizonte Móvil

La aplicación de los modelos propuestos y las técnicas de solución se ilustran con dos ejemplos, el último de ellos corresponde a un problema industrial. Los datos para los problemas se encuentran en los apéndices.

En la segunda columna, que indica el número de variables binarias, podemos ver el funcionamiento del algoritmo. El total de las variables binarias en los ejemplos no cambia ya que estamos resolviendo el mismo ejemplo. El total de las variables relajadas va disminuyendo en cada iteración. El número de variables binarias activas es el mismo para cada iteración ya que se va recorriendo el bloque de tiempo a resolverse, fijando las anteriores y relajando las posteriores del bloque.

En el primer ejemplo, retomamos el ejemplo del modelo de diseño retrofit para diez períodos de tiempo presentado en el Capítulo 3, éste correspondiente al primer renglón de variables de la Tabla 4.1. Posteriormente resuelve el problema relajado, a fin de poder comparar tiempo sus tiempos de cómputo. El problema relajado, un NLP, aunque no es factible, se resuelve en una fracción de su complemento discreto. Ésta es la motivación para el algoritmo de horizontes móviles.

**Tabla 4.1.** Estadísticas del modelo y solución de Caso 1 para 10 períodos

modelo	número de variables binarias	número de variables continuas	número de ecuaciones	tiempo seg. CPU	solución \$
Problema detallado	4,440	10,463	13,958	41226	950.36
Problema detallado (relajado)	4440R	10,463	13,958	425	524.0674
PD 5 períodos					
PR 5 períodos	2200 (2240R)	10,463	13,958	588	578.82
PD 5 fijos 5	2,240	10,463	13,958	239	1087.8
			FINAL	<b>828</b>	<b>1087.8</b>
PD 2 períodos					
PR 8 períodos	920 (3520 R)	10,463	13,958	94	578.82
PD 2-2 períodos PR 6					
períodos	920 (2,640R)	10,463	13,958	70	578.82
PD 4-2 períodos PR 4					
períodos	920 (1760R)	10,463	13,958	98	578.82
PD 6-2 períodos PR 2					
períodos	920 (880R)	10,463	13,958	72	752.5231
PD 8-2 períodos	920	10,463	13,958	65	1087.8365
			FINAL	<b>401</b>	<b>1087.8365</b>

El método de horizontes móviles pretende utilizar la información del método relajado y su velocidad, para ir encontrando durante horizontes más cortos, las variables discretas del sub-problema en cuestión.

Nótese que el número total de variables y de ecuaciones no cambia; solamente las variables discretas cambian de un modelo a otro, sin embargo su efecto en los tiempos de cómputo es considerable.

El tercer renglón de la tabla es el primer paso del horizonte móvil, donde se ejecuta el modelo para los primero cinco períodos de tiempo y se relajan las variables binarias para los siguientes cinco períodos. En el cuarto renglón es el segundo paso para el algoritmo de horizonte móvil, donde se fija las variables binarias obtenidas en el primer sub-problema y se

resuelve para los siguientes cinco períodos. Este método de horizonte móvil encuentra una solución de \$1087, comparado con \$950 del problema original, esto representa un 14.4% de error de optimalidad. Sin embargo, nótese la diferencia en sus tiempos de cómputo 41,226 segundos (11.4hr) para el problema original contra 828 segundos (0.23hr) para la solución de horizonte móvil. Este resultado de horizonte móvil es tomando en cuenta los sub-problemas de cinco períodos de longitud.

Posteriormente en la tabla se presentan los resultados al utilizar el método de horizontes móviles para la misma instancia, pero tomando sub-problemas de dos períodos de longitud. Esto es para observar como se desempeñaba la solución al tomar sub-problemas de menor número de períodos.

Para este ejemplo, como se observa en la Tabla 4.1, el resultado es el mismo que tomando sub-problemas de cinco períodos, sin embargo, si se toman sub-problemas de dos períodos el ahorro en tiempo de computo es aun mayor. Típicamente, entre más grande sea el número de períodos de los sub-problemas, más se acerca el resultado a la solución óptima. En este ejemplo en particular, no fue así, seguramente porque el tamaño del problema no es demasiado grande.

En el segundo ejemplo, probamos una instancia del modelo de diseño retrofit de plantas batch del Capítulo 2, con un horizonte de 20 años. En la Tabla 4.2 se presentan los datos de las iteraciones del algoritmo de horizontes móviles para un horizonte de 20 períodos de tiempo.

Nótese como con el modelo anterior no era posible encontrar solución en 5 días y con el algoritmo propuesto se encuentra una solución en menos de 5 horas.



**Tabla 4.2.** Estadísticas del modelo y solución de ejemplo 2 para 20 períodos

Modelo	Número de Variables Binarias	Número de Variables Continuas	Número de Ecuaciones	Tiempo de Cómputo seg CPU	Valor Objetivo \$	
Problema Detallado	8,840	20,883	27,798	***	***	
Problema Relajado	8840R	20,883	27,798	251	524.07	
Problema de Horizonte Móvil	PD 5 períodos PR 15 períodos	2,240 (6600R)	20883	27,798	937	1801.74
	PD 5 períodos PR 10 períodos	2,240 (4,400R)	20883	27,798	1915	1801.74
	PD 5 períodos PR 5 períodos	2,240 (2,200R)	20883	27,798	6928	2147.0
	PD 5 períodos 15 períodos fijos	2240	20883	27,798	7463	3526.60
				<b>FINAL</b>	<b>17143</b>	<b>3526.60</b>

\*\*\* No encontró solución después de 20 iteraciones o 5días (432,000seg.)

Con esto podemos concluir que el algoritmo está listo para solucionar problemas de gran escala, y podríamos incluso intentar llevarlo a solucionar un problema de más períodos de tiempo.

Aunque originalmente no encontramos solución con el algoritmo del Capítulo 3, si se logró encontrar la solución óptima, alimentándole la solución del algoritmo de horizontes móviles, y bajando los criterios de convergencia entre las iteraciones del MINLP, en nuestro caso DICOPT. La solución óptima encontrada con esta estrategia es de 3244.78, en un tiempo de 440,190 segundos. Aunque los tiempos no son comparables por la diferencia de criterios de optimalidad, el valor de la solución nos da una medida del error de la aproximación encontrada, al comparar este resultado con el valor reportado en la Table 4.2. En este caso tenemos un 8.68% de error de optimalidad.

### **4.3 Planeación de la Producción**

El entender la representación adecuada de la capacidad de las instalaciones de una planta tiene un impacto financiero significativo ya que resulta en una habilidad incrementada de satisfacer órdenes comprometidas y ayuda a identificar nuevas oportunidades de mercado.

La meta en la planeación de la producción es la de determinar la capacidad de producción en términos de decisiones de alto nivel tales como: niveles de producción e inventario para un pronóstico de mercado dado, y demandas sobre un horizonte de mediano plazo que puede ir desde un par de meses hasta un año. Típicamente, los modelos de planeación son lineales y representaciones simplificadas que son usadas para predecir niveles de producción y flujo de materiales. Los algoritmos de planeación de la producción pueden ser discretos o continuos, dependiendo de la manera en que se maneje el tiempo; una extensa bibliografía sobre el tema puede encontrarse en Floudas y Lin (2004).

Generalmente, las metas de producción obtenidas en este nivel son optimistas ya que los efectos de secuenciamiento son ignorados. Efectos de secuenciación ocurren cuando la producción de la línea cambia de un producto a otro. Estos cambios pueden ser asociados a cambios en las condiciones de operación o a la limpieza del equipo. Si la secuenciación depende de la manera en que se cambian los productos en la línea, entonces la utilización de la capacidad dependerá de la secuencia en que son elaborados los productos. Para fines de este trabajo, no se tomarán en cuenta efectos de secuenciación. Principalmente porque el objetivo de investigación de este trabajo es el diseño de plantas batch y no de planeación de su producción. Sin embargo, con el uso adecuado de la receta de cada producto, y la

utilización de castigos o motivaciones en forma de tiempo, se podrían incorporar al modelo sin mucha dificultad la secuenciación de los productos.

Varios modelos de producción táctica manejan simultáneamente el problema de determinar la capacidad disponible. Por ejemplo, McDonald y Karimi (1997) presentan un modelo multiperíodo a mediano plazo de planeación para procesos semi-contínuos donde el objetivo principal es colocar equipos a tareas de producción a manera de satisfacer demandas fluctuantes sobre un horizonte de tiempo extendido. El tiempo y secuenciamiento de las campañas de producción no son determinadas. Otro ejemplo lo presentan Bok, Grossmann y Park (2000), donde se estudia un modelo de planeación multiperíodo para redes de procesos continuos sobre un horizonte a corto plazo que va desde una semana hasta un mes. El modelo propuesto incorpora perfiles de inventarios, abastecimientos intermitentes y producciones por debajo de la meta obteniendo como resultado un calendario de asignación de tareas.

Un enfoque para considerar el problema de determinar la capacidad, es formular simultáneamente los modelos de planeación y de diseño de manera que se cubra todo el horizonte de tiempo de interés. Sin embargo, la limitante con este enfoque es que cuando los horizontes de diseño son considerados, el tamaño del modelo de planeación se hace intratable debido al crecimiento exponencial en el tiempo de cómputo. La manera tradicional de manejar este problema ha sido el seguir una estrategia jerárquica (Bitran y Hax, 1977) en donde el problema de diseño se resuelve primero para definir las metas de producción. El problema de planeación se reduce a un sub-problema de secuenciación y se resuelve para cumplir con las metas exigidas por el modelo de diseño. Una complicación con este enfoque es que la solución obtenida por el problema de diseño no es necesariamente factible

relacionada al problema de planeación. La mayoría de los trabajos reportados aplican técnicas heurísticas para sobrellevar las infactibilidades y no garantizan el óptimo global (Subrahmanyam, Pekny y Reklaitis 1996; Daichendt y Grossmann,1998).

Bassett, Pekny y Reklaitis (1996) propusieron un esquema de descomposición para plantas batch multipropósito donde un problema agregado de diseño se resuelve en un nivel superior, y un problema de planeación detallado se resuelve independientemente para cada período de diseño en nivel inferior. Los autores aplican técnicas heurísticas que hacen uso del cambio de operaciones para evitar inconsistencias entre el diseño y la planeación. Stefansson, Shah y Jensson (2006) proponen un enfoque a la planeación y diseño basado en un horizonte móvil jerárquicamente estructurado para plantas batch multi-propósito. En cada nivel del marco jerárquicamente estructurado proponen modelos de optimización para proveer apoyo para decisiones relevantes donde cada nivel difiere al respecto de la información disponible y obtenida.

Wu e Ierapetritou (2007) también utilizan un algoritmo jerárquico pero además incluyen el manejo de incertidumbre en la demanda de los productos.

Una segunda opción para manejar las dificultades relacionadas al tiempo de cómputo es el utilizar un enfoque de agregación. Wilkinson, Shah y Pantelides (1996) utilizan este enfoque para obtener soluciones aproximadas para el problema a gran escala de producción y distribución, con problemas de planeación para diferentes sitios de producción representando los sitios por redes de tareas y recursos (RTN por sus siglas en inglés “Resource Task Network”).

Pantelides (1994) parte el horizonte de diseño en porciones más pequeñas denotadas como períodos de tiempo agregados (“aggregated time periods”, ATP). Ahí describe cada

ATP por variables agregadas, cada una equivalente a la suma ponderada de las correspondientes variables detalladas sobre los intervalos de tiempo. La formulación agregada es entonces generada reemplazando grupos de variables relacionadas de la formulación detallada por las variables agregadas correspondientes de manera que se reduce el tamaño del problema.

Otro enfoque de atacar simultáneamente el diseño y la planeación es la descomposición a dos niveles. En lugar de resolver el horizonte de planeación completo, se descompone el modelo en una secuencia de sub-problemas que se resuelven recursivamente (Dimitriadis, Shah y Pantelides, 1997).

Dogan y Grossmann (2007) han propuesto un algoritmo de descomposición a dos niveles donde se descompone el problema detallado original en un nivel superior de diseño y uno inferior de diseño/planeación. Como el primero da una cota superior y el segundo una cota inferior, la convergencia se obtiene cuando las cotas caen entre alguna tolerancia dada.

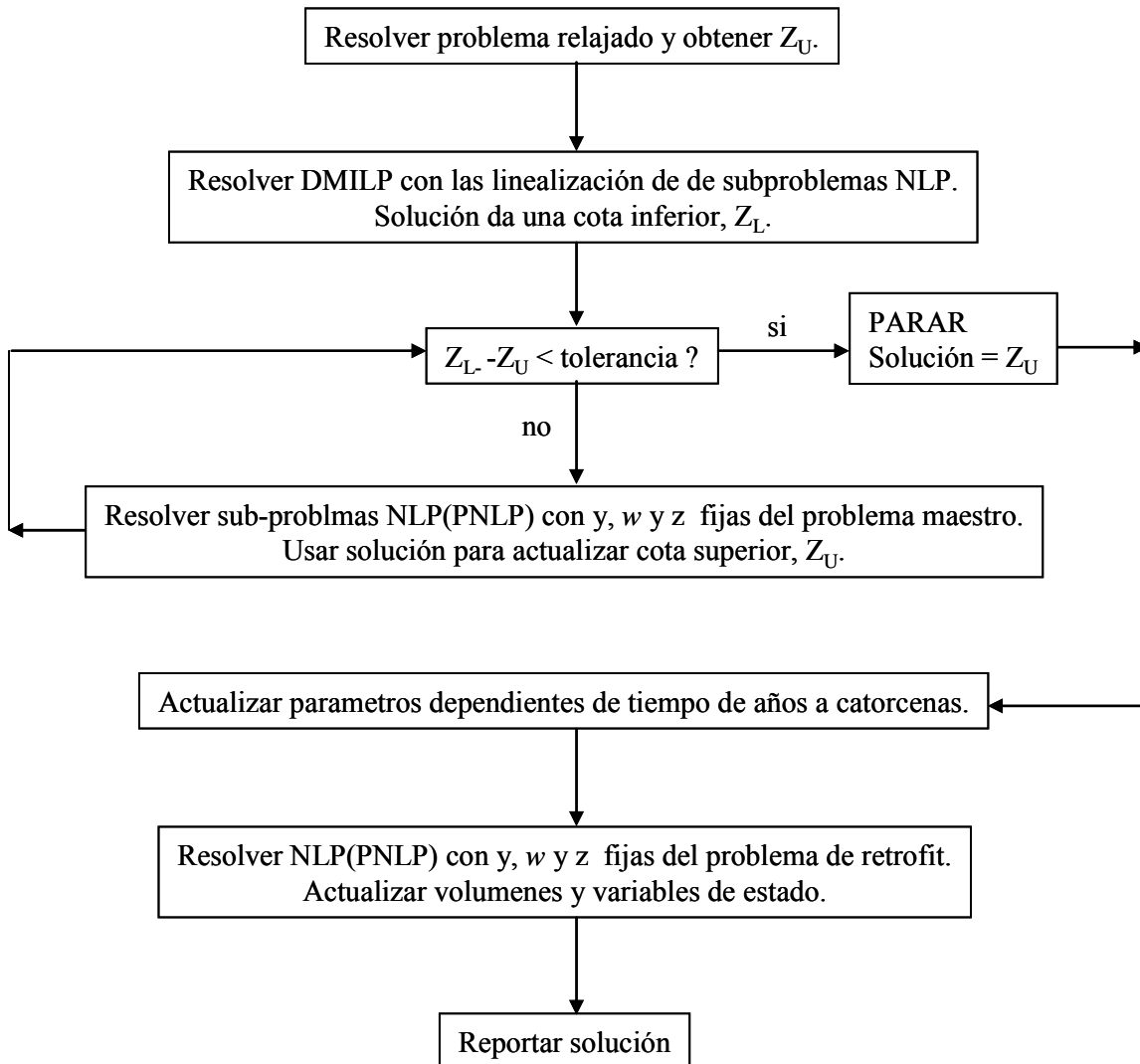
Finalmente, un enfoque reciente es el utilizar la “región de producción alcanzable” (PAR, por sus siglas en inglés, production attainable region) como referencia para los modelos de diseño y planeación. Sung y Maravelias (2006) proponen reemplazar los modelos de planeación (por ejemplo el modelo STN) con una aproximación de la región factible en donde el modelo de planeación es proyectado hacia variables de diseño relevantes como flujos de producción, utilizando un procedimiento numérico para encontrar envolventes convexos.

#### **4.4 Algoritmo Ajustado para Diseño Retrofit de Plantas Batch**

La planeación de la producción se utiliza como herramienta para mitigar el error introducido al utilizar campañas de un solo producto. Dado que contamos con nuestro

modelo de retrofit multiperíodo presentado en el Capítulo 3, la selección natural fue un modelo de descomposición.

Se utiliza el mismo modelo de optimización disyunto presentado en el Capítulo 3 (PDR), para la solución de un problema con demandas catorcenales. El algoritmo propuesto para el diseño retrofit de plantas batch se muestra en la Figura 4.3.



**Figura 4.3.** Algoritmo para diseño retrofit de plantas batch con corrección de planeación

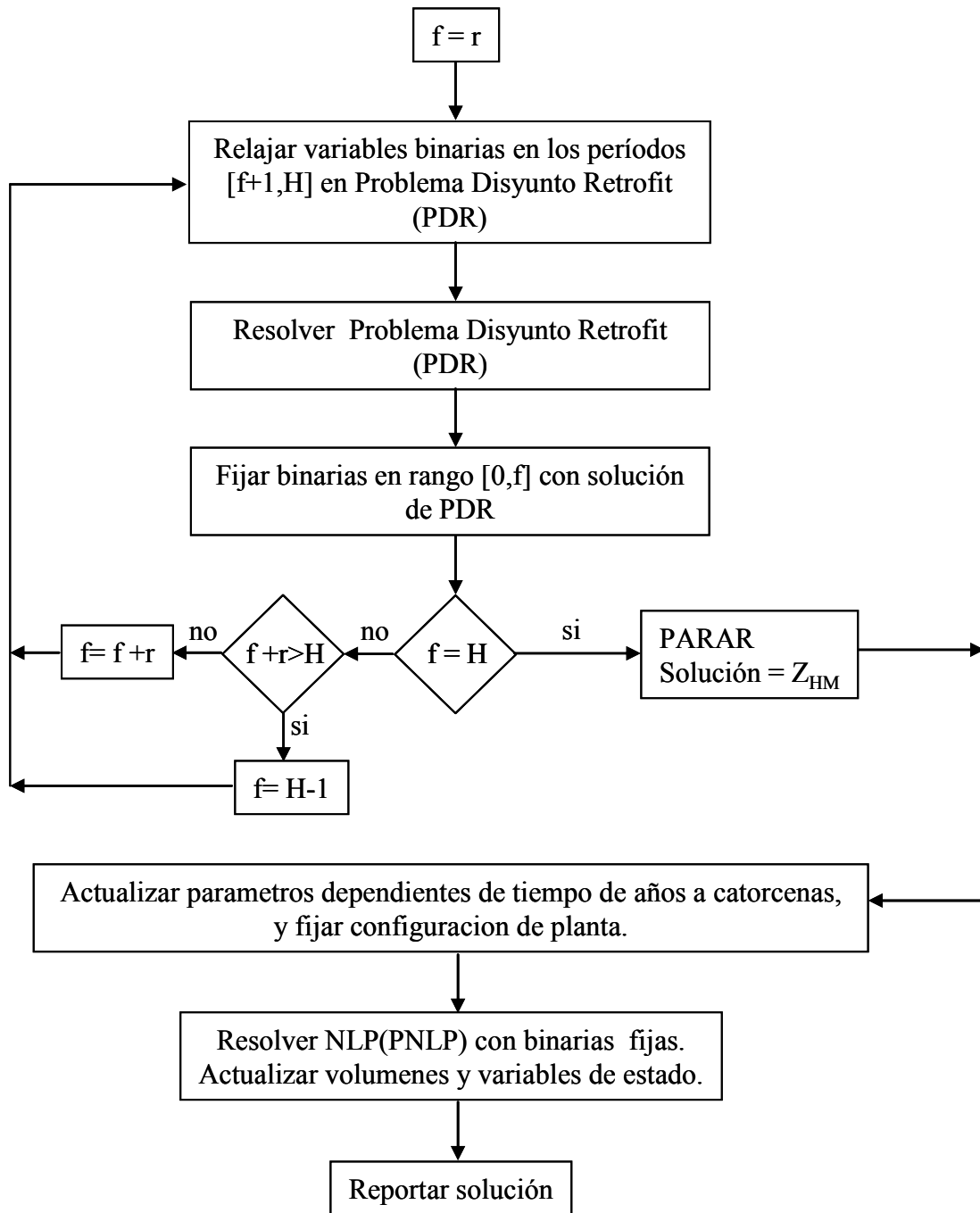
Al fijar las variables binarias, nos quedamos con el modelo no lineal (PNLP) del Capítulo 3.

Se consideró dejar variable a la variable binaria  $w$ , que está relacionada con la operación de la planta, esto resultaría en correr un segundo MINLP para la desagregación. Sin embargo, debido a la estructura jerárquica de la programación disyunta, es imposible que si una unidad fue expandida no sea operada, por lo que dejamos a  $w$  fija también. De éste modo solo obtenemos corrección en los volúmenes de los equipos que ya se habían seleccionado previamente en la solución del modelo (DMILP), por medio de un NLP que es más sencillo de resolver.

Nótese que las instancias a resolver son las mismas que en el Capítulo 3, lo que cambia son los parámetros. Específicamente, la demanda por producto, en lugar de la demanda anual, se opera el modelo con una demanda catorcenal; además el parámetro  $H_t$ , que define el número de horas en que se tiene que cumplir la demanda. En el Capítulo 3,  $H_t$  tenía un valor de 8000 horas (de operación de la planta en un año) y para el problema de planeación, el horizonte de producción es de 330 horas (de producción de la planta en catorce días).

Si juntamos el algoritmo de horizontes móviles para resolver el modelo más rápido y tomamos en cuenta las demandas catorcenales, nos queda el algoritmo de la Figura 4.4 para resolver nuestro problema de diseño retrofit para plantas batch.

En la Figura 4.4,  $f$  es un contador y  $r$  es el número de períodos a tomar para el método de horizonte móvil.  $Z_{HM}$ ,  $w_{HM}$  e  $y_{zHM}$  son, el valor objetivo, la variable binaria relacionada con el modo de operación, y la variable binaria relacionada con la expansión de una unidad, respectivamente, encontradas por el método de horizonte móvil.



**Figura 4.4.** Algoritmo para diseño retrofit de plantas batch con corrección de planeación y horizontes móviles



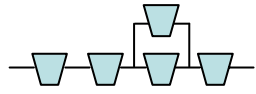
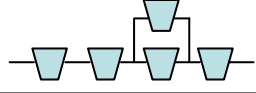
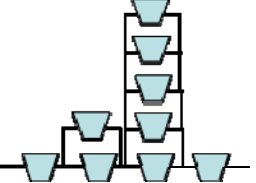
Cabe mencionar que las variables binarias  $y$ , no se fijan en el algoritmo de horizontes móviles puesto que no dependen del tiempo. Éstas son las que definen la superestructura de la red y son “verdaderas” cuando un equipo es seleccionado para formar parte de la planta.

Tomando el ejemplo de diez períodos, con valor objetivo de \$1087.83 encontrado por el método de horizonte móvil, ahora realizamos la desagregación de la demanda, queriendo obtener valores de volumen de equipos corregidos que permitan cumplir con las demandas catorcenales. La configuración de la planta no cambia, puesto que es dada al modelo PNLPCatorcenal. La desagregación de las demandas a catorcenas hace que el valor objetivo aumente a \$1,132.4. Esto es reflejo de que se instalaron las mismas unidades, pero de un mayor volumen.

## **4.5 Casos de estudio**

Con el algoritmo para diseño retrofit de plantas batch con corrección de planeación y horizontes móviles listo para obtener resultados, evaluaremos tres casos de estudio con una serie de problemas para cada caso. En la Tabla 4.3, se describen brevemente cada caso. Estos ejemplos fueron creados pensando en probar al modelo desde diferentes perspectivas. En el primer caso, la idea es generar otra instancia y aumentar el horizonte del problema aún más, para ver si el algoritmo es consistente en sus soluciones. En el segundo caso, se analiza la respuesta otorgada por el modelo; esto se hace resolviendo el modelo en tres puntos de demanda cercanos y comparando los resultados encontrados. Para el último caso, queremos observar el tamaño de la infactibilidad al hacer la desagregación catorcenal; esto mediante una modelación de la configuración de planta encontrada en los casos 1 y 2.

**Tabla 4.3.** Casos Capítulo 4

	configuración inicial	número de etapas	número de productos	número de periodos
Caso 1		4	3	5 10 15 20 25
Caso 2		4	3	10
Caso 3		4	3	10

En el primer caso utilizaremos otros datos de demanda y aumentaremos de 5 en 5 hasta 25, los períodos de tiempo para probar el modelo para períodos aún más grandes de tiempo, evaluaremos el desempeño del algoritmo al incrementar su horizonte de planeación, comparando los resultados de modelos de diferentes tamaños.

Para el segundo caso, tomando en cuenta la incertidumbre presente en la demanda, asumiremos que sigue un comportamiento normal, probando el modelo en dos puntos alrededor de la demanda  $\mu \pm \sigma$ . Esto ayudará como herramienta para la toma de decisiones en cuanto al calendario de inversión que seguirá la empresa en cuestión.

Para el último caso de estudio abordaremos un problema de modelación. La receta de los productos maneja rangos de tiempo y no datos puntuales. Para todos los ejemplos visitados se utilizó un promedio del rango dado de cada parámetro para cada producto. Queremos ver, si al optimizar el calendario de expansiones para el promedio de las recetas, podemos encontrar capacidades factibles al modelar para las cotas inferiores de tiempos de producción por etapa en un par de catorcenas pico. De encontrar respuestas factibles, podríamos eliminar la desagregación del modelo a catorcenas.

Caso 1

Las nuevas curvas de demanda no incrementan con la misma rapidez de los ejemplos revisados anteriormente. El producto 1, considerado el producto de especialidad tiene un aumento de 2% anual, el producto 2, de la familia de productos light y con mayor crecimiento de mercado, tiene un crecimiento de 13% anual para los primeros años, decayendo gradualmente hasta un 10 % anual, y el producto 3 un crecimiento fijo de 6% anual.

Obtenemos nuevas curvas de demanda (Apéndice C), los parámetros y configuración inicial de la planta son los mismos de los casos anteriores, con estos datos resolvemos el modelo hasta 25 períodos con ello obtuvimos los resultados en la Figura 4.5

Con esto podemos comprobar que el modelo funciona bien para otras instancias del problema, encontrado resultados hasta para 25 períodos de tiempo. Los resultados del algoritmo de horizontes móviles pueden ser utilizados como solución para empezar la búsqueda si el óptimo global es de interés.

**Tabla 4.4.** Resultados del algoritmo para diseño retrofit de plantas batch con corrección de planeación y horizontes móviles

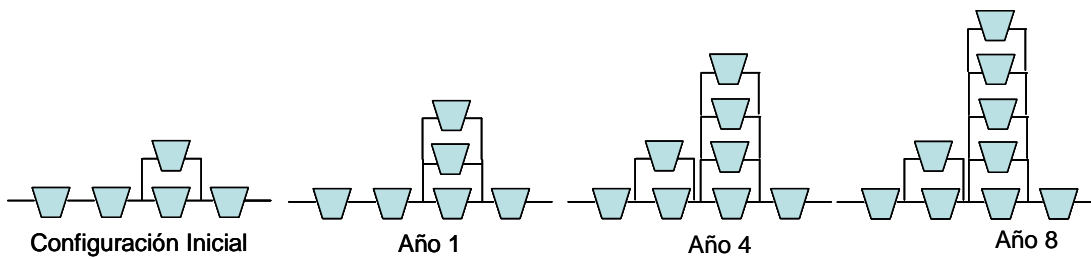
Número de Períodos	Número de Variables Binarias	Número de variables continuas	Número de Ecuaciones	Tiempo de Cómputo seg CPU	Valor Objetivo HM \$1000	Valor Objetivo PNLP \$1000
5	1568	3695	4981	21	253.92*	296.13
10	3,108	7,360	9,876	440	563.77*	584.34
15	4,648	11,025	14,771	8384	732.58	805.84
20	6,188	14,690	19,666	27478	1008.30	1038.55
25	7728	18,355	24,561	4574	1119.73	1212.67

\* no se realizó el algoritmo de horizontes móviles, es la solución óptima

Los resultados reflejan el comportamiento de los ejemplos anteriores, las variables y ecuaciones aumentan en bloques con el número de períodos, y el tiempo sigue una curva ascendente que parecería exponencial.

Caso 2

Con las demandas del Caso 1, obtenemos dos puntos a una distancia de  $\pm 5\%$ . Utilizando el ejemplo de 10 períodos, se pretende obtener las diferentes configuraciones de la planta para cada caso. Así fue como obtuvimos el calendario de expansión que se ve en la Figura 4.5. Es interesante que para los tres puntos de demanda, el calendario de expansión se mantiene. Es decir, siempre se obtiene la misma configuración de planta como solución.



**Figura 4.5.** Configuraciones de planta para diferentes demandas.

Esta información es de mucho valor, puesto que habla sobre lo robusto de la solución encontrada. Evidentemente las respuestas no tienen el mismo valor óptimo, debido a que se requiere de volúmenes diferentes para cumplir las distintas demandas, sin embargo el que no cambie la configuración es un hallazgo de interés. Los tiempos de solución y valores óptimos se presentan en la Tabla 4.5. Aunque los tiempos de solución para la instancia con mayor demanda aumentan considerablemente, las soluciones son muy cercanas, esto es debido principalmente a que el costo de instalación fijo es más alto que el variable que depende del tamaño del equipo.

**Tabla 4.5.** Resultados para tres puntos de demanda.

Modelo 10 Períodos	Número de Variables Binarias	Número de Variables Continuas	Número de Ecuaciones	Tiempo de Cómputo seg CPU	Valor Objetivo \$1000
demanda + 5%	3,108	7,360	9,876	1058	563.77
demanda	3,108	7,360	9,876	440	563.77
demanda - 5%	3,108	7,360	9,876	496	535.80

Incluso, para el caso de la demanda y la demanda +5% el valor objetivo es el mismo. Esto parecía incorrecto, pero no es así. Lo que pasa es que varios de los equipos están expandiéndose en el límite inferior posible, es decir, les sobra algo de capacidad.

Caso 3.

El Caso 3 es un problema de modelación. Queremos ver si la capacidad obtenida por el algoritmo de horizontes móviles es capaz de satisfacer la demanda catorcenal si utilizamos la cota inferior de tiempo por etapa para cada uno de los productos. Si obtenemos resultados factibles, o cuasi-factibles podríamos considerar eliminar la desagregación de la demanda para el modelo a entregar a la empresa. Esta inquietud está basada en que no cambia la estructura de la planta y los resultados obtenidos hasta el momento presentan solamente aumentos marginales a los tamaños de los equipos según la configuración dada.

El modelo sí presenta problemas de factibilidad, en particular en dos ecuaciones:

$$\text{la iv) } \sum_i \exp(x_{it}^{(1)} + x_{it}^{(3)}) \leq H_t + P_2 \quad \forall t, y$$

$$x_{it}^{(2)} - \ln B_{it} \# P_7$$

Para ser factible necesitaríamos las variables  $P_2$  y  $P_7$  con valores de cero después de la simulación, pero no fue el caso.

La primera ecuación iv) tiene que ver con el horizonte de producción, la variable  $P_2$  toma un valor de 0.077. Considerando que H son las horas de operación en una catorcena y tiene un valor de 333, un 0.077 de merma de tiempo es despreciable.

La segunda ecuación, con infactibilidad es más delicada.  $x^{(2)}_{it}$  es la variable transformada del número de batches,  $B_{it}$  para la convexificación del modelo.

$x^{(2)}_{it} = \ln[B_{it}]$  es la definición de la variable. Sin embargo, la variable *holgura7* toma un valor de 0.057, comparándola contra los valores de  $B_{it}$  que para este caso andan en el rango de 0.3-0.5 (miles de litros) es pequeña.

Se plantearía al cliente la posibilidad de eliminar la desagregación del modelo para la corrección de la demanda catorcena, de manera que fuese lo más sencillo posible su algoritmo. Sin embargo, para cuestión matemática es de interés mantenerlo, tratando incluso de encontrar la solución óptima, iterando entre el modelo original y el desagregado.

## 4.6 Conclusiones

En el Capítulo 3 presentamos un modelo disyunto para el diseño retrofit de plantas batch. El modelo presentado se topa en la práctica con dos problemas grandes, no encuentra resultados para el número de períodos de interés al problema y su programación agrupada anual de producción lo aleja mucho de la realidad. En este capítulo, se propone un método que resuelve estos dos puntos en específico por considerarlos importantes para el cliente.

Al hacer uso de un algoritmo de horizontes móviles con el esquema presentado se tiene la ventaja de que utiliza la información de demandas en los períodos posteriores usando sus variables relajadas. Esto es conveniente puesto que permite al modelo anticipar demandas futuras y usar esa información cuando decide instalaciones y volúmenes de equipos nuevos.

Los métodos tradicionales de horizontes móviles solo van fijando y recorriendo el rango del horizonte en donde se esté evaluando el modelo, ignorando la información de períodos de tiempo no pertenecientes al rango del horizonte por resolver. Esto es debido principalmente a que por el uso del horizonte móvil se intenta reducir el tamaño del problema. Con el algoritmo presentado, no queremos reducir el tamaño del problema, sino hacerlo más tratable, ayudándole relajando o fijando la mayoría de las variables binarias que tiene que resolver.

Las soluciones obtenidas son buenas aproximaciones de la solución detallada, con la ventaja de ser significativamente más rápidos. Adicionalmente, el algoritmo da una cota superior del problema, la cual puede ser utilizada para inicializar algún algoritmo detallado de ramificar y acotar. En el ejemplo de 20 períodos que probamos encontramos un 8.68% de error de optimalidad y para el caso de diez períodos de tiempo la solución aproximada tiene un 12% de error, pero mejora sus tiempos de cómputo en un orden de 100 veces.

La planeación de la producción puede hacerse de varias formas. Existen diferentes modelos, cada uno con sus ventajas y desventajas. Aquí se utilizó el modelo de desagregación ya que se contaba con el modelo, con lo cual implementar la desagregación de la demanda fue solo cuestión de cambiar algunos parámetros del problema.

Por último cerramos el capítulo con tres casos de estudio. En el primero intentamos probar el funcionamiento adecuado del algoritmo para otras instancias del problema; los últimos dos casos fueron sugeridos por la industria con la cual estamos trabajando. En el Caso 2 se trató de obtener diferentes escenarios para posibles demandas. Este sería el primer paso a un estudio de sensibilidad e incertidumbre. Con este modelo funcionando, se da un paso importante a una posible extensión de modelación estocástica. Valdría la pena

profundizar para ver cuando cambia la configuración de la planta al aumentar o disminuir la demanda.

El último caso tiene que ver con el hecho de que los tiempos en las recetas de los productos no son puntuales, es decir, hay un rango de tiempo en cada etapa en el cual el producto dado se puede producir. Para el ejemplo modelado obtuvimos resultados cuasi-factibles; lo cual sabemos no importaría al cliente si se elimina un paso del algoritmo. Sin embargo se recomendaría más experimentación al respecto antes de eliminar al modelo de desagregación del algoritmo general.

La pérdida de optimalidad por hacer uso del algoritmo de horizontes móviles y posteriormente la corrección a demandas catorcenales, es un sacrificio prudente puesto que en los problemas de diseño y planeación industrial el obtener el óptimo teórico es menos importante que obtener resultados reales en tiempos razonables.



## **5. Conclusiones y Direcciones Para Trabajo Futuro**

### **5.1 Recopilación**

En esta tesis, un algoritmo para encontrar un calendario de expansiones de una planta batch es propuesto con un modelo multiperíodo MINLP de diseño retrofit como su centro. Se observa que los modelos son combinatoriamente explosivos para problemas de interés real. Para tratar de abordar este problema, una serie de estrategias computacionales se desarrollaron basándose en programación disyunta o modelación lógica, el método de horizonte móvil, y por último un método de desagregación de la demanda. La efectividad del modelo propuesto es ilustrada con diferentes ejemplos.

En este Capítulo se elabora un resumen de la tesis completa, y las conclusiones que se obtuvieron de cada tema visitado. Posteriormente hablamos de las contribuciones de investigación del trabajo al área de diseño de plantas batch y las direcciones en que se puede expandir este trabajo para el futuro.

En este capítulo se elabora un resumen de la tesis completa, y las conclusiones que se obtuvieron de cada tema visitado. Posteriormente hablamos de las contribuciones de investigación del trabajo al área de diseño de plantas batch y las direcciones en que se pudiera expandir este trabajo para el futuro, principalmente enfocándonos al manejo de incertidumbre y las opciones de algoritmos específicos.

## **5.2 Resumen de Tesis**

### **5.2.1 Capítulo 1: Introducción**

En el primer Capítulo de la tesis se describe el problema que estamos tratando de resolver mediante la definición del problema. Adicionalmente se exponen las metas de investigación así como las expectativas y aportaciones de esta tesis doctoral.

### **5.2.2 Capítulo 2: Antecedentes.**

El objetivo del Capítulo 2 es ahondar en los conocimientos generales sobre los diferentes temas y conceptos involucrados con este trabajo.

Se comienza con una descripción general de los procesos batch, y las variables que los caracterizan. Posteriormente, se hace una extensa revisión bibliográfica sobre el diseño de plantas batch con literatura desde los años setenta con el primer modelo propuesto para resolver este problema, hasta artículos escritos en este año en curso.

Se determina el modelo a tomar como base, y se da una descripción de las técnicas computacionales utilizadas para resolver dicho modelo.

Por último definimos el problema industrial que inspira este trabajo, y las suposiciones bajo las cuales se trabajará.

### 5.2.3 Capítulo 3: Modelo multiperíodo de diseño retrofit para plantas batch

Un modelo general multiperíodo entero mixto no lineal (MINLP) es usado para el diseño retrofit de plantas batch. Para mejorar la calidad del modelo y disminuir su esfuerzo computacional se transforma a un modelo multiperíodo de programación disyunta donde se anidan disyunciones para las decisiones de diseño, expansión y operación de la planta. Se utiliza el modelo general propuesto por Van den Heever (2000), y se adapta al modelo de diseño retrofit de plantas batch que se quiere resolver. El modelo de Van den Heever resultó adecuado puesto que incorpora tres niveles de decisión que acoplados al problema de diseño retrofit serían: la selección de la estructura de la red, la operación o desconexión del equipo  $n$  en el período  $t$ , y la expansión del proceso en un período dado.

Este modelo es la base de toda la tesis. Alrededor de él se agregan ecuaciones para manejar los casos específicos de nuestro problema a tratar, en especial el caso de utilizar un mismo equipo donde se juntan dos etapas de producción. Alrededor de este modelo se utiliza una serie de adecuaciones prácticas para tratar de resolver ejemplos de tamaño real y de mejorar sus deficiencias en cuestión de resultados.

La efectividad del modelo propuesto se demuestra a través de dos ejemplos resueltos para horizontes de diferentes tamaños. El primer ejemplo es una versión simplificada donde solo se manejan dos productos. Con los resultados de este ejemplo, se ve la necesidad de ayudar al modelo para realizar sus operaciones de manera más efectiva. Se implementan tres estrategias prácticas sencillas. Primero, explotando el hecho de que el método utiliza una superestructura de red, se colocan cotas superiores a variables continuas de interés. La segunda adecuación práctica es agregar variables de holgura a ecuaciones “difíciles” para el modelo, agilizando así su convergencia.

El segundo ejemplo ya hace uso de estas adecuaciones prácticas mejorando sus tiempos de solución y pudiendo encontrar respuestas para problemas de tamaño considerable. El último ejemplo aumenta los costos fijos de instalación, a manera de poder observar la

opción planteada en nuestro modelo de poder instalar un solo equipo para elaborar dos etapas de producción.

Finalmente, para ilustrar la efectividad de las técnicas aplicadas al modelo, retomamos uno de los ejemplos y lo corrimos sin las adecuaciones propuestas en esta tesis, a manera de poder evaluar el uso de cada una de ellas. Con estas comparaciones queda claro que el tiempo invertido en ellas bien vale la pena generando un modelo matemáticamente más fuerte, además de garantizar el óptimo global.

Se puede ver por los ejemplos de este capítulo que para números de períodos considerables, el algoritmo propuesto en este capítulo pudiera no encontrar una solución en una cantidad de tiempo razonable. Esto motivó la técnica de solución desarrollada en el Capítulo 4.

#### **5.2.4 Capítulo 4: Planeación de la Producción y Horizonte Móvil Incorporados al Modelo Retrofit**

Al modelo del Capítulo 3, se le detectan dos problemas importantes. Primero, el modelo no encuentra soluciones para problemas grandes. Un modelo de diez períodos tarda más de once horas en encontrar solución. Un modelo de 20 corre por cinco días y agota sus recursos. Recordemos que el modelo es planteado para diseño de plantas batch a largo plazo. Lo que el cliente busca es un calendario de expansiones e inversiones. Un algoritmo que sólo pueda resolver un par de períodos no es útil. Para corregir este problema se propone utilizar un algoritmo aproximado de horizonte móvil.

En la industria, la demanda se tiene que cumplir de forma catorcenal, nuestro modelo es de campañas de un solo producto y la demanda se cumple de manera anual. Al escalar la capacidad a catorcenal, el volumen de los tanques no será suficiente, o bien, el problema será infactible. Para corregir este error, se propone un modelo de desagregación, en donde una vez

que se tiene fija la configuración de la planta, se corre un modelo NLP para obtener los valores de las variables continuas como volúmenes, tamaño de batch, etc.

### 5.3 Conclusiones Generales

Los modelos MINLP multiperíodo, para diseño ó incluso para cualquier otra aplicación, rápidamente son combinatoriamente explosivos cuando se requiere tratar un problema realista. La literatura está repleta de algoritmos y modelos, pero rara vez utilizan un problema de tamaño considerable, como los que se presentan en la vida real. Además, los casos multiperíodo para plantas batch no son comunes, encontrando solamente dos referencias. Esto se debe a que principalmente el interés de investigación de diseño ha sido volcado sobre flexibilidad, y no sobre equipos haciendo el mismo trabajo por décadas.

En esta tesis se atacó de manera efectiva el problema del tamaño explosivo de modelos multiperíodos para diseño de plantas batch. Para ello se utiliza un modelo general de programación disyunta que integra diseño y planeación de redes de procesos industriales. Dicho modelo ha sido ajustado a nuestro problema específico.

Se ha demostrado que en efecto se puede utilizar la programación disyunta para la modelación de este problema. Se resuelven una serie de ejemplos para ilustrar la efectividad del modelo así como del método. El método ataca el problema específico de tratar de reducir el tiempo computacional de los MILPs en cada paso de la iteración, dado que por lo general este es el cuello de botella de las computaciones para problemas de optimización multiperíodo.

En el Capítulo 3 presentamos un modelo disyunto para el diseño retrofit de plantas batch. El modelo se basa en el trabajo de Fletcher, Hall y Johns (1991). Se extendió este modelo a multiperíodo, indexando sus variables con el tiempo. Además se añade la variable  $CA_{tk}$ , un costo por ajuste si escoge añadir un equipo en el que se realizarán dos etapas del proceso. Esta decisión lleva una bandera asociada, si la bandera es activada se hace el ajuste

al costo de inversión con la variable  $CA_{tk}$ . Además se hace una corrección al parámetro  $t_{ij}$ , el factor de tiempo asociado a esa etapa, y se ligan las dos etapas. Para fines prácticos, se les coloca a los dos equipos dentro de una caja negra y se corrige su costo de instalación y tiempo relacionado con la etapa.

Al plantear el modelo como un problema de programación disyunta se pretendía que sus tiempos de cómputo fueran más razonables. Al no ser así, se llevaron a cabo dos adecuaciones prácticas en el modelo. Primeramente se le añadió variables de holgura para las ecuaciones relacionadas con la demanda, y con las horas disponibles del horizonte de tiempo. Esto lo que le permite al modelo es no perder tiempo con configuraciones de planta que no sean factibles.

Recuérdese que el modelo se resolvió por el método de aproximación externa (OA) implementado en DICOPT++, se resuelve un problema entero mixto lineal (DLP), para fijar la configuración de la planta, y posteriormente un problema no lineal (PNLP), donde se determinan las variables continuas. No hay certeza de que, con la configuración de planta encontrada en el problema DLP, sea factible realizar la demanda requerida en el tiempo dado. Por lo tanto, el añadir estas variables de holgura es para permitirle al método encontrar las soluciones de estas plantas no factibles y catalogarlas como de costo elevado, en lugar de tener que probar su infactibilidad que resulta más costoso en términos de procesamiento computacional.

En segundo lugar, explotamos el diseño de super-estructura para calcular las cotas superiores de algunas variables como el número de batches o los tiempos de ciclo por producto.

Estas dos adecuaciones ayudaron al modelo de manera notable. En un ejemplo de cinco períodos con cuatro etapas y tres productos donde el tiempo de solución disminuye de

84,790 segundos a 113 segundos; sin embargo, esto prueba ser insuficiente para modelos de más de 10 períodos.

El modelo del Capítulo 3 se topa en la práctica con dos problemas grandes: no encuentra resultados rápidos para el número de períodos de interés al problema y además su programación agrupada anual de producción lo aleja mucho de la realidad. En el Capítulo 4 se hace corrección de estos dos puntos en el problema específico por considerarlos importantes para el cliente.

Para poder resolver hasta 25 períodos de tiempo se utiliza el método de horizonte móvil. La ventaja del algoritmo de horizonte móvil presentado en esta tesis, es que utiliza la información de demandas en los períodos posteriores (no considerados en el horizonte) usando sus variables relajadas. Esto es conveniente puesto que permite al modelo anticipar demandas futuras y usar esa información cuando decide instalaciones y volúmenes de equipos nuevos. Los métodos tradicionales de horizontes móviles, solo van fijando y recorriendo el rango del horizonte en donde se está evaluando el modelo, ignorando la información de períodos de tiempo no pertenecientes al rango.

Las soluciones obtenidas son buenas aproximaciones de la solución detallada, con la ventaja de ser significativamente más rápidos. Adicionalmente, el algoritmo da una cota superior del problema, la cual puede ser utilizada para inicializar algún algoritmo detallado de ramificar y acotar. Si el valor óptimo es de interés para el cliente, se le puede alimentar esta solución encontrada y así mejorar las posibilidades de encontrar solución.

La planeación de la producción puede hacerse de varias formas. Existen muchos modelos cada uno con sus ventajas y desventajas. Aquí se utilizó el modelo de desagregación por que ya se contaba con el modelo, implementar la desagregación de la demanda fue solo

cuestión de cambiar algunos parámetros del problema. Cabe mencionar que al hacer el ajuste a las variables continuas, dejando fijas las variables discretas, se pierde la optimalidad encontrada en el modelo de diseño retrofit.

Se realizaron dos casos con el fin de observar el comportamiento de la solución encontrada por el algoritmo propuesto, el Caso 2 y 3 del Capítulo 4. En el caso 2, se comparan las soluciones obtenidas por el modelo para tres puntos de demanda, demanda +/- 5%. El resultado es una misma calendarización de expansiones para las tres instancias. Para el caso 3, observamos que no existe gran infactibilidad si se modela a tiempos promedio de proceso y luego tratamos la desagrupación de la producción a catorcenas utilizando los límites inferiores de los parámetros de producción en cada etapa. Estos resultados son importantes puesto que, además de probar el modelo en circunstancias diferentes, ayudan a concluir que el algoritmo encuentra buenas soluciones para el problema.

La pérdida de optimalidad por hacer uso del algoritmo de horizontes móviles y posteriormente la corrección a demandas catorcenales, es un sacrificio prudente puesto que en los problemas de diseño y de planeación industrial, el obtener el óptimo teórico es menos importante que obtener resultados reales en tiempos razonables.

#### **5.4 Contribuciones de investigación**

1. Plantear un modelo multiperíodo que represente adecuadamente el problema de expansión de capacidad de plantas batch.
2. Adaptar un algoritmo general de programación disyunta para solucionar un caso específico de un problema de diseño retrofit de plantas batch.



3. Incorporar la posibilidad de manejar diferentes opciones de equipo para una etapa.
4. Incorporar programación disyunta como una herramienta útil para disminuir los tiempos de cómputo del problema maestro (DLP) al solucionar el modelo (MINLP).
5. Utilizar la desagregación de períodos de tiempo para elaborar un modelo de planeación que sirva para corregir y acercar al modelo retrofit a los valores existentes en la realidad.
6. Utilización de estrategias prácticas para reducir los tiempos de cómputo así como lograr obtener soluciones a problemas de largo plazo.
7. Proponer un algoritmo para encontrar soluciones al diseño retrofit multiperíodos para horizontes a largo plazo.

## **5.5 Direcciones para Trabajo Futuro**

### **5.5.1 Programación disyunta.**

El modelo resuelto por programación disyunta fue resuelto en DICOPT++ de GAMS. Éste fue un proceso tedioso y una implementación automática es deseable. Vecchietti (1997) desarrolló un código para este propósito, LOGMIP. Implementar LOGMIP para solucionar el modelo sería de interés. Adicionalmente, no se sabe con prioridad cuales modelos se beneficiarán con el uso de la programación disyunta. El resolver el modelo en el espacio completo del problema, para fines de comparación, también sería deseable para hacer conclusiones sobre la ayuda de la programación disyunta.

Una caracterización teórica sobre este aspecto ha sido presentada por Vecchietti (2000).

### **5.5.2 Integración de planeación y diseño**

En esta tesis manejamos el uso de integración de planeación y diseño haciendo uso de la desagregación de la demanda. Existen muchas formas adicionales de integrar estos dos aspectos del modelo; ahondar en otras posibilidades sería de utilidad.

Una posible dirección de la investigación es utilizar la programación de restricciones, por ejemplo, Van Hentenryck (1989) en donde en el nivel de planeación la factibilidad es más importante que la optimalidad, y en el nivel de diseño la optimalidad es prioritaria.

También se puede explorar el aplicar programación disyunta para la integración de la planeación con el diseño, algo similar a lo que se manejó con el tipo de operación de cada unidad (modo A o B): un nivel de disyunción anidado donde la planeación se lleva a cabo en un nivel inferior y el diseño en un nivel superior.

Existen algoritmos específicos para resolver esta integración; nosotros resolvimos por un método general. Es conveniente tratar de utilizar algún algoritmo específico disponible para comparar los resultados obtenidos por el algoritmo propuesto (Goel, Weijnen y Grievink, 2004; Moreno, Montagna e Iribarren, 2007; Prasad y Maravelias, 2008).

### **5.5.3. Manejo de incertidumbre**

Los modelos que resolvimos en esta tesis asumen que las demandas, costos y disponibilidades son conocidos con certeza. Dada la naturaleza del problema que se nos presentó, el obtener un calendario de expansiones en un horizonte de largo plazo, el manejo de la incertidumbre, sobre todo de la incertidumbre relacionada a la demanda debe ser tomada en cuenta.

En esa tesis, abordamos el problema de reducir el esfuerzo computacional, y lograr mejores resultados integrando la planeación al modelo de diseño. El siguiente paso sería incluir el manejo de incertidumbre de la demanda al modelo, y atender el incremento computacional en el que se incurrirá.

Existen modelos estocásticos para manejar incertidumbre en las demandas y otras variables. Estos modelos incluyen incertidumbre por medio de distribuciones de probabilidades explícitas o por medio del manejo de diferentes escenarios, aunque frecuentemente se requiere de métodos de solución especializados dada su complejidad (Van den Heever, 1999).

Un vasto número de artículos han sido publicados en el área de planeación bajo incertidumbre. Liu y Sahinidis (1996) proponen programación estocástica a dos etapas para el proceso de planeación bajo incertidumbre; primeramente consideran parámetros discretos aleatorios y desarrollan un algoritmo basado en la descomposición de Benders. Irapetritou, Acevedo y Pistokopoulos (1996) proponen un algoritmo de descomposición para un programa estocástico multiperíodo. Clay y Grossmann (1997) consideran problemas de planeación con incertidumbre en las demandas y en los coeficientes de costos. También proponen un algoritmo iterativo de agregación/desagregación para la solución del modelo estocástico de dos etapas. Ahmed y Sahinidis (1998) proponen un método lineal para incorporar robustez a las decisiones de la segunda etapa, para modelos de planeación bajo la incertidumbre, presentando una heurística para su solución.

Algún progreso se ha encontrado en los modelos de diseño y planeación con manejo de incertidumbre. Petkov y Maranas (1997) extendieron el modelo existente de planeación y diseño originalmente propuesto por Birewar y Grossmann (1990) a incorporar incertidumbre en las demandas.

Un modelo jerárquico basado en escenarios para incorporar incertidumbre en el nivel de planeación para una planta multi-producto fue propuesto por Sand et al. (2000).

Mientras los métodos mencionados muestran avances en el manejo de la incertidumbre en el modelo de diseño-planeación a gran escala, las caracterizaciones de la incertidumbre y el desarrollo de algoritmos de solución eficientes siguen siendo complejos y difíciles de estructurar.

#### **5.5.4. Incluir gastos de operación**

En este trabajo no consideramos costos de operación, solamente los de inversión. Es necesario revisar los costos de operación contra los de instalación. Esto nos lleva a eliminar la posibilidad de analizar si el comprar una tecnología más cara, pero con menores costos de operación, pudiera ser una mejor inversión. Este es un caso típico en la mayoría de las plantas, y el análisis de costo vs. tecnología se debe hacer para encontrar su curva de equilibrio.

#### **5.5.5. Optimización de Superestructura**

En el área de diseño y síntesis de procesos frecuentemente aparece el problema de postular una superestructura de alternativas para modelar matemáticamente un problema (Yeomans y Grossmann, 2000).

En nuestro caso, tenemos una superestructura de planta, en la cual todas las posibles alternativas están incluidas. Si se trabaja con una superestructura muy grande, el tamaño del modelo crece y su modelación se entorpece. Si se escoge una demasiado escueta, podrían estarse eliminado mejores soluciones (Mussati et al. 2008). Para obtener una superestructura de manera sistemática, Friedler et al. (1993) propusieron un método basado en teoría de grafos que tiene complejidad polinomial. Este proceso ha sido aplicado con éxito para la elaboración de redes de proceso minimizando desperdicios (Friedler, Kovacs y Fan, 1995). Recomendaríamos empezar el algoritmo con alguna metodología eficiente para definir la superestructura inicial.



## **BIBLIOGRAFÍA**

C.S. Adjiman, I.P. Androulakis y C.A. Floudas (2000). Global optimization of mixed-integer nonlinear problems. *AIChE Journal*, 46(9):1769-1797.

S. Ahmed y N.V. Sahinidis (1998). Robust process planning under uncertainty. *Industrial Engineering and Chemical Research*. 37(5):1883-1892.

E. Balas (1985). Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic and Discrete Methods*. 6(3):466-486.

A.P. Barbosa-Póvoa (2007). A critical review on the design and retrofit of batch plants. *Computers and Chemical Engineering*, 31(7):833-855.

M.H. Bassett, J.F. Pekny y G.V. Reklaitis (1996). Decomposition techniques for the solution of large-scale scheduling problems. *AIChE Journal*, 42(3):3373-3384.

P. Beraldi, G. Ghiaji, A. Grieco y E. Guerriero (2008). Rolling-horizon and fix-and-relax heuristics for the parallel machine lot sizing and scheduling problema with sequense dependent set-up costs. *Computers and Operations Research*, 35(11):3644-3656.

L. Bernal-Haro, C. Azzaro-Pantel, L. Pibouleau y S. Doménech (2002). Multiobjective batch plant design: a two-stage methodology 1. Development of a Design-Oriented Discrete-Event Simulation Model. *Industrial Engineering and Chemical Research*, 41(23):5727-5742.

L.T. Biegler, I.E. Grossmann y A.W. Westerberg (1997). *Systematic Methods of Chemical Process Design*, A.W. Prentice-Hall, Old Tappan, EUA

D.B. Birewar e I.E. Grossmann (1990). Simultaneous production planning and scheduling of multiproduct batch plants. *Industrial Engineering and Chemistry Research*, 29(4):570-580.

G.R. Bitran y A.C. Hax (1977). On the design of hierarchical production planning systems. *Decision Sciences*, 8(1):28-55.

J.K. Bok, I.E. Grossmann y S. Park (2000). Supply chain optimization in continuous flexible process networks. *Industrial and Engineering Chemistry Research*, 39(55):1279-1290.

A. Brooke, D. Kendrick y A. Meeraus (1992). *GAMS: A User's Guide, Release 2.25*. The Scientific Press; San Francisco, EUA.

M.R. Bussieck y A. Pruessner (2003). *Mixed-integer Nonlinear Programming*. [www.gamsworld.org/minlp/siagopt.pdf](http://www.gamsworld.org/minlp/siagopt.pdf).

J.C. Cabada (2005). Proceso Cerveza 090804. Documento Interno de Cervecería Cuauhtémoc-Moctezuma. San Nicolás de los Garza, México..

M. Cardoso, R.L. Salcedo y S. Feyeo de Azevedo (1997). A simulated annealing approach to the solution of MINLP problems. *Computers and Chemical Engineering*, 21(12):1349-1364.

A.L. Cavin, U. Fischer y K. Hungerbuhler (2008). Multiobjective optimization of multipurpose batch plants using superequipment class concept. *Computers and Chemical Engineering*, 32(3):512-529.

S. Chand, V. Ning Hsu y S. Sethi (2004). Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, 4(1):25-43.

R.L. Clay e I.E. Grossmann, (1997). A disaggregation algorithm for the optimization of stochastic programming models. *Computers and Chemical Engineering*, 21(7):751-774



M.M. Daichendt e I.E. Grossmann (1998). Integration of hierarchical decomposition and mathematical programming for the synthesis of process flowsheets. *Computers and Chemical Engineering*, 22(1-2):147-175.

S. Dedieu, L. Pibouleau, C. Azzaro-Pantel y S. Domenech (2003). Design and retrofit of multiobjective batch plants via multicriteria genetic algorithm. *Computers and Chemical Engineering*, 27(12):1723-1740.

I.T. Dedopoulos y N. Shah (1996). Long-term maintenance policy optimization in multipurpose process plants. *Chemical Engineering Research and Design*, 74(A3):307-320.

A. Dietz, C. Azzaro-Pantel, L. Pibouleau y S. Doménech (2008). Strategies for multiobjective genetic algorithm development: Application to optimal batch plant design in process systems engineering. *Computers and Industrial Engineering*, 54(3):539-569.

A.V. Dimitriadis, N. Shah y C.C. Pantelides (1997). RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Computers and Chemical Engineering*, 21(s1):s1061-s1066.

M.A. Duran e I.E. Grossmann (1986). An outer approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*. 36(3):307-339.

T.W.G. Epperly, M.G. Ierapetritou y E.N. Pistikopoulos (1997). On the global and efficient solution of stochastic batch plant design problems. *Computers and Chemical Engineering*, 21(12):1411-1431.

M. Erdirik-Dogan (2007). *Integration of Planning and Scheduling of Batch and Continuous Process Systems*. Disertación Doctoral, Universidad de Carnegie Mellon, Pittsburgh, EUA.

M. Erdirik-Dogan e I.E. Grossmann (2007). A decomposition method for the simultaneous planning and scheduling of single-stage continuous multiproduct plants. *Industrial and Engineering Chemistry Research*, 46(15):5250.

A. Espuña y L. Puigjaner (1989). On the solution of the retrofitting problem for multiproduct batch semicontinuous chemical-plants. *Computers and Chemical Engineering*, 13(4-5):483-490.

R.M. Felder y R.W. Rosseau (2000) *Elementary Principles of Chemical Processes*. Wiley & Sons, New York, EUA.

R. Fletcher, J.A.J. Hall y W.R. Johns (1991) Flexible retrofit design of multiproduct batch plants. *Computers and Chemical Engineering*, 15(12):843-852.

C.A. Floudas y X. Lin (2004). Continuous-time versus discrete time approaches for scheduling of a chemical process: A review. *Computers and Chemical Engineering*, 28(11):2109-2129.

F. Friedler, K. Tarjan, Y.W. Huang y L.T. Fan (1993). Graph-theoretic approach to process synthesis: Polynomial algorithm for maximal structure generation. *Computers and Chemical Engineering*, 17(9):929-942.

F. Friedler, Z. Kovacs y L. T. Fan, (1995). Parametric study of separation network synthesis: Extreme properties of optimal structures. *Computers and Chemical Engineering*, 19(1):107-112.

GAMS Development Corporation (2008). *GAMS-The Solver Manuals*. Washington, D.C.

M.R. Garey y D.R. Johnson (1979). *Computers and Intractability: A guide to the theory of NP-completeness*. W.H. Freeman. New York, EUA.

- A.M. Geoffrion (1972). Generalized Benders decomposition. *Journal of Optimization Theory and Applications*. 10(4):237-260.
- M.C. Georgiadis, G.E. Rotstein y S. Macchietto (1997). Optimal layout design in multipurpose batch plants. *Industrial and Engineering Chemistry Research*, 36(11):4852-4863.
- H.D. Goel, M.P.C. Weijnen y J. Grievink (2004). Optimal reliable retrofit design of multiproduct batch plants. *Industrial and Engineering Chemistry Research*. 43(14):3799-3811.
- I.E. Grossmann I.E. y Z. Kravanja (1995). Mixed-integer nonlinear programming techniques for process systems engineering. *Computers and Chemical Engineering*, 19(s1):189-204.
- I.E. Grossmann I.E. y Z. Kravanja (1997). Mixed-integer nonlinear programming: a survey of algorithms and applications. En L.T. Biegler (editor) *Large-Scale Optimization with Applications, Part II: Optimal Design and Control*, pp. 73-100. Springer-Verlag, Berlin, Alemania.
- P.V. Hentenryck (1989). Constraint satisfaction in logic programming: Logic programming Series. MIT Press, Cambridge, EUA.
- J.N. Hooker (2000). Logic-based methods for optimization. Manuscrito no publicado, Universidad de Carnegie Mellon, Pittsburgh, EUA.
- J.N. Hooker y M.A. Osorio (1999) Mixed logical linear programming. *Discrete Applied Mathematics*. 96-97:395-442.
- M.G. Ierapetritou, J. Acevedo y E.N. Pistokopoulos (1996). An optimization approach for process engineering problems under uncertainty. *Computers and Chemical Engineering*, 20(6-7):703-709.

- R.R. Iyer e I.E. Grossmann (1998). A bilevel decomposition algorithm for long-range planning of process networks. *Industrial Engineering Chemical Research*, 37(2):474-481.
- I. A. Karimi y A. Modi (1989). Design of multiproduct batch processes with finite intermediate storage. *Computers and Chemical Engineering*, 13(1-2):127-139.
- P. Kesavan y P.I. Barton (2000). Generalized branch-and-cut framework for mixed-integer nonlinear optimization programs. *Computers and Chemical Engineering*, 24(2-7):1361-1366.
- S. Lee e I.E. Grossmann (2000). New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering*, 24(9-10):2125-2141.
- S. Lee e I.E. Grossmann (2001). A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. *Computers and Chemical Engineering*, 25(11-12):1675-1697.
- S. Lee e I.E. Grossmann (2003). Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: Applications to process networks. *Computers and Chemical Engineering*, 27(11):1557-1575.
- X. Lin y C.A. Floudas (2001). Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Computers and Chemical Engineering*, 25(4-6):655-674.
- M.L. Liu, y N.V. Sahinidis (1996). Optimization in process planning under uncertainty. *Industrial and Engineering Chemistry Research*, 35(11):4154-4165.
- S. Macchietto (2005). Integrated Batch Processing, A model for advanced manufacturing. En Proceedings of the 5005 *APACT Conference*, pp. 20-22. Birmingham, Reino Unido.

J.T. McCabe (1999). *The Practical Brewer*. Master Brewers Association of the Americas, 3a edición, St. Paul, EUA.

M. McDonald e I.A. Karimi (1997). Planning and scheduling of parallel semicontinuous processes. 1. Production planning. *Industrial and Engineering Chemistry Research*, 36(7):2691-2700.

J.M. Montagna (2003). Optimal retrofit of multiproduct batch plants. *Computers and Chemical Engineering*, 27(8-9):1277-1290.

M.S. Moreno, J.M. Montagna y O.A. Iribarren (2007). Multiperiod optimization for the design and planning of multiproduct batch plants. *Computers and Chemical Engineering*, 31(9):1159-1173.

S.F. Mussati, M. Barttfeld, P.A. Aguirre y N.J. Scenna (2008). A disjunctive programming model for superstructure optimization of power and desalting plants. *Desalination*, 222(1-3):457-465.

C.C. Pantelides (1994). Unified frameworks for the optimal process planning and scheduling. En *Proceedings of the Second Conference of Foundations of Computer Aided Operations (FOCAPO)*, pp. 253-274. Snowmass, EUA..

K.P Papalexandri y E.N. Pistikopoulos (1994). A multiperiod MINLP model for the synthesis of flexible heat and mass exchange networks. *Computers and Chemical Engineering*, 18(11-12):1125-1139.

G.E. Paules y C.A. Floudas (1992). Stochastic programming in process synthesis: A two-stage model with MINLP recourse for multiperiod heat-integrated distillation sequences. *Computers and Chemical Engineering*, 16 (3):189-210.

- S.B. Petkov y C.D. Maranas (1998). Design of multiproduct batch plants under demand uncertainty with staged capacity expansions. *Computers and Chemical Engineering*, 22(s1):s789-s792.
- S.B. Petkov y C.D. Maranas (1997). Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty. *Industrial and Engineering Chemistry Research*, 36(11):4864-4881.
- T. Pinto, F.D. Barbosa-Póvoa y A.Q. Novais (2005). Optimal design of batch plants with a periodic mode of operation. *Computers and Chemical Engineering*, 29(6):1293-1303.
- E. N. Pistikopoulos, A.M. Thomaidis y M.G. Ierapetritou (1996). Flexibility, reliability and maintenance considerations in batch plant design under uncertainty. *Computers and Chemical Engineering*, 20(s2):s1209-s1214.
- O. Pourali, M. Amidpour y D. Rashtchian (2006). Time decomposition in batch process integration. *Chemical Engineering and Processing*, 45(1):14-21.
- P. Prasad y C. Maravelias (2008). Batch selection, assignment and sequencing in multistage multi-product process. *Computers and Chemical Engineering*, 32(6):1106-1119.
- L. Puigjaner y G. Guillen-Gosálbez (2008). Towards an integrated framework for supply chain management in the batch process industry. *Computers and Chemical Engineering*, 32(4-5):650-670.
- I. Quesada e I.E. Grossmann (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16 (10-11): 937-947.
- R. Raman e I.E. Grossmann (1994). Modeling and computational techniques for logic based integer programming. *Computers and Chemical Engineering*, 18(7):563-578.

D.E. Ravemark y D.W. Rippin (1998). Optimal design of multiproduct batch plant. *Computers and Chemical Engineering*, 22(1-2):177-183.

A. Reeve (1995). Batch processing: A recipe for success? *Process Engineering*, 76(4): 49-52.

G.V. Reklaitis (1989). Progress and issues in computer aided batch process design. En *Proceedings of the Third International Conference on Foundations of Computer-Aided Process Design (FOCAPD)* pp. 241-246. Snowmass, EUA.

G.E. Rotstein, R. Lavie y D. R. Lewin (1996). Synthesis of flexible and reliable short-term batch production plans. *Computers and Chemical Engineering*, 20(2): 201-215.

N.V. Sahinidis, I.E Grossmann, R.E Fornari y M. Chathrathi (1989). Optimization model for long range planning in the chemical industry. *Computers and Chemical Engineering*. 13(9):1049-1063.

N.V. Sahinidis y M. Liu (1996). Long range planning in the process industries: A projection approach. *Computational Operational Research*. 23(3):237-253.

G. Sand, S. Engell, A. Markert, R. Schultz y C. Schulz (2000). Approximation of an ideal online scheduler for a multiproduct batch plant. *Computers and Chemical Engineering*. 24(2-5):361-367.

R.W. Sargent y I.E Grossmann (1979). Optimal design of multipurpose chemical plants. *Industrial Engineering and Chemical Process Design and Development*. 18: 343-348.

N. Shah (1998). Single- and multisite planning and scheduling: Current status and future challenges. En J.F. Penky y G. E. Blau (editores), *Proceedings of the Third International Conference on Foundations of Computer-aided Process Operations (FOCAPO)*, pp. 75-90. Snowbird, USA.

- N. Shah y C.C. Pantelides (1991). Optimal long-term campaign planning and design of batch operations. *Industrial and Engineering Chemistry Research*, 30(10):2308-2321.
- E.M.B. Smith y C.C. Pantelides (1999). A symbolic reformulation/spatial branch and bound algorithm for the global optimisation of nonconvex MINLPs. *Computers and Chemical Engineering*, 23(4-5):457-478.
- R. Sousa, N. Shah y L.G. Papageorgiou (2008). Supply chain design and multilevel planning: An industrial case. *Computers and Chemical Engineering*, 32(11):2643-2663.
- H. Stefansson, N. Shah y P. Jansson (2006). Multiscale planning and scheduling in the secondary pharmaceutical industry. *AIChE Journal*, 52(12):4133-4149.
- D.A. Straub e I.E. Grossmann (1992). Evaluation and optimization of stochastic flexibility in multiproduct batch plants. *Computers and Chemical Engineering*, 16(2): 69-87
- S. Subrahmanyam, J.F. Pekny y G.V. Reklaitis (1996). Decomposition approaches to batch plant design and planning. *Industrial and Engineering Chemistry Research*, 35(6):1866-1876.
- C. Sung y C. T. Maravelias (2006). An attainable region approach for effective production planning. *Computer Aided Chemical Engineering*, 21(2):1893-1898.
- M. Tawarmalani y N.V. Sahinidis (2002). Global optimization of mixed integer nonlinear programs. En H. E. Romeijn y P. M. Pardalos (editores), *Handbook of Global Optimization*, Vol. 2, pp 65-86. Kluwer Academic Publishers, Boston, EUA.
- M. Turkay e I.E. Grossmann (1996a). Disjunctive programming techniques for the optimization of process systems with discontinuous investment costs—multiple size regions. *Industrial and Engineering Chemistry Research*, 35(8): 2611-2623.



- M. Turkay e I.E. Grossmann (1996b). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering*, 20(8):959-978.
- S. A. Van den Heever e I.E Grossmann (1999). Disjunctive multiperiod optimization methods for design and planning of chemical process systems. *Computers and Chemical Engineering*, 23(8):1075-1095.
- S. A. Van den Heever, I.E. Grossmann y S. Vasantharanjan. (2000). Integrating complex economic objective with the design and planning of offshore oilfield infrastructures. *Computers and Chemical Engineering*, 24(2-7):1049-1055.
- D.K. Varvarezos, I.E. Grossmann y L.T. Biegler (1992). An outer-approximation method for multiperiod design optimization. . *Industrial and Engineering Chemistry Research*, 31(6):1466-1477.
- J.A. Vaselenak, I.E. Grossmann y A.W. Westerberg (1987). Optimal retrofit design in multiproduct batch plants. *Industrial and Engineering Chemistry Research*, 26(4):718-726.
- A. Vecchietti e I.E. Grossmann (1997). LOGMIP: A discrete continuous nonlinear optimizer. *Computers and Chemical Engineering*, 21(s1):S427-S432.
- A. Vecchietti e I.E. Grossmann (1999). LOGMIP: A disjunctive 0-1 nonlinear optimizer for process systems models. *Computers and Chemical Engineering*, 23(4-5):555-565.
- A. Vecchietti e I.E. Grossmann (2000). Modeling issues and implementation of language for disjunctive programming. *Computers and Chemical Engineering*. 24(9-10):2143-2155.
- A. Vecchietti y J.M. Montagna (1998). Alternatives in the optimal allocation of intermediate storage tank in multiproduct batch plants. *Computers and Chemical Engineering*, 22(s1): S801-S804

J. Viswanathan e I.E. Grossmann (1990). A combined penalty function and outer-approximation method for MINLP optimization. *Computers and Chemical Engineering*, 14(7):769-782.

V.T. Voudourins e I.E. Grossmann (1996). MILP model for scheduling and design of a special class of multipurpose batch plants. *Computers and Chemical Engineering*, 20(11):1335-1360.

S.J. Wilkinson, N. Shah y C.C.Pantelides (1996). Aggregate modeling of multipurpose plant operation. *Computers and Chemical Engineering*, 19(1):583-588.

D. Wu y M. Ierapetritou (2007) Hierarchical approach for production planning and scheduling under uncertainty. *Chemical Engineering and Processing: Process Intensification*, 46(11):1129-1140

H. Yeomans e I.E. Grossmann (2000). Disjunctive programming models for the optimal design of distillation columns and separation sequences. *Industrial and Engineering Chemistry Research*, 39(6):1637-1648.

D.J. Yoo, H.K. Lee, J.H. Ryu e I.B. Lee (1999). Generalized retrofit design of multiproduct batch plants. *Computers and Chemical Engineering*, 23(6):683-695.

## APÉNDICE

### Apéndice A: Derivación para el envolvente convexo del problema (DMILP)

Raman y Grossmann (1994) mostraron como convertir un programa lineal disyunto a programas entero-mixtos a través de la formulación del envolvente convexo para cada disyunción, basados en el trabajo previo por Balas (1985). Con esas ideas, Van den Heever, 2001 convierte el problema (DLP) al problema (DMILP) por la formulación del envolvente convexo para cada disyunción. La idea básica es remplazar cada variable booleana por variables binarias correspondientes y desagregar las variables continuas para tener una variable para cada disyunción. Como se puede ver a continuación, una gran cantidad de las variables desagregadas se convierten en redundantes y se pueden remover para simplificar el modelo. Considerar al problema general (DLP):

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

$$g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \leq 0 \quad \forall t$$

$$Dx_t \leq 0 \quad \forall t$$

$$\left[ \begin{array}{c} y_j \\ w_{jt} \\ h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T (x_{t,nz} - x_{t,nz}^k) + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T (x_{t,z} - x_{t,z}^k) \leq a \\ Ax_{t,nz} + Bx_{t,z} + CQ_{jt} \leq b \\ k \in K^j \\ CO_{jt} = \gamma_{jt} \\ \left[ \begin{array}{c} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array} \right] \\ \vee \left[ \begin{array}{c} -w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array} \right] \forall t \\ \vee \left[ \begin{array}{c} -y_j \\ B^{jt} x_t = 0 \\ \forall t \end{array} \right] \end{array} \right] \quad \forall j$$

$$y_j \Rightarrow \bigvee_{t=1}^T w_{jt} \quad \forall j \qquad w_{jt} \Rightarrow y_j \quad \forall j, t$$

$$w_{jt} \Rightarrow \bigvee_{\tau=1}^t z_{j\tau} \quad \forall j, t \qquad z_{j\tau} \Rightarrow w_{j\tau} \quad \forall j, t$$

$$\Omega(y) = \text{Verdadero}$$

$$CE, CO, Q, QE, x \geq 0, \quad y, w, z \in \{\text{Verdadero}, \text{Falso}\}$$

Para convertir cada disyunción a su forma entera-mixta, por la formulación del envolvente convexo, primero se convierte la disyunción interna, y se trabaja de adentro hacia afuera en cada disyunción hasta que todo el problema se haya transformado a un problema MILP. Por simplicidad se ignorarán la mayoría de los sub y super índices por el resto de la derivación, con excepción de los que sean considerados relevantes. Primero, considerar la disyunción interna:

$$\left[ \begin{array}{c} z_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jt} \\ CE_{jt} = 0 \\ QE_{jt} = 0 \end{array} \right]$$

(A1)

Las ecuaciones de expansión pueden ser movidas afuera de la disyunción para simplificar la derivación del envolvente convexo, dado que siempre se aplican si el proceso es escogido. El primer paso es el desagregar las variables. El superíndice 1 se refiere al lado izquierdo de la

disyunción mientras el superíndice dos se refiere al lado derecho. Una variable binaria es asignada a cada disyunción. La desagregación es entonteces como sigue:

$$\begin{aligned} z^1 + z^2 &= 1 \\ QE &= QE^1 + QE^2 \\ CE &= CE^1 + CE^2 \end{aligned} \quad (A2)$$

Substituyendo variables desagregadas a la disyunción 1:

$$CE^1 = \alpha QE^1 + \beta z^1 \quad (A3)$$

$$0z^1 \leq QE^1 \leq Uz^1 \quad (A4)$$

Substituyendo variables desagregadas a la disyunción 2:

$$CE^2 = 0z^2 = 0 \quad (A5)$$

$$QE^2 = 0z^2 = 0 \quad (A6)$$

Simplificando:

$$CE = \alpha QE + \beta z$$

$$QE \leq Uz$$

$$QE, CE \geq 0$$

$$z \in \{0,1\}$$

$$(A7)$$

Ahora considerar la disyunción en medio de (DLP) después de reemplazar la disyunción interna con (A7):

$$\left[ \begin{array}{c} w_{jt} \\ h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T (x_{t,nz} - x_{t,nz}^k) + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T (x_{t,z} - x_{t,z}^k) \leq a \\ Ax_{t,nz} + Bx_{t,z} + CQ_{jt} \leq b \\ k \in K^j \\ CO_{jt} = \gamma_{jt} \end{array} \right] \vee \left[ \begin{array}{c} \neg w_{jt} \\ B^j x_{t,z} = 0 \\ CO_{jt} = 0 \end{array} \right]$$

$$(A8)$$

Los mismos costos y variables continuas también se van a cero en el lado derecho de la disyunción como los de (A7), pero se dejaron fuera por simplicidad. Después de un procedimiento similar al aplicado para la derivación de (A7), específicamente la

desagregación de las variables continuas, la sustitución y simplificación, el siguiente envolvente convexo entero-mixto es obtenido:

$$\begin{aligned}
 x_{t,nz} &= x_{t,nz}^1 + x_{t,nz}^2 \\
 Q_{jt} &= Q_{jt}^1 + Q_{jt}^2 \\
 \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^1 + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z} &\leq \left[ -h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^k + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z}^k + a \right] w_{jt} \\
 Ax_{t,nz}^1 + Bx_{t,z} + CQ_{jt}^1 &\leq bw_{jt} \\
 CO_{jt} &= \gamma_{jt} w_{jt} \\
 CE_{jt} &= \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} \\
 QE_{jt} &\leq Uz_{jt} \\
 x_{t,z} &\leq Uw_{jt} \\
 x_{t,nz}^1 &\leq Uw_{jt} \\
 Q_{jt}^1 &\leq Uw_{jt} \\
 x_{t,nz}^2 &\leq U(1 - w_{jt}) \\
 Q_{jt}^2 &\leq U(1 - w_{jt}) \\
 z_{jt} &\leq w_{jt} \\
 x, Q, QE, CE, CO &\geq 0, \quad z, w \in \{0,1\}
 \end{aligned} \tag{A}$$

9)

Finalmente, considerar la disyunción externa en (DLP) con (A9):

$$\left[ \begin{array}{c} y_j \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ (A9) \end{array} \right] \vee \left[ \begin{array}{c} \neg y_j \\ B^{jt} x_t = 0 \\ Q_{jt} = 0 \end{array} \right]$$

Aunque no se demuestra anteriormente, las mismas variables son fijadas a cero del lado derecho de la disyunción tal como en la formulación de (A9), pero ellas son fijadas a cero para todos los períodos de tiempo. Después de la desagregación de variables, sustitución y simplificación, las restricciones afuera de las disyunciones son añadidas y las proposiciones lógicas son expresadas de manera algebraica para obtener la formulación (MILP) final, el modelo (DMILP):

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

sujeto a:

$$\begin{aligned}
 & g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \leq 0 \quad \forall t \\
 & Dx_t \leq d \quad \forall t \\
 & \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^1 + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z} \leq - \left[ h_{jt}^{nl}(x_t^k) - \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^k + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z}^k - a \right] w_{jt} \\
 & \quad \forall j, t, k \in K^j \\
 & Ax_{t,nz}^1 + Bx_{t,z} + CQ_{jt}^1 \leq bw_{jt} \quad \forall j, t \\
 & x_{t,nz} = x_{t,nz}^1 + x_{t,nz}^2 \quad \forall t \\
 & x_{t,nz}^1 \leq Uw_{jt} \quad \forall j, t \\
 & x_{t,nz}^2 \leq U(1 - w_{jt}) \quad \forall j, t \\
 & x_{t,z} \leq Uw_{jt} \quad \forall j, t \\
 & Q_{jt} = Q_{jt}^1 + Q_{jt}^2 \quad \forall j, t \\
 & Q_{jt}^1 \leq Uw_{jt} \quad \forall j, t \\
 & Q_{jt}^2 \leq U(1 - w_{jt}) \quad \forall j, t \\
 & CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} \quad \forall j, t \\
 & CO_{jt} = \gamma_{jt} w_{jt} \quad \forall j, t \\
 & Q_{jt} = Q_{j,t-1} + QE_{jt} \quad \forall j, t \\
 & QE_{jt} - Uz_{jt} \leq 0 \quad \forall j, t \\
 & y_j \leq \sum_t w_{jt} \quad \forall j \\
 & w_{jt} \leq y_j \quad \forall j, t \\
 & w_{jt} \leq \sum_{\tau=1}^t z_{j\tau} \quad \forall j, t \\
 & z_{jt} \leq w_{jt} \quad \forall j, t \\
 & Ey \leq e \\
 & CE, CO, Q, QE, x \geq 0, \quad y, w, z \in \{0,1\}
 \end{aligned}$$

**Apéndice B: Transformación Exponencial para Convexificación del modelo (Vaselenak, 1987).**

El modelo de retrofit para plantas batch, sin el uso de la transformación exponencial utilizada en el Capítulo 2 es el siguiente:

xii) Función Objetivo:

$$\min \sum_t \sum_j \sum_k CE_{ijk} - \sum_t \sum_k CA_{tk}$$

xiii) Meta de producción:

$$n_{it} B_{it} \leq Q_{it} \quad \forall i, t$$

xiv) Tiempo de ciclo limitante del producto  $i$ :

$$\frac{t_{ij}}{T_{Lit}} - \sum_k w_{ijkt}^C \leq N_j^{old} \quad \forall i, j, t$$

xv) Tiempo de producción anual:

$$\sum_i n_{it} T_{Lit} \leq H_t \quad \forall t$$

xvi) Cota en el numero total de unidades nuevas:

$$\sum_j \sum_k y_{jk} \leq Z^U$$

xvii) Restricciones de capacidad Opción B:

$$\sum_k V_{ijkmt}^B + (V_j^{old})_m \geq S_{ij} B_{it} \quad \forall i, j, m, t$$

xviii) Asignación única de unidades nuevas:

$$y_{jk} \geq y_{j,k+1} \quad \forall j, k = 1 \dots Z_j - 1$$

xix) Costo de ajuste por escoger dos etapas en un equipo

$$CA_{kt} = c_{2t} \quad \forall e_{2kt} = e_{3kt}$$



xx) Disyunción de cada unidad  $k$  añadida a la etapa  $j$ :

$$\left[ \begin{array}{c} y_{jk} \\ V_{jkt} \leq V_j^U \\ V_{jkt} = V_{jk,t-1} + e_{jkt} \\ w_{jkt} \\ V_{jkt} \geq V_j^L \\ \left[ \begin{array}{c} w_{ijk1t}^B \\ V_{ijk1t}^B \leq V_{jkt} \\ V_{ijk1t}^B \leq V_j^U \end{array} \right] \vee \dots \vee \left[ \begin{array}{c} w_{ijkmt}^B \\ V_{ijkmt}^B \leq V_{jkt} \\ V_{ijkmt}^B \leq V_j^U \end{array} \right] \vee \left[ \begin{array}{c} w_{ijkt}^C \\ V_{ijkt}^C \leq V_{jkt} \\ V_{ijkt}^C \leq V_j^U \\ V_{ijkt}^C \geq B_{it} S_{ij} \end{array} \right] \forall i \\ \left[ \begin{array}{c} z_{jkt} \\ CE_{jkt} = K_{jt} + c_{jt} e_{jkt} \\ V_j^L \leq e_{jkt} \leq V_j^U \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jkt} \\ CE_{jkt} = 0 \\ e_{jkt} = 0 \end{array} \right] \\ \left[ \begin{array}{c} \neg w_{jkt} \\ V_{jkt} \geq 0 \end{array} \right] \forall t \end{array} \right] \vee \left[ \begin{array}{c} \neg y_{jk} \\ V_{jkt} = 0 \end{array} \right] \forall j, k$$

xxi) Ecuaciones Lógicas:

$$\begin{aligned} \sum_m w_{ijkmt}^B + w_{ijkt}^C &= w_{jkt} \quad \forall i, j, k, t \\ \sum_t z_{jkt} &= y_{jk} \quad \forall j, k \\ y_{jk} &\geq \sum_t w_{jkt} \quad \forall j, k \\ w_{jkt} &\leq y_{jk} \quad \forall j, k, t \\ w_{jkt} &\leq \sum_{\tau=1}^t z_{jk\tau} \quad \forall j, k, t \\ z_{jkt} &\leq w_{jkt} \quad \forall j, k, t \end{aligned}$$

xxii) Variables:

$$\begin{aligned} n_{it}, B_{it}, T_{Lit}, V_{jk}, e_{jkt}, V_{jkt}, V_{ijkmt}^B, V_{ijkt}^C &\geq 0 \quad y_{jk}, w_{jkt}, w_{ijkmt}^B, w_{ijkt}^C, z_{jkt} = \{\text{Verdadero}, \text{Falso}\} \\ i &= 1 \dots N \quad j = 1 \dots N \quad t = 1 \dots T \quad k = 1 \dots Z_j \quad m = 1 \dots N_j^{old} \end{aligned}$$

Se definen las siguientes variables para aplicar transformación exponencial y convexificar el modelo:

$$x1_{it} = \ln n_{it},$$

$$x2_{it} = \ln B_{it},$$

$$x3_{it} = \ln T_{Lit}.$$

Substituyendo en las ecuaciones ii), iii) y iv) queda:

ii) Meta de producción:

$$n_{it} B_{it} \leq Q_{it} \quad \forall i, t$$

$$\ln n_{it} + \ln B_{it} \leq \ln Q_{it} \quad \forall i, t$$

$$x1_{it} + x2_{it} \leq \ln Q_{it} \quad \forall i, t$$

iii) Tiempo de ciclo limitante del producto  $i$ :

$$\frac{t_{ij}}{T_{Lit}} - \sum_k w_{ijkt}^C \leq N_j^{old} \quad \forall i, j, t$$

$$N_j^{old} + \sum_k w_{ijkt}^C \geq \frac{t_{ij}}{T_{Lit}} \quad \forall i, j, t$$

$$N_j^{old} + \sum_k w_{ijkt}^C \geq t_{ij} (T_{Lit})^{-1} \quad \forall i, j, t$$

$$N_j^{old} + \sum_k w_{ijkt}^C \geq t_{ij} \exp(-\ln T_{Lit}) \quad \forall i, j, t$$

$$N_j^{old} + \sum_k w_{ijkt}^C \geq t_{ij} \exp(-x3_{it}) \quad \forall i, j, t$$

iv) Tiempo de producción anual:

$$\sum_i n_{it} T_{Lit} \leq H_t \quad \forall t$$

$$\sum_i \exp(\ln(n_{it} T_{Lit})) \leq H_t \quad \forall t$$

$$\sum_i \exp(\ln n_{it} + \ln T_{Lit}) \leq H_t \quad \forall t$$

$$\sum_i \exp(x1_{it} + x3_{it}) \leq H_t \quad \forall t$$

La formulación multiperíodo obtenida es, a pesar de seguir siendo no-lineal, convexa, y por ello podemos asegurar un óptimo global, quedando finalmente el modelo presentado en el Capítulo 2:

i) Función Objetivo:

$$\min \sum_t \sum_j \sum_k CE_{ijk} - \sum_t \sum_k CA_{tk}$$

ii) Meta de producción:

$$x1_{it} + x2_{it} \leq \ln Q_{it} \quad \forall i, t$$

iii) Tiempo de ciclo limitante del producto  $i$ :

$$N_j^{old} + \sum_k w_{ijkt}^C \geq t_{ij} \exp(-x3_{it}) \quad \forall i, j, t$$

iv) Tiempo de producción anual:

$$\sum_i \exp(x1_{it} + x3_{it}) \leq H_t \quad \forall t$$

v) Cota en el numero total de unidades nuevas:

$$\sum_j \sum_k y_{jk} \leq Z^U$$

vi) Restricciones de capacidad Opción B:

$$\sum_k V_{ijkmt}^B + (V_j^{old})_m \geq S_{ij} B_{it} \quad \forall i, j, m, t$$

vii) Asignación única de unidades nuevas:

$$y_{jk} \geq y_{j,k+1} \quad \forall j, k = 1 \dots Z_j - 1$$

viii) Costo de ajuste por escoger dos etapas en un equipo

$$CA_{kt} = c_{2t} \quad \forall e_{2kt} = e_{3kt}$$

ix) Disyunción de cada unidad  $k$  añadida a la etapa  $j$ :

$$\left[ \begin{array}{c} y_{jk} \\ V_{jkt} \leq V_j^U \\ V_{jkt} = V_{j,k,t-1} + e_{jkt} \\ \\ w_{jkt} \\ V_{jkt} \geq V_j^L \\ \\ \left[ \begin{array}{c} w_{ijk1t}^B \\ V_{ijk1t}^B \leq V_{jkt} \\ V_{ijk1t}^B \leq V_j^U \end{array} \right] \vee \dots \vee \left[ \begin{array}{c} w_{ijkmt}^B \\ V_{ijkmt}^B \leq V_{jkt} \\ V_{ijkmt}^B \leq V_j^U \end{array} \right] \vee \left[ \begin{array}{c} w_{ijkt}^C \\ V_{ijkt}^C \leq V_{jkt} \\ V_{ijkt}^C \leq V_j^U \\ V_{ijkt}^C \geq B_{it} S_{ij} \end{array} \right] \forall i \\ \\ \left[ \begin{array}{c} z_{jkt} \\ CE_{jkt} = K_{jt} + c_{jt} e_{jkt} \\ V_j^L \leq e_{jkt} \leq V_j^U \end{array} \right] \vee \left[ \begin{array}{c} \neg z_{jkt} \\ CE_{jkt} = 0 \\ e_{jkt} = 0 \end{array} \right] \\ \\ \vee \left[ \begin{array}{c} \neg w_{jkt} \\ V_{jkt} \geq 0 \end{array} \right] \forall t \\ \\ \vee \left[ \begin{array}{c} \neg y_{jk} \\ V_{jkt} = 0 \end{array} \right] \forall j, k \end{array} \right]$$

x) Ecuaciones Lógicas:

$$\sum_m w_{ijkmt}^B + w_{ijkt}^C = w_{jkt} \quad \forall i, j, k, t$$

$$\sum_t z_{jkt} = y_{jk} \quad \forall j, k$$

$$y_{jk} \geq \sum_t w_{jkt} \quad \forall j, k$$

$$w_{jkt} \leq y_{jk} \quad \forall j, k, t$$

$$w_{jkt} \leq \sum_{\tau=1}^t z_{jk\tau} \quad \forall j, k, t$$

$$z_{jkt} \leq w_{jkt} \quad \forall j, k, t$$

xi) Variables:

$$n_{it}, B_{it}, T_{Lit}, V_{jk}, e_{jkt}, V_{jkt}, V_{ijkmt}^B, V_{ijkmt}^C \geq 0 \quad y_{jk}, w_{jkt}, w_{ijkmt}^B, w_{ijkmt}^C, z_{jkt} = \{\text{Verdadero}, \text{Falso}\}$$

$$i = 1 \dots N \quad j = 1 \dots N \quad t = 1 \dots T \quad k = 1 \dots Z_j \quad m = 1 \dots N_j^{old}$$

A este modelo lo llamamos PDR (programa disyunto retrofit).

**Apéndice C: Datos para ejemplos**

Etapa 2 Fermentadores

\* Etapa 2: fermentacion 5, 9 7 dias

CT (1,2)	120
CT (2,2)	216
CT (3,2)	168

S(i,2) size factor for product i in stage j in hl per hl 11.0585885 = KGS. DE EXTRACTO PRODUCIDOS semana/VOL. DE CERVEZA PRODUCIDA POR SEMANA

Etapa 3 Reposo

\* Etapa 3: 6 dias de reposo

CT (1,3)	144
CT (2,3)	144
CT (3,3)	144

S(i,3) size factor for product i in stage j in hl per hl 11.0585885 = KGS. DE EXTRACTO PRODUCIDOS semana/VOL. DE CERVEZA PRODUCIDA POR SEMANA

Etapa 4 Filtración

SE CONSIDERAN 125 MIN DE TIEMPO DE FILTRACIÓN PARA COCTOS REGULARES SU CAPACIDAD ES: 958515.041 HL. MENSUALES

CT (i,4) tiempo de ciclo producto i en etapa j 2.08333333  
 S(i,4) size factor for product i in stage j in hl per hl 5.76702051 = hl filtrados mes/VOL. DE CERVEZA PRODUCIDA POR mes

La demanda de los productos se utilizó conforme a la siguiente tabla y los períodos requeridos para el ejemplo en cuestión.

Q(T,l) demanda de producto i en período t por 1000 kg

	1	2	3
1	200	600	1000
2	220.0	720.0	1150.0

3	242.0	864.0	1322.5
4	266.2	1036.8	1520.9
5	292.8	1244.2	1749.0
6	322.1	1493.0	2011.4
7	354.3	1791.6	2313.1
8	389.7	2149.9	2660.0
9	428.7	2579.9	3059.0
10	471.6	3095.9	3517.9
11	518.7	3715.0	4045.6
12	570.6	4458.1	4652.4
13	627.7	5349.7	5350.3
14	690.5	6419.6	6152.8
15	759.5	7703.5	7075.7
16	835.4	9244.2	8137.1
17	919.0	11093.1	9357.6
18	1010.9	13311.7	10761.3
19	1112.0	15974.0	12375.5
20	1223.2	19168.8	14231.8

## DEMANDAS CAPITULO 4

	2%	13%	6%
	1 especialidad	2 light	3 restantes
1	200.00	600.00	1000.00
2	204.00	678.00	1060.00
3	208.08	766.14	1123.60
4	212.24	858.08	1191.02
5	216.49	961.05	1262.48
6	220.82	1076.37	1338.23
7	225.23	1205.54	1418.52
8	229.74	1350.20	1503.63
9	234.33	1512.22	1593.85
10	239.02	1693.69	1689.48
11	243.80	1880.00	1790.85
12	248.67	2086.80	1898.30
13	253.65	2316.34	2012.20
14	258.72	2547.98	2132.93
15	263.90	2802.78	2260.90
16	269.17	3083.06	2396.56
17	274.56	3391.36	2540.35
18	280.05	3730.50	2692.77
19	285.65	4066.24	2854.34
20	291.36	4432.20	3025.60
21	297.19	4831.10	3207.14
22	303.13	5265.90	3399.56
23	309.20	5739.83	3603.54
24	315.38	6256.42	3819.75
25	321.69	6819.49	4048.93

**Apéndice D: Listado**

GAMS Rev 230 WIN-VIS 23.0.2 x86/MS Windows 02/24/09 16:08:08 Page 1

General Algebraic Modeling System  
Compilation

```

1
2 *
3 * Ejemplo 10 rolling: 3 productos 4 etapas 10 periodos
4 *
5 *$offlisting
6
7 *
8
9     SETS
10    I  products /1*3/
11 *Bh 1 TKL 2 CB 3*
12    J  stages /1*4/
13    K  new units /1*10/
14    M  old units /1*2/
15    T  time periods /1*20/
16
17 ;
18    ALIAS (T,TP);
19
20    PARAMETERS
21 * parameters for both problems
22    Z(J)      maximum number of units to be added to stage j
23    /1 10,2 10,3 10,4 10/
24    Nold(J)   number of initial existing units in stage j
25    /1 1,2 1,3 2,4 1/
26    VLO(J)   minimum volume of new units in stage j in 1000 l
27    /1 1,2 2,5,3 2,5,4 1/
28    VU(J)   maximum volume of new units in stage j in 1000 l
29    /1 10,2 10,3 10,4 10/
30    H(T)     operating time period in h for period t
31    u /10000/ ;
32
33    h(t)=8000;
34
35    TABLE FC(T,J)  annualized fixed charge in period t of new unit i stage j in 1000 dollar
36           1      2      3      4
37    1  13.29 35.21 42.85 7.19
38    2  13.29 35.21 42.85 7.19
39    3  13.29 35.21 42.85 7.19
40    4  13.29 35.21 42.85 7.19
41    5  13.29 35.21 42.85 7.19
42    6  13.29 35.21 42.85 7.19
43    7  13.29 35.21 42.85 7.19
44    8  13.29 35.21 42.85 7.19
45    9  13.29 35.21 42.85 7.19
46    10 13.29 35.21 42.85 7.19
47    11 13.29 35.21 42.85 7.19
48    12 13.29 35.21 42.85 7.19
49    13 13.29 35.21 42.85 7.19
50    14 13.29 35.21 42.85 7.19
51    15 13.29 35.21 42.85 7.19
52    16 13.29 35.21 42.85 7.19
53    17 13.29 35.21 42.85 7.19
54    18 13.29 35.21 42.85 7.19
55    19 13.29 35.21 42.85 7.19
56    20 13.29 35.21 42.85 7.19
57 ;
58
59    TABLE VCT(T,J)  annualized cost coefficient of a new unit in stage j in period t
60           1      2      3      4
61    1  0.01329 0.03670 0.04483 0.00786

```



62	2	0.01329	0.03670	0.04483	0.00786
63	3	0.01329	0.03670	0.04483	0.00786
64	4	0.01329	0.03670	0.04483	0.00786
65	5	0.01329	0.03670	0.04483	0.00786
66	6	0.01329	0.03670	0.04483	0.00786
67	7	0.01329	0.03670	0.04483	0.00786
68	8	0.01329	0.03670	0.04483	0.00786
69	9	0.01329	0.03670	0.04483	0.00786
70	10	0.01329	0.03670	0.04483	0.00786
71	11	0.01329	0.03670	0.04483	0.00786
72	12	0.01329	0.03670	0.04483	0.00786
73	13	0.01329	0.03670	0.04483	0.00786
74	14	0.01329	0.03670	0.04483	0.00786
75	15	0.01329	0.03670	0.04483	0.00786
76	16	0.01329	0.03670	0.04483	0.00786
77	17	0.01329	0.03670	0.04483	0.00786
78	18	0.01329	0.03670	0.04483	0.00786
79	19	0.01329	0.03670	0.04483	0.00786
80	20	0.01329	0.03670	0.04483	0.00786
81					
82					

83 TABLE QU(T,I) upper bound on production of product i in period t in 1000 kg per y

84		1	2	3	
85	1	200	600	1000	
86	2	220.0	720.0	1150.0	
87	3	242.0	864.0	1322.5	
88	4	266.2	1036.8	1520.9	
89	5	292.8	1244.2	1749.0	
90	6	322.1	1493.0	2011.4	
91	7	354.3	1791.6	2313.1	
92	8	389.7	2149.9	2660.0	
93	9	428.7	2579.9	3059.0	
94	10	471.6	3095.9	3517.9	
95	11	518.7	3715.0	4045.6	
96	12	570.6	4458.1	4652.4	
97	13	627.7	5349.7	5350.3	
98	14	690.5	6419.6	6152.8	
99	15	759.5	7703.5	7075.7	
100	16	835.4	9244.2	8137.1	
101	17	919.0	11093.1	9357.6	
102	18	1010.9	13311.7	10761.3	
103	19	1112.0	15974.0	12375.5	
104	20	1223.2	19168.8	14231.8	

105 ;  
 106 \* 166 M hl por mes /3 productos = 55.333  
 107 \* 55M hl por mes por producto \*12 meses = 664 M hl por año por producto

108  
109

110 TABLE Vold(M,J) volume of initial existing unit m in stage j in 1 000 l

111		1	2	3	4
112	1	1	4	3	3
113	2			3	

114 ;

115 TABLE CT(I,J) unit cycle time of product i in stage j in h

116		1	2	3	4
117	1	3.73	288	336	2.08
118	2	3.73	216	216	2.08
119	3	3.73	168	120	2.08

120 ;  
 121  
 122 \* : cocimientos 224 min  
 123 \* Etapa 2: fermentacion 12, 9 7 dias  
 124 \* Etapa 3: 14, 9, 5 dias de reposo  
 125 \* Etapa 4: 125 min de filtración

126  
127

128 TABLE S(I,J) size factor for product i in stage j in l per kg

129		1	2	3	4
130	1	.3	11	11	5.76
131	2	.3	11	11	5.76

```

132          3 .3 11 11 5.76
133          ;
134
135 SCALAR
136     ZU      maximum number of units to be added to the plant
137     /40/
138     ;
139
140 vARIABLES
141 *variables for all problems
142 * continuous variables
143     VOL(J,K)      volume of new unit k in stage j in1000 l
144     V(T,J,K)      volume of new unit k in stage j in p erio d t in 1000 l
145     E(T,J,K)      volume of new unit k in stage j in perio d t in 1000 l for objective
146     VB(T,I,J,K,M) processing volume option B with unit m in 1000 l
147     VC(T,I,J,K)   processing volume option C in 1000 l
148     LNNB(T,I)     transformed number of batches ofproduct i
149     LNB(T,I)      transformed batch size of product i
150     LNTL(T,I)     transformed limiting cycle time of product i
151     B(T,I)        batch size of product i in 1000 k
152 *variables for both master problems
153     B1(T,I,J,K)   disaggregated batch size of product iin 1000 kg
154     B2(T,I,J,K)   disaggregated batch size of product in 1000 kg
155     Y(J,K)        decision variable for selection of unit j
156     W(T,J,K)      operate unit k in stage j for product i in period t
157     WB(T,I,J,K,M) operate in phase with existing unit m
158     WC(T,I,J,K)   operate in sequence
159     ZE(T,J,K)     expand unit k in stage j in period t
160     CTMD          cost of expansion
161 *diff(t,i)
162     APPX(T,I)
163 *nb(T,I)  NUMBER OF BATCHES
164
165     ;
166
167     POSITIVE VARIABLE V, VB, VC, LNNB,X,B,b1,b2,VOL, E,gap1,gap2
168 *nb(T,I)  NUMBER OF BATCHES
169
170     ;
171 *diff(t,i) ;
172 *integer variable nb;
173 BINARY VARIABLE y, W,WB,WC,ZE;

373
374 *$include fixedbinaries2.inc
375 *these variables are the result of teh master problem
376 *****bounds on variables*****
377 * Tight lower bounds are computed below for cycle times and batch sizes
378
379     PARAMETERS  nb(T,I)  NUMBER OF BATCHES
380               LNTLL(I)  LOWER BOUND ON LNTL
381               LNTLU(I)  UPPER BOUND ON LNTL      ;
382 *             LNTLL(I) = LOG(SMAX(J, CT(I,J)/(NOLD(J)+Z(J)))) ;
383 *             LNTLU(I) = LOG(SMAX(J, CT(I,J)/NOLD(J)))      ;
384
385 *LNTL.LO(T,I) = LNTLL(I) ;
386 *LNTL.UP(T,I) = LNTLU(I) ;
387
388     PARAMETERS  BL(T,I)  LOWER BOUND ON B
389               BU(I)      UPPER BOUND ON B      ;
390               BI(T,I) = QU(T,I) * ( SMAX(J, CT(I,J)/(NOLD(J)+Z(J)))/ H ( T);
391               Bu(I) = SMIN(J, (Z(J)*VU(J)+SMAX(M,VOLD(M,J)))/S(I,J)
392               ;
393
394 *B.lo(T,I) = BL(T,I) ;
395 *LNB.lo(T,I) = LOG(BL(T,I));
396 *B.up(T,I) = BU(I) ;
397
398 *LNB.UP(T,I) = LOG(BU(I));
399

```

```

400 *LNNB.LO(T,I) = LOG(10);
401 *LNNB.UP(T,I) = LOG(QU(T,I)*1000/BL(T,I));
402 VOL.UP(J,K) = VU(J);
403 V.UP(T,J,K) = VU(J);
404 E.UP(T,J,K) = VU(J);
405 VC.UP(T,I,J,K) = VU(J);
406 VB.UP(T,I,J,K,M) = VU(J);
407 CTMD.lo=0;
408
409 *****
410
411 EQUATIONS
412 MPROD(T,I)    production of product i less than upper bound
413 VOLB(T,I,J,M) volume of new unit in phase w old unit
414 MHORIZONR(T) horizon constraint
415 BATCHR(T,I)  constraint to use B without transformati on in other equations
416 MVOLC(T,I,J,K)    volume of new unit in sequence
417 *DISAGGB(T,I,J,K)    disaggregate B
418 VOLRESB(T,I,J,K,M)  volume restriction opt B
419 VOLRESC(T,I,J,K)    volume restriction opt A
420 VOLUME(T,J,K)      calculate volume for objective
421 *VOLTOT(T,J,K)     overall volume
422 ELOWD(T,J,K)      e greater than lower bound
423
424 ELOWD(T,J,K)      e greater than lower bound
425 COSTMD           define objective function
426 VOLLD(t,J,K)     volume of new unit should be greater than lower bound
427 VOLUD(t,J,K)     volume of unit zero if not installed
428 VOLBZEROD(T,I,J,K,M)  set volume to zero if not installed
429 VOLCZEROD(T,I,J,K)  set volume to zero if not installed
430 EZEROD(T,J,K)    set e to zero if not installed in period t
431 NEW_UPPER       number of new units added less than upper bound
432
433
434 SOSD(J,K)       SOS for instalment of unit
435
436 CYCLE_TD(T,I,J)    determine the limiting cycle time
437
438 *SPECIAL
439 LOGIC1(T,I,J,K)    new unit is installed if operated in some mode
440 LOGIC2(J,K)        to assign new units in order
441 LOGIC3(T,I,J,K,M)  binary zero
442 LOGIC4(T,J,K)      do not use if never instal
443 LOGIC5D(J,K)       must use if installed
444 LOGIC5(T,J,K)      must use if installed
445 LOGIC6(T,J,K)      can only instal if used
446 LOGIC7(T,J,K)     sequence of installations
447
448
449 *BATCH1ND(T,I,J,K)    for B1
450 *BATCH2ND(T,I,J,K)    for B2
451 DISAGGBN(T,I,J,K)    disaggregate B
452 ;
453
454
455 COSTMD..
456 CTMD =E= SUM((T,J,K)$ (ORD(K) LE Z(J)),(FC(T,J)*ZE(T,J,K)
457 +VCT(T,J)*E(T,J,K)*1000)) +(gap1+gap2)*100000000 ;
458
459 MPROD(T,I)..
460 LNNB(T,I)+LNB(T,I) =g= LOG(QU(T,I)/100)-gap1;
461
462
463 VOLB(T,I,J,M)$ (ORD(M) LE Nold(J))..
464 * S(I,J)*B2(T,I) =I= Vold(M,J)+SUM(K$(ORD(K) LE Z(J)),VB(T,I,J,K,M));
465 S(I,J)*(EXP(LNB(T,I)))=L= Vold(M,J)+SUM(K$(ORD(K) LE Z(J)),VB(T,I,J,K,M))
466 ;
467
470 *BATCH1ND(T,I,J,K)$Y(J,K)..
471 * B1(T,I,J,K) =L= BU(I)*(1-WC(T,I,J,K));

```

```

472 *BATCH2ND(T,I,J,K)$Y(J,K)..
473 *   B2(T,I,J,K) =L= BU(I)*WC(T,I,J,K);
474 DISAGGBN(T,I,J,K)..
475   B(T,I) =E= B1(T,I,J,K)+B2(T,I,J,K);
476
477 MHORIZONR(T)..
478   SUM(I,(EXP(LNNB(T,I)+LNTL(T,I))))-H(T) -gap2=l= 0;
479 BATCHR(T,I)..
480   EXP(LNB(T,I)) - B(T,I) =l= 0;
481 *diff(t,i) ;
482 MVOLC(T,I,J,K)$ (ORD(K) LE Z(J))..
483   S(I,J)*B2(T,I,J,K) =l= VC(T,I,J,K)+u*(1- wC(t,i,j,k));
484
485 VOLRESB(T,I,J,K,M)$ ((ORD(K) LE Z(J)) AND (ORD(M) LE NOLD(J)))..
486   VB(T,I,J,K,M) =L= V(T,J,K);
487 VOLRESC(T,I,J,K)$ (ORD(K) LE Z(J))..
488   VC(T,I,J,K) =L= V(T,J,K);
489 VOLUME(T,J,K)..
490   V(T,J,K) =E= V(T-1,J,K)+E(T,J,K);
491 *VOLUME(T,J,K)..
492 *   V(T,J,K) =E= V(T-1,J,K)*(Y(J,K))+E(T,J,K);
493 *VOLTOT(T,J,K)..
494 *   VOL(J,K) =g= V(T,J,K);
495 *   ELOWD(T,J,K)..
496 *   E(T,J,K) =G= VLO(J)*ZE(T,J,K);
497
498 *VOLLD(t,J,K)$ (ORD(K) LE Z(J))..
499 *   V(t,J,K) =G= VLO(J)*w(t,J,K);
500
501 VOLLD(t,J,K)$ (ORD(K) LE Z(J))..
502   V(t,J,K) =G= VLO(J)*w(t,J,K);
503 VOLUD(t,J,K)$ (ORD(K) LE Z(J))..
504   e(t,J,K) =L= VU(J)*w(t,J,K);
505 VOLBZEROD(T,I,J,K,M)$ ((ORD(K) LE Z(J)) AND (ORD(M) LE Nold(J)))..
506   VB(T,I,J,K,M) =L= VU(J)*WB(T,I,J,K,M);
507 VOLCZEROD(T,I,J,K)$ (ORD(K) LE Z(J))..
508   VC(T,I,J,K) =L= VU(J)*WC(T,I,J,K);
509 EZEROD(T,J,K)..
510   E(T,J,K) =L= VU(J)*ZE(T,J,K);
511
512 NEW_UPPER..
513   SUM((J,K)$ (ORD(K) LE Z(J)), Y(J,K)) =L= ZU;
514
515 LOGIC1(T,I,J,K)$ (ORD(K) LE Z(J))..
516   SUM(M$(ORD(M) LE Nold(J)), WB(T,I,J,K,M))+WC(T,I,J,K) =E= W(T ,J,K)
517 ;
518 LOGIC2(J,K)$ (ORD(K) LE (Z(J)-1))..
519   -Y(J,K)+Y(J,K+1) =L= 0;
520 LOGIC3(T,I,J,K,M)$ (ORD(M) GT Nold(J))..
521   WB(T,I,J,K,M) =E= 0;
522 LOGIC4(T,J,K)..
523   W(T,J,K) =L= Y(J,K);
524 LOGIC5D(J,K)..
525   SUM(T,W(T,J,K)) =g= Y(J,K);
526 LOGIC5(T,J,K)..
527   SUM(TP$(ORD(TP) LE ORD(T)), ZE(TP,J,K)) =G= W(T,J,K);
528 LOGIC6(T,J,K)..
529   ZE(T,J,K) =L= W(T,J,K);
530 LOGIC7(T,J,K).. ZE(T,J,K) =G= ZE(T,J,K+1);
531
532 SOSD(J,K)..
533   SUM(T,ZE(T,J,K)) =E= Y(J,K);
534
535 CYCLE_TD(T,I,J)..
536 * CT(I,J)*EXP(-LNTL(T,I))-SUM(K$(ORD(K) LE Z(J)), WC(T,I,J,K)) =L= NOLD(J);
537 * CT(I,J)*EXP(-LNTL(T,I))-SUM(K$(ORD(K) LE Z(J)), WC(T,I,J,K)) =L= NOLD(J);
538 *SPECIAL.. SUM((J,K), Y(J,K)) =G= 1;
539 *numBatch(t,i).. NB(T,I)=e= exp(LNNB(T,I));

```

```

546 FILE FD /dicopt.opt/;
547 PUT FD;
548 PUT "*/dicopt.opt"/;
549 PUT "stop 1"/;
550 PUT "infeasder 1"/;
551 PUT "maxcycles=100"/;
552 PUTCLOSE;
553
554 model conDP /all/;
555
557 OPTION LIMROW = 0;
558 OPTION LIMCOL = 0;
559 oPTION ITERLIM = 1000000000;
560 OPTION RESLIM = 1000000000;
561 OPTION OPTCR = 0;
562 OPTION OPTCA = 0;
563 condp.WORKSPACE=40;
564
565 OPTION MIP = cplex;
566 OPTION rmiNLP = minos;
567 *OPTION RMINLP = coinipopt;
568 *OPTION SYSOUT = on;
569 *OPTION SOLPRINT = Off;
570
571 *OPTION
572 * SOLPRINT=OFF,
573 * LIMROW=0,
574 * LIMCOL=0,
575 * BRATIO=0,
576 * SYSOUT=OFF,
577 * OPTCR=101.10,
578 *OPTCA = 10,
579 * ITERLIM=100000000,
580 * RESLIM= 10000000
581 *;
582
583 condp.PRIOROPT = 1;
584 OPTION BRATIO = 1;
585
586 condp.OPTFILE = 1;
587 *WC.fx(T,I,J,K)=0;
588
589 *// LNNB.fx(T,I)=1;
590
591 *b.fx(t,i)=1;
592
595 solve conDP using MINLP minimizing ctmd;
596 NB(T,I)= exp(LNNB.L(T,I))
597 diSPLAY NB,B.L,gap1.l, gap2.l;

```

\*\*\*\* LIST OF STRAY NAMES - CHECK DECLARATIONS FOR SPURIOUS COMMAS  
 \*\*\*\* STRAY NAME X OF TYPE VAR

COMPILATION TIME = 0.000 SECONDS 3 Mb WIN230-230 Feb 12, 2009  
 GAMS Rev 230 WIN-VIS 23.0.2 x86/MS Windows 02/24/09 16:08:08 Page 2  
 General Algebraic Modeling System  
 Model Statistics SOLVE conDP Using MINLP From line 595

MODEL STATISTICS

BLOCKS OF EQUATIONS	27	SINGLE EQUATIONS	27,798
BLOCKS OF VARIABLES	18	SINGLE VARIABLES	20,883
NON ZERO ELEMENTS	78,055	NON LINEAR N-Z	720
DERIVATIVE POOL	12	CONSTANT POOL	26
CODE LENGTH	6,441	DISCRETE VARIABLES	2,240
		RELAXED VARIABLES	4,400 ****

GENERATION TIME = 0.609 SECONDS 10 Mb WIN230-230 Feb 12, 2009  
 EXECUTION TIME = 0.641 SECONDS 10 Mb WIN230-230 Feb 12, 2009

GAMS Rev 230 WIN-VIS 23.0.2 x86/MS Windows 02/24/09 16:08:08 Page 3  
 General Algebraic Modeling System  
 Solution Report SOLVE conDP Using MINLP From line 595

S O L V E S U M M A R Y

MODEL conDP            OBJECTIVE CTMD  
 TYPE MINLP            DIRECTION MINIMIZE  
 SOLVER DICOPT         FROM LINE 595

\*\*\*\* SOLVER STATUS    1 NORMAL COMPLETION  
 \*\*\*\* MODEL STATUS    8 INTEGER SOLUTION  
 \*\*\*\* OBJECTIVE VALUE         1801.7464

RESOURCE USAGE, LIMIT    2692.334 1000000000.000  
 ITERATION COUNT, LIMIT   37185 1000000000  
 EVALUATION ERRORS        0        0

-----  
 Dicopt2x-C    Feb 14, 2009 23.0.2 WIN 5247.9411 VIS x86/MS Windows  
 -----

Aldo Vecchietti and Ignacio E. Grossmann  
 Engineering Design Research Center  
 Carnegie Mellon University  
 Pittsburgh, Pennsylvania 15213  
 -----

Reading parameter(s) from "C:\tesis UNIJIE\dicopt.opt"  
 >> \*dicopt.opt  
 >> stop 1  
 >> infeasder 1  
 >> maxcycles=100  
 Finished reading from "C:\tesis UNIJIE\dicopt.opt"

--- DICOPT: Stopped on crossover

The search was stopped because a crossover occurred.  
 The bound provided by the MIP master problem exceeds  
 the objective of the last NLP problem.

## RESUMEN BIOGRÁFICO

María Gabriela García Ayala

Candidato para el Grado de

Doctor en Ingeniería con Especialidad en Ingeniería de Sistemas

Tesis: Expansión de Capacidad de Plantas Tipo Batch

Campo de Estudio: Investigación de Operaciones.

Datos Personales: Nacida en Monterrey, Nuevo León el 23 de Mayo de 1975, hija de José Humberto García Mata y Bertha Isabel Ayala Paul. Casada con Juan Carlos Cabada Amaya.

Educación: Egresada del Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey, obteniendo el grado de Ingeniera Química y Administradora en 1996.

Maestra en Ciencias en Ingeniería Química por la Universidad de Carnegie Mellon, grado obtenido en 2004.