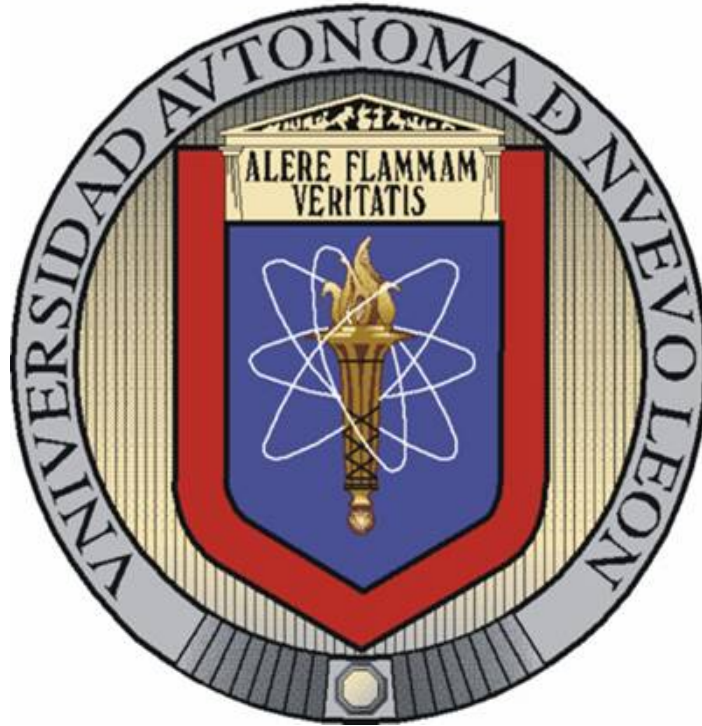


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
DIVISIÓN DE ESTUDIOS DE POSGRADO



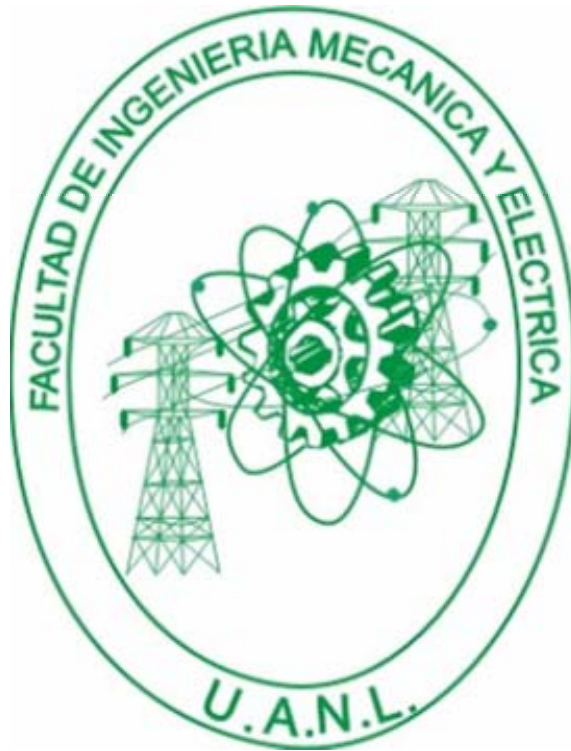
**OPTIMIZACIÓN DE INVENTARIOS EN
SISTEMAS DE PRODUCCIÓN**

por

SAMUEL DAVID PACHECO LEAL

**Como requisito parcial para obtener el grado de
MAESTRO EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

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



**OPTIMIZACIÓN DE INVENTARIOS EN
SISTEMAS DE PRODUCCIÓN**

por

SAMUEL DAVID PACHECO LEAL

**Como requisito parcial para obtener el grado de
MAESTRO EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

San Nicolás de los Garza, Nuevo León.

Diciembre de 2006

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la tesis **“OPTIMIZACIÓN DE INVENTARIOS EN SISTEMAS DE PRODUCCIÓN”**, realizada por el alumno Samuel David Pacheco Leal, matrícula 970963, sea aceptada para su defensa como opción al grado de Maestro en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dr. Igor S. Litvinchev

Asesor

Dr. Fernando López Irarragorri

Revisor

M. C. Gerardo Naranjo Sotomayor

Revisor

Vo. Bo.

Dr. Guadalupe Alan Castillo Rodríguez

División de Estudios de Posgrado

Índice general

Dedicatoria	x
Agradecimientos	xi
Resumen	xii
1. Introducción	1
1.1. Descripción del Problema	1
1.2. Motivación	8
1.3. Sistemas de Producción	8
1.4. Literatura Previa	9
1.5. Objetivo de la Tesis	11
1.6. Descripción de la Tesis	12
2. Modelación	14
2.1. Suposiciones Generales	15
2.2. Modelos para Sistemas Uniproducto	16
2.2.1. Nomenclatura	16

	IV
2.2.2. Modelo para sistemas de producción tipo simple	21
2.2.3. Modelo para sistemas de producción tipo ensamble	24
2.3. Modelos para Sistemas Multiproducto	27
2.3.1. Nomenclatura	29
2.3.2. Sistemas de tipo de producción simple	31
2.3.3. Sistemas de tipo de producción de ensamble	35
2.4. Producir contra Inventario.	39
2.5. Producir contra Demanda.	39
2.6. Método de Solución	39
2.6.1. Herramientas Computacionales	41
3. Datos	43
3.1. Importancia de los Datos.	43
3.2. Horizonte de Tiempo a Optimizar	44
3.3. Coeficientes de Transferencia (α y β)	46
3.4. Coeficientes de Costos	46
3.4.1. Costo por generar salidas de almacén	47
3.4.2. Costo por mantener inventario	47
3.4.3. Costo por uso de nueva tecnología	48
3.4.4. Costo por incumplir demanda	49
4. Metodología	51
4.1. Identificar el cuello de botella del sistema	51
4.2. Reconocer la capacidad máxima instalada	53

4.3. Establecer el número de salidas de almacén	54
4.4. Diferentes Tecnologías	55
5. Experimentación	56
5.1. Experimentación	56
5.2. Resultados	58
6. Conclusiones y Recomendaciones	65
6.1. Conclusiones	65
6.2. Aportaciones	66
6.3. Recomendaciones	67
A. Corte y Ramificación (Branch and Bound)	68
B. AMPL	72
B.1. Lenguaje en AMPL	72
B.1.1. Reglas léxicas de AMPL	75
B.1.2. Los elementos de un conjunto	78
B.1.3. Expresiones que indexan y subíndices	79
B.1.4. Expresiones aritméticas, lógicas y de conjuntos. Funciones matemáticas	81
B.1.5. Declaraciones de elementos del modelo	87
B.1.6. Especificación de datos	95
B.1.7. Comandos del lenguaje	112
B.1.8. Optimizadores	116

Ficha autobiográfica

120

Índice de tablas

1.1. Simbología para grafos	5
2.1. Conjuntos para sistemas uniproducto	17
2.2. Parámetros para sistemas uniproducto	19
2.3. Variables para modelos uniproducto	21
2.4. Modelo SPSU	22
2.5. Conjuntos adicionales para SPEU	25
2.6. Parámetros adicionales para SPEU	25
2.7. Modelo SPEU	26
2.8. Declaración de variables del modelo SPEU	27
2.9. Correspondencia de restricciones entre SPSU y SPEU	28
2.10. Parámetros para sistemas multiproducto	30
2.11. Variables para modelos multiproducto	31
2.12. Modelo SPSM	32
2.13. Modelo SPEM	36
2.14. Declaración de variables del modelo SPEM	37
2.15. Correspondencia de restricciones entre SPSM y SPEM	38

3.1. Ejemplo: Parametrización de T	45
5.1. Capacidades de producción del sistema generado	58
5.2. Nomenclatura de los problemas generados	59
5.3. Parámetros de costos utilizados en la experimentación	59
5.4. Parámetros de costos por mantener inventario utilizados en la experi- mentación	60
5.5. Resultados de la experimentación	61
5.6. Resultados de la experimentación	62
5.7. Resultados de la experimentación	63
5.8. Resultados de la experimentación	64
B.1. Modelo básico del ejemplo	73
B.2. Modelo general del ejemplo	76
B.3. Datos para el ejemplo.	77
B.4. Operadores aritméticos (A), lógicos (L) y de conjuntos (S).	83
B.5. Funciones de generación de variables aleatorias en AMPL.	84
B.6. Algunas funciones aritméticas en AMPL.	85
B.7. Fichero de lotes para el modelo del ejemplo 1.1	114
B.8. Algunos comandos del entorno AMPL.	115

Índice de figuras

1.1. Flujo en serie.	5
1.2. Ejemplo 1 de sistema con configuración de red	6
1.3. Ejemplo 2 de sistema con configuración de red	6
2.1. Esquema de los diferentes modelos desarrollados.	15
5.1. Red generada para la experimentación	57
A.1. Un árbol de subproblemas: el conjunto de soluciones factibles, ramificado en F1 y F2; además F2 también ramificado en F3 y F4	69

Dedicatoria

A mis padres por su cariño, apoyo, esfuerzo y oraciones.

A mis hermanos, Víctor e Hilda por animarme en este trayecto.

A Alessa, mi prometida, por todo su apoyo y tiempo.

Agradecimientos

A Dios por guiarme para cumplir esta meta que forma parte de mi crecimiento personal y profesional.

A el Dr. Igor Litvinchev, mi asesor de tesis, por su apoyo, dedicación, guía, por compartir su tiempo y conocimiento para el desarrollo y conclusión de este proyecto.

Al personal docente de PISIS por compartir su experiencia y conocimiento enseñándome las bases para la culminación de este proyecto.

A mis compañeros de generación por hacer ameno este camino.

A Sintec, por dedicar tiempo para desarrollar el planteamiento del problema que abarca esta tesis.

A CONACYT por el apoyo que me brindó para cursar este programa de Maestría.

A mis padres, por su ejemplo de dedicación y constancia, además de el gran apoyo que me han dado en mi formación personal y profesional, gracias familia por estar ahí siempre.

A Alessa, mi prometida, por su gran apoyo, muchas gracias por todo el tiempo y esfuerzo dedicado a este trabajo.

Resumen

Samuel David Pacheco Leal

Candidato para el Grado de Maestro en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Título de Estudio:

Optimización de Inventarios en Sistemas de Producción

Número de páginas: 119

Objetivo y método de estudio:

Las empresas manufactureras que producen grandes cantidades de uno o varios productos incurren en diversos costos fuertes, entre los cuales se encuentran los costos de mantener inventario en proceso, posible adquisición de maquinaria nueva o asignar personal extra a determinadas estaciones de trabajo.

Debido a los grandes retos que enfrentan las compañías es imperativo optimizar la asignación de los recursos. La complejidad de los sistemas de producción hacen difícil esta asignación.

Los inventarios de producto en proceso constituyen un factor muy importante para la productividad de las empresas debido a que estos absorben las diferencias entre los tiempos de proceso evitando así que el flujo de producto se detenga. Los inventarios en proceso sirven como amortiguadores para las posibles interrupciones en el flujo de producto causados por paros en las diferentes estaciones de trabajo.

Mantener inventario conlleva un costo asociado considerable debido a la inversión necesaria para mantener producto así como por la asignación de espacio en piso dentro de las instalaciones, una disminución de los inventarios pueden incrementar las utilidades sustancialmente.

Otro aspecto importante dentro de los sistemas de producción está en el almacén de materias primas, el cual provee a varias áreas de trabajo dentro de la planta, generalmente el personal encargado de surtir materia prima, es asignado a otras actividades como la recepción y acomodo de material en sus respectivas ubicaciones, todo esto resulta en un deseo de generar el menor número posible de requisiciones a éste departamento, sin descuidar las necesidades del área de producción.

La capacidad total de un sistema productivo está dado por las estaciones de trabajo con menor tasa de producción, muchas veces se desea simular el que pasaría si se agregase capacidad a estas estaciones críticas, hemos modelado la opción de expandir la capacidad en estaciones de trabajo críticas.

Hemos desarrollado un enfoque de optimización sobre los inventarios de producto

en proceso para apoyar a la toma de decisiones de este tipo de problemas de la industria manufacturera. El objetivo de minimizar los costos incurridos en la producción de bienes es modelado a través de la modelación no-lineal, la cual se puede relajar programación entera mixta sin perder calidad en la solución.

Firma del asesor:

Dr. Igor S. Litvinchev

Capítulo 1

Introducción

1.1. Descripción del Problema

Los sistemas de producción son diseñados para producir grandes cantidades de uno o varios productos, producción es el proceso mediante el cual se manejan recursos para , transformar la materia prima en productos.

En los sistemas de producción es típico el producir grandes cantidades, siendo la solución de problemas de balanceo importantes problemas de medio término en la planeación de la producción [5].

Las líneas de producción necesitan mantener el inventario mínimo de producto en proceso debido a limitaciones de espacio o capital. De hecho, varias propuestas de metodologías justo-a-tiempo recomiendan una meta de cero inventario. Beneficios potenciales incluyen bajo costo, menos espacio, además de una rápida retroalimentación en cuestiones de calidad. Desafortunadamente, en presencia de diferentes tiempos de procesos, la salida de producto terminado puede ser drásticamente restringida si no

hay inventario de producto en proceso. Así, las líneas de ensamble generalmente requieren un inventario mínimo, pero no cero[4].

Durante las últimas décadas la manufactura en estaciones de trabajo ha emergido como una de las mejores técnicas utilizadas para la mejora competitiva. Adquirirla permite a las empresas que manejan lotes pequeños de producción tener ventajas económicas como las de aquellas que manejan lotes grandes manteniendo su flexibilidad de productos. El objetivo de la manufactura celular es reducir el tiempo de set-up, de flujo, de manejo de material, inventarios y el tiempo de respuesta a la demanda.

El incremento en la competencia entre empresas manufactureras ha provocado que se mejore el conocer la necesidad del cliente mediante la flexibilidad y responsabilidad de la empresa para satisfacerla. Éstos sistemas están conformados por estaciones de trabajo independientes, donde la configuración de estaciones de trabajo trae como beneficio:

- Línea de producción en lotes flexible
- Reducción de espacio en planta
- Disminuye el manejo de material
- Disminuye el tiempo de entrega
- Disminuye material en proceso
- Mejora la comunicación dentro de la empresa, con el cliente y con los proveedores

Una parte importante para el desarrollo de sistemas en estaciones de trabajo es la

distribución dentro de la estación, porque contribuye a la velocidad y la flexibilidad de producción.

Originalmente el esquema de trabajar en estaciones de trabajo no funcionaba porque se habían agrupado máquinas diferentes, además no se tenía un control de mantenimiento, planeación y supervisión de las prácticas como en el kanban, justo a tiempo, círculos de calidad los cuales miden la calidad, el trupt y el trabajo en proceso.

Una vez que la conexión entre las estaciones de trabajo y todo el proceso ha sido establecido es cuando se puede introducir la tecnología para apoyar el proceso de producción.

La configuración en estaciones de trabajo aumentan las utilidades ya que reducen costos e incrementan la productividad y la flexibilidad. Son un grupo de trabajo compuesto por procesos y herramientas utilizadas para cumplir con los objetivos de producción de la empresa.

El producir mediante estaciones de trabajo provee los siguientes beneficios:

- Ya que cada estación es independiente, se puede identificar fácilmente en qué lugar es necesario inventario y dónde se tiene inventario de más
- La velocidad del sistema de producción es esencial para satisfacer la demanda, por lo que se requiere una coordinación entre las estaciones que permita que el producto fluya de manera constante
- Ahorra dinero reduciendo costos, ya que se disminuye el inventario de producto en proceso y la coordinación entre las estaciones permite un flujo constante de

producto terminado

El tema desarrollado en esta tesis se enfoca a las conexiones entre las estaciones de trabajo, las cuales deben soportar las diferentes tasas de producción de cada estación manteniendo un flujo de producto en el sistema. Las conexiones entre las estaciones es mediante inventarios de producto en proceso, mejor conocidos por la palabra ‘*buffers*’, los cuales mantienen producto en proceso. Llamamos producto en proceso a aquel material que está en alguna parte del sistema sin ser procesado. Mantener producto en proceso es necesario para absorber la diferencia de los tiempos de producción entre las estaciones de trabajo, amortiguar interrupciones en el flujo que se pudieran presentar debido a mantenimientos programados y/o posibles fallas dentro del sistema. Tomando en cuenta esto, los sistemas que mantengan inventarios de producto en proceso son capaces de obtener una mayor producción respecto a sistemas similares que no los posean, así los inventarios juegan un rol muy importante dentro de los sistemas de producción. Existe un nivel mínimo de producto en los inventarios donde la capacidad máxima instalada del sistema es lograda, dicho nivel está dado por las capacidades de las estaciones de trabajo anterior y posterior a cada inventario donde puede estar restringido por el espacio disponible dentro de la planta. Si el inventario es mantenido en exceso, su costo puede resultar mayor al beneficio de incrementar la producción. La introducción de inventarios requiere una inversión extra de capital, además del espacio asignado dentro de la planta.

Enfocándonos en los inventarios que en nuestros grafos estarán representados por los nodos, las operaciones de producción son asociadas a los arcos. Esto se muestra

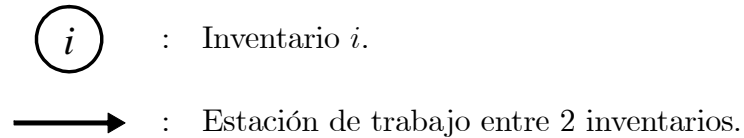


Tabla 1.1: Simbología para grafos

en las siguientes figuras en las que usaremos la simbología siguiente:

La organización de los sistemas de producción pueden ser en serie o en red, para los sistemas en red la producción puede ser realizando operaciones de tipo simple o de tipo ensamble en cada estación de trabajo. Supongamos que existen I inventarios en el sistema numerados consecutivamente de 1 a I , resultando $I - 2$ estaciones de trabajo, donde 1 corresponde al almacén de materias primas e I corresponde al almacén de producto terminado, los inventarios de producto en proceso van de 2 hasta $I - 1$. Así el flujo de material pasa por todos los inventarios en forma consecutiva, esto se muestra en la Figura 1.1.

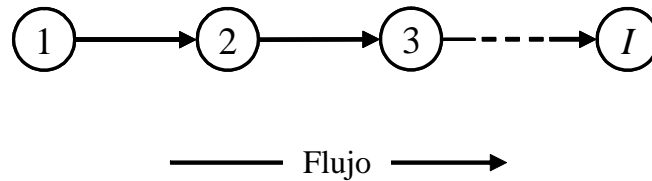


Figura 1.1: Flujo en serie.

Las configuraciones de red son más complejas y no hay relación entre el número de inventarios y el número de estaciones de trabajo, por ejemplo en la Figura 1.2 y Figura 1.3 se muestran sistemas con el mismo número de inventarios, pero la cantidad

de estaciones de trabajo es diferente, incluso es posible agregar estaciones entre los inventarios existentes.

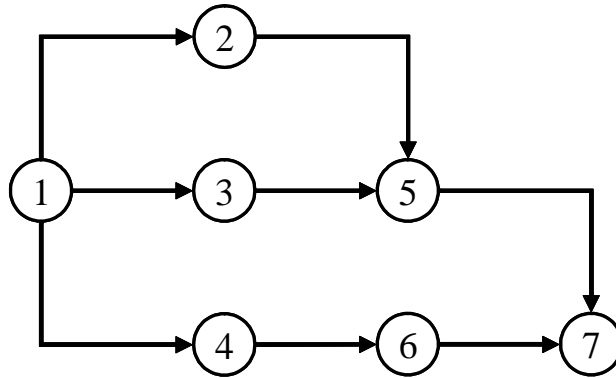


Figura 1.2: Ejemplo 1 de sistema con configuración de red

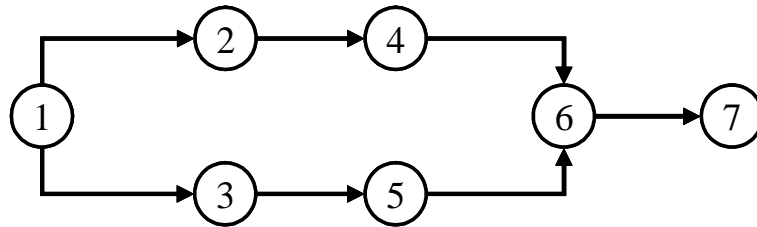


Figura 1.3: Ejemplo 2 de sistema con configuración de red

En cuanto a la configuración física del orden de las operaciones en los sistemas hemos realizado las dos diferencias anteriores, en cuanto a la forma en que el material fluye en el sistema haremos la diferencia entre producción tipo simple y producción tipo ensamble, las operaciones en los sistemas en serie, así como las operaciones en sistemas de red que cuenten con una sola operación anterior queda descartada la opción de tipo ensamble. Para aquellos nodos que cuenten con 2 o más operaciones

anteriores es posible considerar operaciones de tipo ensamble.

Operaciones de tipo ensamble son aquellas que requieren 2 o más de un tipo de producto para efectuar su producción, a esto le llamamos ensamble, por ejemplo, la operación que se encuentra entre los inventarios 6 y 7 de la Figura 1.3 es de tipo ensamble si para producir necesita producto proveniente de las operaciones localizadas entre los inventarios 4 y 6, 5 y 6, dicho de otra forma, si no hay producto disponible de al menos una de sus operaciones anteriores, no es posible producir. Caso contrario si fuese operación de tipo simple, en este caso se puede producir cuando exista producto disponible de cuando menos una de sus operaciones anteriores.

Teniendo en cuenta las características de los diferentes sistemas de producción mencionados anteriormente, bajo nuestros modelos también se contempla el abastecimiento de materias primas, dicho abastecimiento lo contemplamos desde el almacén de materias primas hacia las operaciones iniciales de producción. El personal del almacén de materias primas tiene que surtir material a diferentes áreas de trabajo dentro de la empresa, además de cumplir con sus tareas como la recepción y acomodo del material. Un número mínimo de requisiciones es deseado, buscando consumir el menor tiempo al personal. Debido a esto, el nivel óptimo para cada inventario es perturbado.

Es posible incrementar la capacidad en alguna estación de trabajo, este movimiento tiene un costo asociado. Asignar personal adicional, cambiar maquinaria son algunas de las opciones para lograr un incremento en la producción.

El objetivo de ésta tesis es encontrar los niveles óptimos de los inventarios de producto en proceso, generando un número mínimo de requisiciones de materia prima

así como incrementar la capacidad en alguna estación de trabajo en caso de ser necesario para sistemas productivos .

1.2. Motivación

Los modelos presentados en esta tesis fueron desarrollados en conjunto con SINTEC, importante empresa de consultoría líder en México en la cadena de valor. Un conjunto de modelos matemáticos son propuestos para dar respuesta a la problemática común de establecer los niveles de inventario de producto en proceso al mismo tiempo de minimizar el número de requisiciones de materia prima, así como la adecuada asignación de capacidad extra en las operaciones críticas para el sistema productivo.

1.3. Sistemas de Producción

Definimos un sistema de producción como un conjunto de actividades que son desarrolladas por capital humano y/o maquinaria dentro de una instalación, dichas actividades se realizan en estaciones de trabajo donde el material se mueve entre las estaciones a espacios asignados donde se mantiene el inventario de producto en proceso. Este inventario mantiene un flujo de material a lo largo del sistema soportando las diferentes tasa de producción. El producto en proceso es aquel material que está en alguna parte del sistema esperando a ser procesado. Los sistemas de producción pueden tener configuraciones muy diversas, desde un sistema lineal de un solo producto, hasta sistemas con configuraciones complejas y una gama de productos. Diferentes tipos de

producción como la producción de tipo de ensamble y de tipo simple se presentan comúnmente en la industria manufacturera.

1.4. Literatura Previa

Actualmente, los sistemas de producción pueden ser muy complejos. Es difícil desarrollar métodos para maximizar las utilidades por que existen muchos factores que afectan el costo y las compañías requieren más capacitación y preparación para competir. La utilización de modelos matemáticos puede ser una herramienta que brinda beneficios a las empresas.

Por definición, la programación de recursos concierne a la asignación de recursos limitados a tareas, por ejemplo, de equipo de trabajo a partes a materiales a procesar [2], el problema de asignación de recursos es ordenar las operaciones de forma que se optimice el criterio de desempeño. Este proceso es llamado estático, o programación off-line, si la información sobre los materiales y la disponibilidad de recursos se conoce de antemano. Por consiguiente la programación se debe realizar antes de que los recursos comiencen a operar. Ajustar la asignación para soportar situaciones inesperadas, tales como cambios en tiempos de proceso de una operación, mantenimiento o reparación de una máquina, se refiere a reprogramación.

Programación dinámica o programación on-line [8], [11], se refiere cuando el programador no tiene información de la llegada del material a procesar, en este caso, el programa es planeado basado en la situación actual de los recursos y será ejecutada hasta que requiera una modificación. La efectividad del programa puede ser mejorar

si el programador pronostica el tipo y el tiempo, para hacer algunas previsiones [2].

Desde el punto de vista de Control, la programación es llamada, deadlock-free si ésta satisface los requerimientos de control evitando el desabasto y sobreabasto en los inventarios de producto en proceso. Estos requerimientos son impuestos por la configuración en estaciones de trabajo. Existen dos métodos para establecer los requerimientos de control en una programación. El primer método es revisar el estado del desabasto o sobreabasto y evitarlos mientras se realiza la programación, comúnmente llamado método de deadlock avoidance. El segundo es crear un programa sobre una plataforma que considere el abasto y el desabasto al cual se le llama método de deadlock prevention.

La programación on-line puede considerarse como una repetición de la programación off-line en instantes cuando ocurren cambios. Enfoques y modelación de programación off-line son presentados a continuación.

Enfoque de Programación Matemática

Los sistemas de producción pueden ser formulados como modelos de Programación Enter (IP, por sus siglas en inglés) [2], [12]. La modelación consiste en una función objetivo y un conjunto de restricciones las cuales representan las condiciones de factibilidad y optimalidad, en función de ciertas variables de decisión, del sistema modelado. La programación óptima puede ser obtenida matemáticamente a partir de la solución de un modelo, [2] considerando varias suposiciones tales como espacio para inventarios ilimitados, rutas determinadas por tareas, sin tener en cuenta el manejo de materiales y no se consideran desperfectos en los recursos, el problema fue modelado con un modelo entero (IP). Sin embargo, la programación resultante de este modelo, debido a las

consideraciones no realistas antes mencionadas, no es posible aplicarlo en situaciones reales y pudiera resultar en desabasto o sobreabasto. Se hizo un esfuerzo para integrar las suposiciones antes mencionadas al modelo, lo cual, incrementó el número de variables y restricciones al modelo de IP. La complejidad computacional y la dificultad para integrar todas las suposiciones que se dan en el sistema son las dos grandes restricciones en la aplicación del este enfoque para la programación de sistemas de producción, donde el tiempo para generar nuevas programaciones es limitado.

Enfoque Heurístico

A diferencia de encontrar una solución óptima a problemas de programación complejo, existen varias técnicas de uso general que buscan una solución cerca del óptimo en relativamente poco tiempo. Una clase de dichas técnicas aplica a las reglas de asignación, donde las tareas son programadas en base a los atributos de los recursos o del material [6]. La clase de técnicas que están basadas en búsquedas locales, apuntan a encontrar una solución o una programación mejor que la actual mediante la búsqueda en el vecindario de la programación actual. Dentro de estas técnicas tenemos recocido simulado, búsqueda tabú y algoritmos genéticos [2].

1.5. Objetivo de la Tesis

Se han modelado sistemas productivos de tipo producción y de tipo ensamble, ambos con configuraciones de línea y de red, trabajando contra pedido o contra inventario de sistemas uniproducción o multiproducción. La característica que hace la diferencia entre el tipo de producción y de ensamble es la ruta que sigue el materi-

al hasta obtener producto terminado. En el de tipo producción la ruta en la que el material sigue a través de las estaciones de trabajo no necesita interceptar diferentes materiales para seguir su flujo. En los sistemas de tipo ensamble existen algunas intersecciones necesarias. En los sistemas con configuración de línea cada estación de trabajo cuenta con una entrada y una salida, en aquellos sistemas con configuración de red es posible que algunas estaciones de trabajo cuenten con más de una entrada y/o salida. Trabajar contra pedido tiene la consideración de que la cantidad específica de producto terminado es requerida por cada unidad de tiempo, en cambio cuando se trabaja contra inventario se requiere una cantidad considerable de material en un lapso dado de tiempo.

1.6. Descripción de la Tesis

La estructura de este trabajo es de la siguiente forma: En el Capítulo 2 presentamos la modelación de los sistemas de producción, presentando todos los detalles de los parámetros, variables y restricción que forman cada modelo. Este Capítulo lo dividimos en dos secciones, donde en la primera abarcamos los sistemas de producción uniproducción y en la segunda sección abarcamos la modelación de sistemas multiproducción.

En el Capítulo 3 tenemos una descripción más amplia de los parámetros a los que se debe atender con cuidado. Proponemos una serie de pasos para la optimización de sistemas productivos utilizando estos modelos, la cual se presenta en el Capítulos 4.

La experimentación realizada se expone en el Capítulo 5 y las conclusiones y

recomendaciones se encuentran en el Capítulo 6.

Capítulo 2

Modelación

Los modelos desarrollados en esta tesis pueden ser aplicados para sistemas que tengan la restricción de integralidad dentro de su proceso, es decir, un número entero de unidades es transformado en el sistema. Esta característica resulta en una mayor complejidad a la hora de resolver los problemas. Sistemas sin restricción de integralidad pueden ser tratados por los modelos. Nos abocaremos mayormente a sistemas con la característica de integralidad.

Diferentes modelos han sido desarrollados debido a las diferentes características que pueden presentar los sistemas de producción, entre las cuales se encuentran: tipo de producción simple o de tipo ensamble, se produce un solo producto o múltiples productos, se trabaja contra pedido o contra inventario. A continuación se muestra un esquema que muestra las características que diferencian a los modelos:

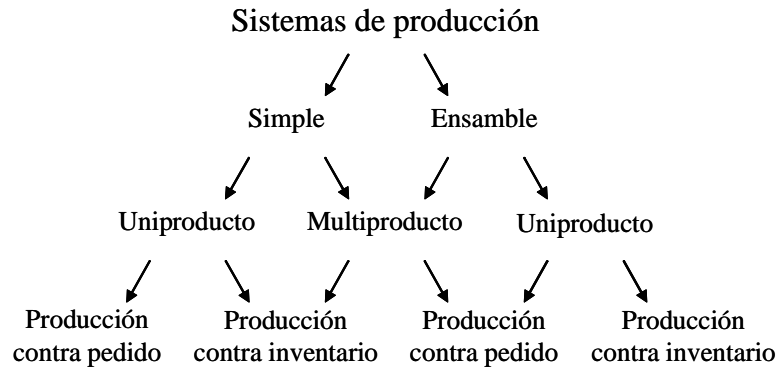


Figura 2.1: Esquema de los diferentes modelos desarrollados.

2.1. Suposiciones Generales

Se analizan los sistemas tomando en cuenta ciertas suposiciones en el desarrollo de los modelos, las cuales se muestran a continuación:

- No se toman en cuenta los tiempos de traslado de material entre los inventarios-estaciones de trabajo ni estaciones de trabajo-inventarios. Suponemos que la distancia es lo suficientemente pequeña o es un traslado que no hay forma de evitarlo, por lo cual no se puede minimizar, así las distancias dentro de las instalaciones no son tomadas en cuenta.
- El almacén de materias primas así como el de producto terminado son modelados como inventarios para los cuales no se incurre en ningún costo por mantener cualquier cantidad de inventario.
- El inventario de materias primas tiene producto y capacidad suficiente para satisfacer la demanda dada en cualquier tiempo.
- No se consideran desperfectos en la materia prima y en caso de que existan, se

asume que son inevitables.

- La tasa de producción es fija para cada estación de trabajo.

2.2. Modelos para Sistemas Uniproducto

Sistemas uniproducto son aquellos en los que se produce un solo tipo de bien, donde es posible utilizar una gama de diferentes materiales durante el proceso pero al final sólo un tipo de producto terminado. Estos sistemas pueden desarrollarse utilizando configuraciones en serie o en red. Cuando se presentan sistemas en red establecemos una restricción: cada inventario i puede tener tantas entradas como se requieran, pero sólo una salida, dicho de otra forma, cada operación se abastece de uno y sólo de un inventario. Esta restricción limita que cada estación de trabajo cuente con un espacio para mantener inventario de producto en proceso.

Sistemas multiproducto pueden ser modelados como uniproducto, siempre y cuando dentro del horizonte de optimización se produzca solo un tipo de producto terminado, como ejemplo podemos mencionar sistemas de la industria alimenticia, donde puede existir la restricción por consideraciones de sanidad o ajustes necesarios al equipo, no se pueden combinar materiales de diferentes productos en los contenedores donde se realiza la producción.

2.2.1. Nomenclatura

La nomenclatura empleada en los conjuntos, parámetros y variables utilizados en los modelos para sistemas uniproducto con tipo de producción simple y de ensamble,

produciendo contra demanda o contra pedido, se desarrollan a continuación.

Conjuntos

Los conjuntos utilizados para la modelación son mostrados en la Tabla 2.1.

- I** : Conjunto de todos los inventarios en el sistema, donde el inventario 1 representa el almacén de materias primas, I el almacén de producto terminado y para cada $i \neq \{1, I\}$ el i -ésimo inventario en el sistema
- A** : Matriz de incidencia entre los inventarios existentes del sistema de forma que $a_{ij} = \begin{cases} 1 & \text{si existe conexión entre } i \text{ y } j \\ 0 & \text{si no} \end{cases}$
- K** : Conjunto de las tecnologías disponibles en el sistema, para determinadas estaciones de trabajo, siendo k la k -ésima tecnología
- T** : Horizonte de tiempo a optimizar. Para $t \in \{1, 2, \dots, T\}$

Tabla 2.1: Conjuntos para sistemas uniproducto

Resultado así sistemas con $\mathbf{I} - 2$ inventarios de producto en proceso sobre los cuales hacemos la optimización. El número de estaciones de trabajo está dado por $[|\mathbf{A}| - (|a_{1j}| + |a_{iI}|)]$, para todos los $\{i, j\} (a_{1j}, a_{iI}) \in \mathbf{A}$.

Definimos I^1 como aquellos inventarios que reciben materia primas del inventario de materias primas. $I^1 \subset \mathbf{I}$, donde el envío de materia prima estará restringido a ciertas capacidades del sistema que más adelante se verán a más detalle.

Para algunas estaciones de trabajo definidas por A^1 se contempla un posible cambio de tecnología, esto para elevar su capacidad de producción, siendo \mathbf{K} el número

máximo de las diferentes opciones de cambio, ésta opción no es para todas las estaciones de trabajo. $A^1 \subset \mathbf{A}$.

Para el conjunto \mathbf{T} se asigna un número T , el cual se considera que abarca el tiempo total en que el sistema puede producir durante un día o cualquier período deseado. La amplitud de t será definida más adelante en el Capítulo 3.

Parámetros

Generalmente la optimización en sistemas de producción se enfoca en la programación directa de la maquinaria o equipo de producción, nuestro enfoque es optimizar los sistemas de producción mediante la optimización de los inventarios de producto en proceso. debido a este enfoque, los parámetros describen las necesidades de demanda, capacidades de flujo entre los inventarios, información respecto al sistema en estudio, entre otras cosas. Los parámetros para sistemas uniproducto y sus variantes se muestran a continuación:

Q	: Costo asociado por solicitar materia prima
e_{ij}^k	: Costo por utilizar la tecnología k entre los inventarios i y j
ρ_{ij}^k	: Variación a favor en capacidad de producción por el uso de la tecnología k entre los inventarios i y j
α_{ij}	: Número de unidades necesarias realizar una operación entre los inventarios i y j
β_{ij}	: Número de unidades que se producen por cada operación entre los inventarios i y j
ω	: Número de intervalos de tiempo para surtir materia prima
r_t	: Demanda al tiempo t
ν_i	: Número de unidades en el inventario i al inicio del período
C	: Costo asociado por unidad incumplida, respecto a la demanda
M	: Número suficientemente grande

Tabla 2.2: Parámetros para sistemas uniproducto

Parte de nuestra optimización es el generar un número *pequeño* de requisiciones de materias primas, Q representa la penalización por requisición, cabe mencionar que no es proporcional a la cantidad requerida. Para aquellas estaciones de trabajo donde esté contemplado un posible cambio de tecnología, se tiene un costo e_{ij}^k por utilizar la tecnología k entre los inventarios i y j , si elegimos hacer un cambio, se contempla para todo el tiempo, es por esto que es un parámetro independiente del tiempo. Cuando se realiza el cambio a la tecnología k entre los inventarios i y j se eleva la producción en ρ_{ij}^k unidades.

La producción en las estaciones de trabajo está dada por lo que llamaremos *coeficientes de transferencia*, donde α_{ij} representa el número de unidades necesarias del inventario i para realizar una operación entre los inventarios i y j , de modo similar β_{ij} son las unidades que resultan de introducir α_{ij} unidades entre los inventarios i y j . Éstos parámetros deben ser enteros y serán presentados con mayor detalle en el Capítulo 3.

Cuando materias primas son surtidas hacia diferentes estaciones de trabajo, asignamos un lapso de tiempo ω como el tiempo suficiente para preparar y surtir el material hacia cualquier estación de trabajo. Existe una demanda de producto terminado de cantidad r_t en el tiempo t , dicha cantidad debe ser no-negativa. El material en proceso que queda de un día para otro es considerado en los modelos por medio de ν_i , la cual es la cantidad de material en el inventario i . Incumplir demanda tiene una penalización C por cada unidad incumplida respecto a la demanda para cada tiempo t . Para cuestiones de modelación, definimos M como un número suficientemente grande.

Variables

Utilizamos un conjunto de variables para hacer las decisiones dentro del modelo, las cuales son presentadas en la Tabla 2.3.

- x_{it} : Número de unidades mantenidas en el inventario i que no serán procesadas al tiempo t
- y_{ijt} : Número de múltiplos ($\alpha_{ij} \Rightarrow \beta_{ij}$) a procesar entre los inventarios i y j al tiempo t
- z_t : Número de unidades incumplidas respecto a la demanda al tiempo t
- φ_i : Costo unitario por mantener inventario en i
- γ_{it} : Variable binaria asociada a la requisición al almacén de materias primas de forma que $\gamma_{it} = \begin{cases} 1 & \text{si el inventario } i \text{ pide material al tiempo } t \\ 0 & \text{si no} \end{cases}$
- δ_{ij}^k : Variable binaria asociada al uso alguna tecnología nueva de forma que: $\delta_{ij}^k = \begin{cases} 1 & \text{si se utiliza la tecnología } k \text{ entre los inventarios } i \text{ y } j \\ 0 & \text{si no} \end{cases}$

Tabla 2.3: Variables para modelos uniproducto

En la optimización del sistema, se penaliza el nivel x_{it} de cada inventario i que no se va a procesar en el tiempo t , de ésta forma de las x_{it} unidades se penalizarán $x_{it} - \sum_j \alpha_{ij} y_{ijt}$ unidades por cada lapso de tiempo t , tomando en cuenta que la penalización aplica siempre y cuando $x_{it} \geq \sum_j \alpha_{ij} y_{ijt}$.

2.2.2. Modelo para sistemas de producción tipo simple

El modelo para sistemas de producción tipo simple uniproducto (SPSU) lo presentamos en la Tabla 2.4.

El modelo SPSU es no lineal, en (I.1) el objetivo es minimizar la suma del costo

$$\text{mín } w = \sum_t \left(\sum_i \varphi_i x_{it} + Cz_t \right) + \sum_t \sum_i Q \gamma_{it} + \sum_i \sum_j \sum_k e_{ij}^k \delta_{ij}^k \quad (\text{I.1})$$

s.a.

$$x_{It} + z_t \geq r_t \quad \forall I, t \quad (\text{I.2})$$

$$\nu_i - \sum_j \alpha_{ij} y_{ij1} = x_{i1} \quad \forall i, (i, j) \in \mathbf{A} \quad (\text{I.3})$$

$$\sum_j \beta_{ji} y_{ji,t-1} + x_{i,t-1} - \sum_j \alpha_{ij} y_{ijt} = x_{it} \quad \forall i < I, \{(i, j), (j, i)\} \in \mathbf{A}, t > 1 \quad (\text{I.4})$$

$$\sum_j \beta_{jI} y_{jIt} + x_{I,t-1} - r_{t-1} = x_{It} \quad I, \forall (j, I) \in \mathbf{A}, t \quad (\text{I.5})$$

$$y_{ijt} \leq b_{ij} + \sum_k \rho_{ij}^k \delta_{ij}^k \quad \forall (i, j) \in \mathbf{A}, t \quad (\text{I.6})$$

$$y_{1jt} \leq M \gamma_{jt} \quad \forall j \in I^1, t \quad (\text{I.7})$$

$$\sum_j \sum_t^{t+\omega} \gamma_{jt} \leq 1 \quad \forall j \in I^1, t \in \{2, \dots, T - \omega\} \quad (\text{I.8})$$

$$\sum_k \delta_{ij}^k \leq 1 \quad \forall (i, j) \in \mathbf{A}^1 \quad (\text{I.9})$$

$$y_{ijt} \in \mathbb{Z}^+ \quad \forall (i, j) \in \mathbf{A}, t \quad (\text{I.10})$$

$$0 \leq x_{it} \leq \bar{x}_i \quad \forall i, t \quad (\text{I.11})$$

$$z_t \geq 0 \quad \forall t \quad (\text{I.12})$$

$$\gamma_{it} \in \{0, 1\} \quad \forall i \in I^1, t \quad (\text{I.13})$$

$$\delta_{ij}^k \in \{0, 1\} \quad \forall k, (i, j) \in \mathbf{A}^1 \quad (\text{I.14})$$

$$\varphi_i \in \mathbb{R} \quad \forall i \quad (\text{I.15})$$

Tabla 2.4: Modelo SPSU

total por mantener producto en proceso, así como la cantidad de demanda incumplida y las requisiciones de materia prima, ésta parte de la función es multiperíodo pues es para cada período de tiempo t . El costo por incurrir en un cambio de tecnología se expresa en la segunda parte de la función objetivo.

Para asegurarnos de cumplir la demanda establecemos (I.2) donde definimos una relación de desigualdad entre lo que se produce y lo que se demanda, lo cual evita infactibilidad en el caso de que la capacidad de la planta sea insuficiente, o bien, infactibilidad por que no exista un múltiplo entre r_t y $\sum_j \beta_{jI} y_{jIt} + x_{I,t-1}$, si se diera éste último caso, un excedente de $\left(\sum_j \beta_{jI} y_{jIt} + x_{I,t-1}\right) - r_t$ unidades de producto terminado, pero el problema es factible.

Usualmente al inicio de un período de producción existe un remante en algún inventario de producto en proceso acumulado en el período previo, el cual debe de ser contemplado para la optimización. El remanente para cada inventario i es de ν_i unidades. La inclusión de este material remanente la hacemos en (I.3).

El balance del flujo de sistema esta dado por (I.4) donde las unidades en inventario i al tiempo t es igual a la suma de lo que llega desde otros inventarios más lo acumulado del período anterior, $t-1$, menos lo que se programa para enviar hacia otros inventarios para el período actual t . Ésta ecuación es muy importante ya que es la que mantiene la integralidad del producto en todo el sistema. (I.4) aplica para todos los inventarios a excepción del inventario I (producto terminado). Las ecuaciones (I.4) y (I.5) son muy similares en su estructura, la diferencia es que en (I.5) en lugar de restar el producto que se manda a producir, como en (I.4), se resta la demanda del período anterior. Restar la demanda nos facilita mucho el manejo de los datos de demanda,

(I.5) solo aplica para el inventario I .

La capacidad de producción de cada estación de trabajo se refleja en (I.6), donde la máxima producción alcanzable por unidad de tiempo es b_{ij} , siempre y cuando no se haga un cambio de tecnología, lo cual aumentaría dicha cantidad.

En (I.7) se amarra el sacar materia prima con su respectivo costo. El personal en el almacén de materias primas necesita cierto tiempo para preparar el pedido y entregar el producto, éste tiempo lo expresamos como ω , por lo que (I.8) toma en cuenta esta consideración. Para el período 1 se contempla la posibilidad de surtir a todas las estaciones que requieran material, así en (I.8) el tiempo corre para $t = \{2, \dots, T - w\}$, por lo que se puede requerir material hasta el tiempo $T - \omega$. Además de lo anterior, se restringe el surtir a solo una estación de trabajo por intervalo de tiempo ω .

Sólo se permite un cambio de tecnología en todo el sistema, esto se expresa por convención en (I.9), lo que se busca es aumentar la capacidad en la operación cuello de botella. En caso de que el cuello de botella este dado por una combinación de estaciones, se puede modificar esta ecuación.

La declaración de variables se realiza en (I.10 - I.15), para éste modelo se tienen una variable entera, dos binarias y 3 variables continuas.

2.2.3. Modelo para sistemas de producción tipo ensamble

Existen dos conjuntos que son necesarios para la modelación de este tipo de sistemas, los cuales separan a los inventarios predecesores a estaciones de trabajo donde se realiza una operación de tipo ensamble de aquellos de los que preceden a una

operación de tipo simple.

I^E : Inventarios que preceden a operaciones de tipo ensamble

I^S : Inventarios que preceden a operaciones de tipo simple

I : Todos los inventarios en el sistema, $I = \{I^E \cup I^S\}$

Tabla 2.5: Conjuntos adicionales para SPEU

α_{iu}^j : Número de unidades necesarias realizar una operación

entre los inventarios i y u , que provienen del inventario j

Tabla 2.6: Parámetros adicionales para SPEU

Con estas definiciones podemos plantear el Modelo II, para sistemas de producción tipo ensamble uniproducto (SPEU) el cual lo presentamos en la Tabla 2.7.

El modelo SPEU al igual que SPSU es no lineal, existen restricciones que son idénticas que en SPSU teniendo el mismo significado, la correspondencia de ellas se ilustra en la Tabla.2.9.

Las restricciones particulares para SPEU son (II.4 - II.6). La combinación de (II.4) y (II.5) mantienen la integralidad del sistema, (II.6) se enfoca a integrar las diferentes cantidades de las distintas materias que se encuentran en el inventario i .

En (II.4) se expresa el balance de flujo para los inventarios del conjunto I^S donde las unidades en inventario en i al tiempo t es igual a la suma de los que llega desde otros inventarios más lo acumulado del período anterior, $t - 1$, menos lo que se programa para enviar hacia otros inventarios $j, j \in I^S$ menos lo que se programa para enviar hacia otros inventarios $k, k \in I^E$.

$$\text{mín } w = \sum_t \left(\sum_i \varphi_i x_{it} + Cz_t \right) + \sum_t \sum_i Q \gamma_{it} + \sum_i \sum_j \sum_k e_{ij}^k \delta_{ij}^k \quad (\text{II.1})$$

$$\text{s.a. } x_{it} + z_t \geq r_t \quad \forall i, t \quad (\text{II.2})$$

$$\nu_i - \sum_j \alpha_{ij} y_{ij1} = x_{i1} \quad \forall i, (i, j) \in \mathbf{A} \quad (\text{II.3})$$

$$\sum_j \beta_{ji} y_{ji,t-1} + x_{i,t-1} - \sum_j \alpha_{ij} y_{ijt} = x_{it} \quad \forall i \in I^S, t > 1, \\ \{(j, i), (i, j)\} \in \mathbf{A} \quad (\text{II.4})$$

$$\sum_j \beta_{ji} y_{ji,t-1} + x_{i,t-1}^j - \sum_u \alpha_{iu}^j y_{iut} = x_{it}^j \quad \forall i \in I^E, t > 1, \\ \{(j, i), (i, u)\} \in \mathbf{A} \quad (\text{II.5})$$

$$\sum_j x_{it}^j = x_{it} \quad \forall i \in I^E, (j, i) \in \mathbf{A} \quad (\text{II.6})$$

$$\sum_j \beta_{jI} y_{jIt} + x_{I,t-1} - r_{t-1} = x_{It} \quad I, \forall (j, I) \in \mathbf{A}, t \quad (\text{II.7})$$

$$y_{ijt} \leq b_{ij} + \sum_k \rho_{ij}^k \delta_{ij}^k \quad \forall i \in I^S, (i, j) \in \mathbf{A}, t \quad (\text{II.8})$$

$$y_{jit}^j \leq b_{ji} + \sum_k \rho_{ji}^k \delta_{ji}^k \quad \forall i \in I^E, (j, i) \in \mathbf{A}, t \quad (\text{II.9})$$

$$y_{1jt} \leq M \gamma_{jt} \quad \forall j \in I^1, t \quad (\text{II.10})$$

$$\sum_j \sum_t^{t+\omega} \gamma_{jt} \leq 1 \quad \forall j \in I^1, t \in \{2, \dots, T - \omega\} \quad (\text{II.11})$$

$$\sum_k \delta_{ij}^k \leq 1 \quad \forall k, (i, j) \in \mathbf{A}^1 \quad (\text{II.12})$$

Tabla 2.7: Modelo SPEU

$$y_{ijt} \in \mathbb{Z}^+ \quad \forall (i, j) \in \mathbf{A}, t \quad (\text{II.13})$$

$$y_{ijt}^i \in \mathbb{Z}^+ \quad \forall (i, j) \in \mathbf{A}, t \quad (\text{II.14})$$

$$0 \leq x_{it} \leq \bar{x}_i \quad \forall i, t \quad (\text{II.15})$$

$$z_t \geq 0 \quad \forall t \quad (\text{II.16})$$

$$\gamma_{it} \in \{0, 1\} \quad \forall i \in I^1, t \quad (\text{II.17})$$

$$\delta_{ij}^k \in \{0, 1\} \quad \forall k, (i, j) \in \mathbf{A}^1 \quad (\text{II.18})$$

$$\varphi_i \in \mathbb{R} \quad \forall i \quad (\text{II.19})$$

Tabla 2.8: Declaración de variables del modelo SPEU

Para los sistemas con tipo de producción de ensamble se identifica desde que inventario por separado para los inventarios $i, i \in I^E$, diferenciado por el inventario desde donde se manda el material, esto queda manifestado en (II.5). La cantidad de producto que se penaliza para los inventarios $i, i \in I^E$, se expresa en (II.6).

Las definiciones de I^1 y de A^1 aplican también para SPEM. Ver definiciones en la página 17.

2.3. Modelos para Sistemas Multiproducto

Es muy común encontrar en la industria sistemas que producen más de un tipo de producto, tanto de características similares como productos totalmente diferentes. Hemos desarrollado una modelación para este tipo de sistemas. La esencia de los criterios de modelación para sistemas multiproducto es la misma que se diseñó para sistemas uniproducto, sin embargo, el sistema de ecuaciones debe ser modificado para cubrir la parte multiproducto del sistema.

(II.1)	→	(I.1)
(II.2)	→	(I.2)
(II.3)	→	(I.3)
(II.7)	→	(I.5)
(II.8, II.9)	→	(I.6)
(II.10)	→	(I.7)
(II.11)	→	(I.8)
(II.12)	→	(I.9)
(II.13, II.14)	→	(I.10)
(II.15)	→	(I.11)
(II.16)	→	(I.12)
(II.17)	→	(I.13)
(II.18)	→	(I.14)
(II.19)	→	(I.15)

Tabla 2.9: Correspondencia de restricciones entre SPSU y SPEU

2.3.1. Nomenclatura

Para la modelación de los sistemas multiproducto hemos empleado nomenclatura muy similar a la empleada en modelos anteriores, para facilitar la explicación y el desarrollo de los modelos siguientes. No detallaremos aquellos conjuntos, parámetros o variables que tengan la misma aplicación que en modelos anteriores.

Conjuntos

I : Conjunto de todos los inventarios en el sistema, donde el inventario 0 representa al almacén de materias primas, I al almacén de producto terminado y para cada $i \neq \{0, I\}$ el i -ésimo inventario en el sistema

A : Matriz de incidencia entre los existentes entre los inventarios del sistema

$$\text{de forma que } a_{ij} = \begin{cases} 1 & \text{si existe conexión entre } i \text{ y } j \\ 0 & \text{si no} \end{cases}$$

P : Conjunto de los diferentes productos que se producen en el sistema

K : Conjunto de las tecnologías disponibles

T : Conjunto del tiempo a optimizar

Los modelos para sistemas uniproductos y multiproductos comparten las definiciones de los conjuntos, a excepción del conjunto **P** el cual contiene a todos los p productos terminados que se producen en el sistema.

Parámetros

Es necesario redefinir algunos parámetros dada la gama de productos del sistema, los cuales son compartidos en los sistemas multiproducto, éstos parámetros se mues-

tran en la tabla siguiente.

- α_{ij}^p : Número de unidades necesarias realizar una operación entre los inventarios i y j
- β_{ij}^p : Número de unidades que se producen por cada operación entre los inventarios i y j
- r_t^p : Demanda al tiempo t
- ν_i^p : Número de unidades en el inventario i al inicio del período
- C^p : Costo asociado por unidad incumplida, respecto a la demanda

Tabla 2.10: Parámetros para sistemas multiproducto

Los *coeficientes de transferencia* deben ser establecidos para cada uno de los p productos debido a las diferentes necesidades (espacio, tasas de consumo, etc.) que se manejan en el sistema, así α_{ij}^p representa el número de unidades necesarias del inventario i para realizar una operación entre los inventarios i y j del producto p , y por cada operación tenemos que un número β_{ij}^p de unidades resultantes del producto p entre los inventarios i y j . Cabe recordar que éstos parámetros deben ser enteros.

La demanda es expresada para cada producto p en cada unidad de tiempo t , siendo $r_t^p > 0$ dicho número. El remanente de material en proceso en el inventario i del producto p que pudiese quedar entre diferentes turnos (diferentes corridas del modelo), es denotado por ν_i^p . Quedar por debajo de los requerimientos de producto terminado tiene una penalización de C^p por cada unidad incumplida del producto p .

Variables

De forma similar a los parámetros, las variables son redefinidas para soportar la gama de productos en el sistema, las modificaciones a las variables son definidas en 2.11.

- x_{it}^p : Número de unidades mantenidas en el inventario i del producto p que no serán procesadas al tiempo t
- y_{ijt}^p : Número de múltiplos ($\alpha_{ij}^p \Rightarrow \beta_{ij}^p$) a procesar entre los inventarios i y j al tiempo t del producto p
- z_t^p : Número de unidades incumplidas del producto p respecto a la demanda al tiempo t
- φ_i^p : Costo unitario por mantener inventario en i del producto p
- γ_{it} : Variable binaria asociada a la requisición al almacén de materias primas de forma que $\gamma_{it} = \begin{cases} 1 & \text{si el inventario } i \text{ pide material al tiempo } t \\ 0 & \text{si no} \end{cases}$
- δ_{ij}^k : Variable binaria asociada al uso alguna tecnología nueva de forma que: $\delta_{ij}^k = \begin{cases} 1 & \text{si se utiliza la tecnología } k \text{ entre los inventarios } i \text{ y } j \\ 0 & \text{si no} \end{cases}$

Tabla 2.11: Variables para modelos multiproducto

2.3.2. Sistemas de tipo de producción simple

El modelo multiproducto para sistemas de tipo de producción simple se muestra en la Tabla 2.12

$$\text{mín } w = \sum_t \left(\sum_p \sum_i \varphi_i^p x_{it}^p + C^p z_t^p \right) + \sum_t \sum_i Q \gamma_{it} + \sum_i \sum_j \sum_k e_{ij}^k \delta_{ij}^k \quad (\text{III.1})$$

s.a.

$$x_{it}^p + z_t^p \geq r_t^p \quad \forall i, p, t \quad (\text{III.2})$$

$$\nu_i^p - \sum_j \alpha_{ij}^p y_{ij1}^p = x_{i1}^p \quad \forall i, p, (i, j) \in \mathbf{A} \quad (\text{III.3})$$

$$\sum_j \beta_{ji}^p y_{ji,t-1}^p + x_{i,t-1}^p - \sum_j \alpha_{ij}^p y_{ijt}^p = x_{it}^p \quad \forall i < I, p, \{(i, j), (j, i)\} \in \mathbf{A} \quad (\text{III.4})$$

$$t > 1$$

$$\sum_j \beta_{jI}^p y_{jIt}^p + x_{I,t-1}^p - r_{t-1}^p = x_{It}^p \quad I, \forall p, (j, I) \in \mathbf{A}, t \quad (\text{III.5})$$

$$\sum_p \lambda_{ij}^p y_{ijt}^p \leq b_{ij} + \sum_k \rho_{ij}^k \delta_{ij}^k \quad \forall p, (i, j) \in \mathbf{A}, t \quad (\text{III.6})$$

$$y_{1jt}^p \leq M \gamma_{jt} \quad \forall p, j \in I^1, t \quad (\text{III.7})$$

$$\sum_j \sum_t^{\omega} \gamma_{jt} \leq 1 \quad \forall j \in I^1, \quad (\text{III.8})$$

$$t \in \{2, \dots, T - \omega\}$$

$$\sum_k \delta_{ij}^k \leq 1 \quad \forall (i, j) \in \mathbf{A}^1 \quad (\text{III.9})$$

$$0 \leq \sum_p \eta^p x_{it}^p \leq \bar{x}_i \quad \forall i, p, t \quad (\text{III.10})$$

$$y_{ijt}^p \in \mathbb{Z}^+ \quad \forall p, (i, j) \in \mathbf{A}, t \quad (\text{III.11})$$

$$z_t^p \geq 0 \quad \forall p, t \quad (\text{III.12})$$

$$\gamma_{it} \in \{0, 1\} \quad \forall i \in I^1, t \quad (\text{III.13})$$

$$\delta_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in \mathbf{A}^1, k \quad (\text{III.14})$$

$$\varphi_i^p \in \mathbb{R} \quad \forall i, p \quad (\text{III.15})$$

$$\lambda_{ij}^p \geq 0 \quad \forall p, (i, j) \in \mathbf{A} \quad (\text{III.16})$$

Tabla 2.12: Modelo SPSM

El modelo SPSM es no lineal, la primera parte de la función objetivo de (III.1) es minimizar la suma del costo total por mantener inventario de producto en proceso de los diferentes productos en el sistema, en la primer parte también buscamos minimizar el número de unidades incumplidas de cada producto de acuerdo a su respectiva demanda, además de usar un número mínimo de requisiciones de materia prima generadas dentro del horizonte de optimización. El costo por incurrir en un cambio de tecnología se expresa en la segunda parte de la función objetivo, esta parte de la función objetivo no es afectada por los diferentes tipos de producto.

Se define una desigualdad para satisfacer la demanda de cada producto, así lo establecemos para soportar la posible falta de capacidad por parte de sistema, o bien, para soportar el hecho de que no exista un múltiplo entre r_t^p y $\sum_j \beta_{jI}^p y_{jIt}^p + x_{I,t-1}^p$, para cada tipo de producto.

El remanente de producto en proceso para cada uno de los productos del sistema entran a la optimización como ν_i^p unidades, esto se establece en (III.3)

El balance del flujo de sistema esta dado por (III.4) donde las unidades en inventario i del producto p al tiempo t es igual a la suma de lo que llega para cada producto p desde otros inventarios más lo acumulado de cada producto p del período anterior, $t - 1$, menos lo que se programa para enviar hacia otros inventarios para cada producto p para el período actual t . En ésta ecuación se plasmamos la integralidad del sistema donde el producto fluye entre las estaciones de trabajo hasta terminar como producto terminado. (III.4) aplica para todos los inventarios a excepción del inventario I (productos terminados). La estructura de (III.4) y (III.5) son muy parecidas, la diferencia es que en (III.5) en lugar de restar para cada producto p la cantidad que

se manda a producir, como en (III.4), restamos la demanda del período anterior. Esto lo hacemos para facilitar el manejo de los datos. (III.5) solo aplica para el inventario I de productos terminados.

La capacidad de producción de cada estación de trabajo se refleja en (III.6), donde la máxima producción de la estación de trabajo alcanzable por unidad de tiempo es b_{ij} , lo anterior considerando una mezcla de producción de $\sum_p y_{ijt}^p$ para cada unidad de tiempo t , donde λ_{ij}^p es la tasa de consumo por producir una unidad del producto p entre los inventarios i y j . Esta capacidad esta puede estar sujeta a un posible cambio de tecnología, lo cual aumentaría dicha cantidad.

Hacer requisiciones de materia prima conlleva un costo, esto se establece en (III.7). Otra consideración es que consideramos un tiempo en el cual el personal del almacén de materias primas necesita para preparar una requisición, dicho tiempo se considera como ω , esto se refleja en (III.8).

Consideramos que para el período inicial del horizonte de optimización se contempla la posibilidad de abastecer de materia prima a todas las estaciones que requieran material, así en (III.8), donde el tiempo aplica para $t = \{2, \dots, T - \omega\}$, por lo que se puede requerir material hasta el tiempo $T - \omega$. Además de lo anterior, se restringe el surtir a solo una estación de trabajo por intervalo de tiempo ω .

Por consideraciones de análisis se considera solo un posible cambio de tecnología en todo el sistema, esto se expresa en (III.9), lo que se busca es aumentar la capacidad en la operación cuello de botella. En caso de que el cuello de botella este dado por una combinación de estaciones, se puede modificar esta ecuación, en ambos casos es necesario un análisis posterior, la cual no se incluye en esta tesis.

La capacidad de cada inventario esta limitada por \bar{x}_i permitiendo una mezcla $\sum_p \eta^p x_{it}^p \leq \bar{x}_i$, donde η^p represente el consumo de espacio del producto p , lo anterior lo restringimos en (III.10).

La declaración de variables se realiza en (III.11 - III.16), dónde tenemos una variable entera, dos binarias y cuatro variables continuas, de las cuáles tres son no-negativas.

2.3.3. Sistemas de tipo de producción de ensamble

El modelo multiproducto para sistemas de tipo de producción simple se muestra en la Tabla 2.13

El modelo SPEM en no lineal, como todos los modelos que se han planteados en esta tesis, los modelos para sistemas multiproducto son semejantes en su estructura de modelación por lo que comparten ecuaciones, la correspondencia entre ecuaciones semejantes se ilustra en la Tabla.2.15.

Las restricciones particulares para SPEM son (IV.4 - IV.6). Debido a la estructura de los modelos que presentamos en esta tesis, la combinación de (IV.4) y (IV.5) mantienen la integralidad de producto en el sistema que fluye a través de los inventarios i , para $i \in I$, donde $I^S \cup I^E = I$. La capacidad del inventario i se establece en (IV.6), donde permitimos mantener una mezcla de productos $\sum_j x_{it}^{jp}$ que en total componen la utilización x_{it}^p para los inventarios $i \in I^E$ para todos los inventarios $(j, i) \in \mathbf{A}$.

El balance del flujo para los inventarios $i \in I^S$ se establece en (IV.4), donde las

$$\min w = \sum_t \left(\sum_p \sum_i \varphi_i^p x_{it}^p + C^p z_t^p \right) + \sum_t \sum_i Q \gamma_{it} + \sum_i \sum_j \sum_k e_{ij}^k \delta_{ij}^k \quad (\text{IV.1})$$

s.a.

$$x_{it}^p + z_t^p \geq r_t^p \quad \forall i, p, t \quad (\text{IV.2})$$

$$\nu_i^p - \sum_j \alpha_{ij}^p y_{ij1}^p = x_{i1}^p \quad \forall i, p, (i, j) \in \mathbf{A} \quad (\text{IV.3})$$

$$\sum_j \beta_{ji}^p y_{ji,t-1}^p + x_{i,t-1}^p - \sum_j \alpha_{ij}^p y_{ijt}^p \quad \forall i \in I^S, t > 1, \\ \{(j, i), (i, j)\} \in \mathbf{A}, p \quad (\text{IV.4})$$

$$\sum_j \beta_{ji}^p y_{ji,t-1}^p + x_{i,t-1}^{jp} - \sum_u \alpha_{iu}^{jp} y_{iut}^p = x_{it}^{jp} \quad \forall i \in I^E, p, t > 1 \\ \{(j, i), (i, u)\} \in \mathbf{A} \quad (\text{IV.5})$$

$$\sum_j x_{it}^{jp} = x_{it}^p \quad \forall i \in I^E, p, (j, i) \in \mathbf{A} \quad (\text{IV.6})$$

$$\sum_j \beta_{jI}^p y_{jIt}^p + x_{I,t-1}^p - r_{t-1}^p = x_{It}^p \quad I, \forall p, (j, I) \in \mathbf{A}, t \quad (\text{IV.7})$$

$$\sum_p \lambda_{ij}^p y_{ijt}^p \leq b_{ij} + \sum_k \rho_{ij}^k \delta_{ij}^k \quad \forall p, i \in I^S, (i, j) \in \mathbf{A}, t \quad (\text{IV.8})$$

$$\sum_p \lambda_{ji}^p y_{jit}^{jp} \leq b_{ji} + \sum_k \rho_{ji}^k \delta_{ji}^k \quad \forall p, i \in I^E, (j, i) \in \mathbf{A}, t \quad (\text{IV.9})$$

$$y_{1jt}^p \leq M \gamma_{jt} \quad \forall p, j \in I^1, t \quad (\text{IV.10})$$

$$\sum_j \sum_t^{\omega} \gamma_{jt} \leq 1 \quad \forall j \in I^1, t \in \{2, \dots, T - \omega\} \quad (\text{IV.11})$$

$$\sum_k \delta_{ij}^k \leq 1 \quad \forall (i, j) \in \mathbf{A}^1 \quad (\text{IV.12})$$

$$0 \leq \sum_p \eta^p x_{it}^p \leq \bar{x}_i \quad \forall i, p, t \quad (\text{IV.13})$$

Tabla 2.13: Modelo SPEM

$$y_{ijt}^p \in \mathbb{Z}^+ \quad \forall (i, j) \in \mathbf{A}, t \quad (\text{IV.13})$$

$$y_{ijt}^{ip} \in \mathbb{Z}^+ \quad \forall (i, j) \in \mathbf{A}, t \quad (\text{IV.14})$$

$$z_t \geq 0 \quad \forall t \quad (\text{IV.15})$$

$$\gamma_{it} \in \{0, 1\} \quad \forall i \in I^1, t \quad (\text{IV.16})$$

$$\delta_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in \mathbf{A}^1, k \quad (\text{IV.17})$$

$$\varphi_i \in \mathbb{R} \quad \forall i \quad (\text{IV.18})$$

$$\lambda_{ij}^p \geq 0 \quad \forall p, (i, j) \in \mathbf{A} \quad (\text{IV.19})$$

Tabla 2.14: Declaración de variables del modelo SPEM

unidades en inventario en i al tiempo t del producto p es igual a la suma de las unidades que llegan desde otros inventarios más el acumulado de unidades remanente de cada producto p del período anterior, $t - 1$, menos lo que se programa para enviar hacia otros inventarios $j, j \in I^S$ menos lo que se programa para enviar hacia otros inventarios $k, k \in I^E$ para cada producto p .

Para los sistemas con tipo de producción de ensamble se identifica desde que inventario se envía el producto que se ensamblará para cada producto p por separado para todos los inventarios $i, i \in I^E$, manteniendo identificado desde que inventario se recibo material para cada producto p , ésta relación la expresamos en (IV.5).

Las definiciones de I^1 y de A^1 aplican también para SPEM. Las definiciones se pueden consultar en la página 17.

(IV.1)	→	(III.1)
(IV.2)	→	(III.2)
(IV.3)	→	(III.3)
(IV.7)	→	(III.5)
(IV.8, IV.9)	→	(III.6)
(IV.10)	→	(III.7)
(IV.11)	→	(III.8)
(IV.12)	→	(III.9)
(IV.13, IV.14)	→	(III.10)
(IV.15)	→	(III.11)
(IV.16)	→	(III.12)
(IV.17)	→	(III.13)
(IV.18)	→	(III.14)
(IV.19)	→	(III.15)

Tabla 2.15: Correspondencia de restricciones entre SPSM y SPEM

2.4. Producir contra Inventario.

Cuando el producto tiene gran demanda y se está seguro de que lo que se produce se vende, es deseable producir contra inventario. Debido a la gran demanda del producto es deseado producir tanto como sea posible utilizando gran parte, si no es que toda, de la capacidad instalada del sistema.

2.5. Producir contra Demanda.

La demanda puede ser específica para cada tiempo, teniendo así un sistema que produce contra demanda. Para esta modalidad es deseable que la suma de las demandas en cada tiempo sea cubierta por la capacidad de la planta multiplicada por dicho tiempo, $\sum_1^t r_t \leq t \cdot (\text{Capacidad de producción})$, de otra forma z_t será positiva.

2.6. Método de Solución

En contraste con los problemas de programación lineal, los problemas formulados en programación entera tienen mayor dificultad para resolverlos. Los modelos desarrollados en esta tesis fueron declarados como no-lineales, para resolverlos definimos un esquema de costos de inventarios en el Capítulo 3, en la sección de Coeficientes de Costos, con las consideraciones en dicho capítulo permiten utilizar el método de solución descritos en esta sección.

No existe un algoritmo de aplicación general que pueda resolver este tipo de problemas. La falta de este algoritmo general da cabida a intentar resolver este tipo de

problemas mediante varios tipos de algoritmos. Existen tres categorías principales de algoritmos de solución: [3]

- (a) Algoritmos Exactos, estos aseguran la obtención de una solución óptima a expensas de un número alto de iteraciones. Dentro de este grupo se encuentran los algoritmos planos cortantes, ramificación - cota, ramificación-corte y programación dinámica.
- (b) Algoritmos de Aproximación, estos obtienen una solución subóptima en un tiempo polinomial junto con una cota respecto al grado de suboptimalidad presente.
- (c) Algoritmos heurísticos, proveen una solución subóptima, pero sin garantizar su calidad, además su tiempo de ejecución no se garantiza que sea polinomial, estudios empíricos apuntan a que en la mayoría de los casos se logra una buena solución rápidamente, algunos ejemplos de estos métodos son las búsquedas locales, el recocido simulado, la búsqueda TABU, los algoritmos genéticos, entre otros.

En la búsqueda de la solución de nuestra formulación se utiliza la técnica de ramificación-cota, la cual es muy utilizada en las herramientas computacionales diseñadas para resolver problemas de este tipo. Como se explica más adelante, las herramientas computacionales logran implementar estos algoritmos con éxito, además de resolver problemas muy grandes en un lapso corto tiempo.

2.6.1. Herramientas Computacionales

Aunque las técnicas de solución puedan realizarse a “mano”, en la realidad es difícil encontrar problemas que se puedan solucionar de forma manual, el apoyo de las herramientas computacionales ha permitido a través de los años incrementar la dificultad de los problemas a resolver, el avance en las tecnologías computacionales a permitido resolver problemas mas complejos que requieran de mas cálculos, en períodos razonables de tiempo.

Para solucionar los modelos matemáticos planteados en esta tesis se utilizaron dos herramientas computacionales, la primera es el lenguaje modelador, este programa es utilizado en la creación de los modelos matemáticos de forma que la computadora pueda interpretarlos y así poder resolverlos. Existe gran variedad de modeladores matemáticos entre los que se encuentran LINDO, GAMMS, AMPL, XPress MP, etc [1]. Un lenguaje modelador toma la formulación matemática anteriormente descrita y transforma en un conjunto de instrucciones que mas adelante serán interpretadas por otro programa conocido como el motor de solución (solver), esta es la segunda herramienta es necesaria para la optimización, ya que es la que realiza las operaciones para resolver los problemas.

Uno de los motores de solución más conocido y robusto es CPLEX de Ilog, este motor de solución contiene los algoritmos de solución para LP, MIP, QP, además de este existen otros motores como el LINGO, MOSEK, SOPT, XPress con variantes de capacidad respecto al tipo de problemas que pueden resolver. En el artículo [1] se describe una encuesta actualizada del software disponible para solucionar los problemas

de la programación lineal.

Como lenguaje modelador para resolver nuestros problemas, hemos seleccionado AMPL por ser una herramienta en la que fácilmente se puede implementar cualquier formulación matemática propuesta.

En AMPL la formulación matemática se divide en dos elementos, uno el modelo matemático, es decir, la función objetivo, las restricciones y las variables, el otro elemento son los datos concretos del problema a resolver. Estos dos elementos se guardan por separado en dos archivos uno es el *modelonombre.mod* y el que contiene los datos *modelonombre.dat*.

Una vez que se tiene la formulación del problema en AMPL se resuelve mediante el solver CPLEX de Ilog, en nuestro caso CPLEX implementa un algoritmo de Branch and Bound. Para mas detalle sobre AMPL y CPLEX en la sección de Apéndices, se encuentra un resumen con lo más relevante de éstas dos herramientas.

Capítulo 3

Datos

3.1. Importancia de los Datos.

Los parámetros (datos) de entrada son de suma importancia para los modelos, ya que la calidad de los resultados dependerán totalmente de la calidad de los datos. El buen uso de los datos de entrada repercute totalmente en los resultados.

Para los modelos desarrollados en esta tesis son necesarios parámetros que son asignados por el usuario, entre ellos se encuentran los costos por mantener inventario, la amplitud del intervalo de tiempo t . Otros datos son en base a las capacidades tecnológicas del sistema en análisis, pudiendo mencionar los coeficientes α y β , los parámetros b_{ij} , entre otros.

Como hemos mencionado antes, los cuatro modelos desarrollados en esta tesis son no-lineales, lo que complica en extremo su solución. Una forma de evitar la no-linealidad de los modelos es mediante el establecimiento de parámetros de costo por mantener producto en los inventarios, el parámetro de costo en la función objetivo es

φ_i^p , donde el súper índice p aplica obviamente para los sistemas multiproducto.

En éste capítulo desarrollamos una opción para establecer coeficientes de parámetros para los diferentes modelos, además de recomendaciones para su aplicación.

3.2. Horizonte de Tiempo a Optimizar

El horizonte a optimizar depende del período que se desea analizar, las divisiones de tiempo dependen de las capacidades singulares de cada sistema. Por ejemplo si se desea optimizar un período de un turno laboral, de 8 hrs., supongamos que entre tiempo de comida, mantenimiento y limpieza restan 7 horas para la producción, hagamos intervalos de tiempo de 15 minutos, es decir, el parámetro T tomará el valor de 28, así, t tendría los valores de $\{1, \dots, 28\}$. Entre menos intervalos de tiempos se consideren en la optimización, la estructura del modelo tendrá menor número de variables, por lo tanto, el tiempo para entregar la solución será menor. Hay que tener en cuenta que es posible que entre menor sea T se pueda perder realidad en la modelación. Todo esto depende de la particularidad de cada sistema.

Los intervalos de tiempo deben considerar las capacidades del sistema, en el sentido de que hay que buscar un múltiplo en los tiempos de producción de forma que las diferencias entre t y un múltiplo de los tiempos de producción sean lo menos posible.

Supongamos que en nuestro sistema hemos definido t como intervalo de 10 minutos, ahora, supongamos también que una de las estaciones de trabajo del sistema trabaja a una tasa de 12 ciclos por hora, esto resulta en que cada ciclo dura 5 minutos. Tomando en cuenta el intervalo que definimos para t , la estación de trabajo puede

realizar 2 ciclos por cada intervalo de tiempo.

Supongamos que tenemos tres estaciones de trabajo con las características como se muestra en la Tabla 3.1. A medida que el número de estaciones del sistema de producción sea mayor resulta más complicado conciliar un número para los intervalos de t , esto es, que sea difícil el que las tasas de producción de todas las estaciones de trabajo coincidan en un número ‘pequeño’. Para manejar esto, recomendamos buscar que las estaciones de trabajo con las tasas de producción mas pequeñas, es decir, con menores ciclos/hora, tengan un número entero para los $\#Ciclos/t$, y redondear al entero menor aquellas estaciones de trabajo con tasas mayores.

$t : 15$ minutos			
Estación de Trabajo	Ciclos/Hora	Minutos/Ciclo	$\#Ciclos/t$
E1	12	5	3
E2	16	3.75	4
E3	8	7.5	2

Tabla 3.1: Ejemplo: Parametrización de T

El mejor horizonte de planeación se debe calcular para cada sistema, hay que tomar en cuenta el detalle que se desea en cuanto al número de intervalos, así como al detalle de las capacidades por cada intervalo de tiempo.

3.3. Coeficientes de Transferencia (α y β)

Los coeficientes de transferencia que hemos utilizado en el desarrollo de los modelos en esta tesis es la forma en la que estructuramos el flujo de material a través de las estaciones de trabajo, nuestros coeficientes de transferencia α y β . Éstos son parámetros que deben ser definidos con cuidado y deben ser estrictamente enteros positivos.

Los coeficientes de transferencia aplican para cuando existen estaciones de trabajo donde la relación entre unidades entrantes difiere de las salientes, por ejemplo, consideremos que en una operación donde necesita tres unidades de material para trabajar, en la operación se producen cinco unidades, para este ejemplo $\alpha = 3$ y $\beta = 5$. De igual manera es posible que para algunas operaciones sea necesario introducir, digamos, 8 unidades de material, para producir 3 unidades, en este caso $\alpha = 8$ y $\beta = 3$.

Éstos parámetros tiene el objetivo de interpretar las tasas de producción de cada estación de trabajo en cantidad de unidades de producto en proceso necesarias y producidas por intervalo de tiempo t , de forma que los modelos optimicen sobre la realidad del sistema de producción. Buscamos que los modelos trabajen sobre una abstracción de la realidad y que al mismo tiempo, el sistema de ecuaciones sea lo más sencillo posible.

3.4. Coeficientes de Costos

La definición de los parámetros de costos para los modelos desarrollados son fundamentales para la calidad de la solución. Es posible definir parámetros con fines de

simulación del sistema, para identificar las estaciones de trabajo críticas.

En esta sección hacemos una introducción a los coeficientes de costos utilizados en el desarrollo de los modelos presentados en esta tesis.

3.4.1. Costo por generar salidas de almacén

El costo por requisición de materia prima Q es aplicado cada vez que una orden es colocada, independientemente de la cantidad requerida.

El valor de Q tiene la función de programar un requerimiento de material siempre y cuando $\sum_i \varphi_i x_{it} > Q$, para $t > t^R$, donde t^R es el tiempo t donde se plantea la requisición. En otras palabras, el sistema de ecuaciones está diseñado para programar salidas de materia prima siempre y cuando el costo por mantener producto en proceso en los inventarios sea mayor a Q .

Un valor grande para este parámetro implica que cada que sea requerida materia prima, esta será en grandes cantidades, además que el tiempo entre requisiciones puede ser amplio. Un valor pequeño para Q generará un número de requisiciones grande, con salida de producto necesario para períodos cortos de tiempo.

3.4.2. Costo por mantener inventario

Inicialmente, los coeficientes de costo por mantener inventario está declarado como una variable en la modelación de los sistemas, lo cual hace a los modelos no-lineales, dificultando su método de solución.

3.4.3. Costo por uso de nueva tecnología

Las diferentes k tecnologías disponibles deben tener su costo relacionado al beneficio que pueden ofrecer al sistema. Estos costos tienen como variable el tiempo, ya que especificamos que si se necesita el uso de alguna tecnología, esta estará disponible en todo el horizonte de optimización.

Como diferentes tecnologías queremos decir el uso, en ciertas estaciones de trabajo, de nueva maquinaria, personal extra o cualquier otra modificación que incremente la capacidad de producción de las estaciones de trabajo donde se contemplen los posibles cambios. De forma general en todos los modelos permitimos un solo cambio de tecnología, lo cual dos objetivos:

- Evaluar el impacto de aumentar la capacidad de producción en ciertas estaciones de trabajo
- Identificar en que estación de trabajo es más imperativo mayor capacidad de producción

De manera sencilla es posible asignar un costo de cero para el uso de las diferentes tecnologías y permitir el cambio para todas las estaciones de trabajo. Al correr la optimización con este sencillo escenario, permitirá identificar la operación de cuello de botella, como mencionamos anteriormente, es posible que en sistemas complejos con configuración de red, el cuello de botella ese dado por una combinación de estaciones de trabajo, considerando esto, recomendamos para este tipo de sistemas correr el ejercicio permitiendo que $\sum_k \delta_{ij}^k \leq n$, donde n representa el número incrementos de capacidades de producción en las estaciones de trabajo.

El hacer uso de algún cambio de tecnología permite:

- Disminuir el costo por mantener inventario
- Disminuir la demanda incumplida, en caso de que aplique
- Aumentar la capacidad de producción del sistema

El incremento en capacidad de estaciones de trabajo críticas permite un mayor flujo de producto en proceso el cual se refleja directamente en la disminución del costo de mantener inventario, además de disminuir el número de unidades incumplidas de la demanda, en caso de que hubiese.

El uso de nuevas tecnologías se incentiva cuando el costo de mantener inventario es mayor al beneficio en el que se incurre al hacer uso de las nuevas tecnologías, o bien, el costo por la demanda incumplida más el costo por mantener inventario es mayor que el costo del uso de alguna nueva tecnología para la estación de trabajo más crítica.

3.4.4. Costo por incumplir demanda

En la mayoría de los sistemas de producción es más costoso incurrir en incumplimiento de unidades demandadas que incurrir en los costos de mantener inventario de producto en proceso, esto por varias razones entre las cuales podemos mencionar las siguientes:

- Incumplir demanda genera desagrado con los clientes, por lo que es posible que busquen otro proveedor del producto faltante

- En caso de que nuestro producto terminado sea un insumo para la fabricación de otros productos, es posible que existan contratos donde se penalice el desabasto de insumos, tales penalizaciones suelen ser costosas
- Los productos terminados entregados fuera de tiempo corren el riesgo de no ser aceptados por el cliente, produciendo pérdidas económicas

Una cota inferior para el costo de incumplir una unidad es $\sum_i \varphi_i \bar{x}_i < C$, el costo por incumplir una unidad de producto terminado debe ser mayor a la suma del producto de la capacidad de cada inventario por el costo de mantener inventario.

Capítulo 4

Metodología

4.1. Identificar el cuello de botella del sistema

El cuello de botella se refiere a la operación que limita la capacidad total del sistema. Es importante identificar las estaciones de trabajo que limitan la capacidad del sistema, pues estas estaciones requieren de especial atención, parámetros como t son definidos en base a las capacidades de este tipo de operaciones.

La complejidad para encontrar el cuello de botella es más sencilla en sistemas uniproducto con tipo de producción simple y flujo en serie que en sistemas multiproducto con tipo de producción ensamble y flujo en red.

Hemos encontrado que para sistemas con flujo en red, es posible tener más de una estación de trabajo que limite la capacidad total del sistema.

Para identificar el cuello de botella para cualquier tipo de sistema se asigna una demanda suficientemente alta, la cual esta definida según las características del sistema, asignado un costo positivo al almacén de materias primas, costo de cero a los

inventarios de producto en proceso y un costo negativo al almacén de producto terminado. Antes de realizar la corrida, debemos habilitar un posible cambio de tecnología en todas las estaciones de trabajo del sistema. Una vez definidos estos parámetros es necesario realizar una corrida de modelo donde la operación cuello será aquella donde el cambio de tecnología sea activado por el modelo. Sería recomendable identificar las operaciones que también sean críticas además de la operación cuello de botella, esto se puede hacer permitiendo más de un cambio de tecnología.

Para sistemas lineales el cuello de botella esta dado por una sola estación de trabajo, esto puede diferir en sistemas con configuración de red, donde el cuello de botella puede estar dado por más de una estación de trabajo. El cuello de botella puede estar dado por tres factores o una combinación de ellos:

- Por el espacio disponible si la capacidad de la estación de trabajo sobrepasa la capacidad del espacio disponible.
- La estación de trabajo que tenga el mínimo cociente de $\frac{\sum_j \alpha_{ij} b_{ij}}{\sum_j \beta_{ji} b_{ji}}$ o bien que siendo el mínimo la estación siguiente no tenga la capacidad para procesar ese mínimo de la estación.
- El espacio disponible para una mezcla de productos sea insuficiente.

Para sistemas multiproducto es posible que existan estaciones cuello de botella para cada producto o inclusive para ciertas combinaciones de demanda de productos. Identificar el cuello de botella es de suma importancia ya que el inventario de producto en proceso anterior a esta estación debe de estar siempre con material disponible. El

cuello de botella en sistemas de red multiproducto puede ser diferente dependiendo de la mezcla de producto demandado. En estos casos es necesario simular varias mezclas de demanda para identificar estaciones de trabajo críticas en cada escenario de demanda simulada que se plantearon.

Conocer las estaciones de trabajo que realmente limitan la capacidad de todo el sistema da una visión importante al momento de planear inversiones, pues se tiene visibilidad de que estaciones de trabajo son las que necesitan capacidad.

4.2. Reconocer la capacidad máxima instalada

La capacidad máxima instalada en el sistema puede ser identificada aprovechando la estructura de la modelación, donde se pueden manipular los datos de manera similar al identificar las estaciones críticas para el sistema, la manipulación de datos que recomendamos se expone en los siguientes párrafos.

Asignar una demanda suficientemente alta, tomando en cuenta las características de cada sistema, asignar un costo positivo al almacén de materias primas, costo de cero a los inventarios de producto en proceso y un costo negativo a el almacén de producto terminado. Con cantidad suficiente de materia prima asignada, realizamos una corrida del modelo. La capacidad máxima instalada será reflejada en el aumento de producto terminado en el inventario de producto terminado por cada intervalo de tiempo t .

La capacidad máxima instalada para sistemas multiproducto puede estar sujeta a la mezcla de la demanda, debido a que las estaciones de trabajo produce a diferentes

tasas. Podemos modelar diferentes escenarios, el costo negativo por unidad de cada producto p dará las prioridades en el modelo al tiempo de asignar utilización de recursos en las estaciones de trabajo.

Tener conciencia de la capacidad máxima instalada es de suma importancia al momento de la planeación, además de se de gran apoyo al momento de pactar compromisos de parte de producción.

4.3. Establecer el número de salidas de almacén

Una de las consideraciones incluidas en esta tesis es que el número de entregas por parte del almacén de materias primas está limitado por alguna razón, las materias primas son de difícil manejo o simplemente el personal del almacén es limitado.

Contamos con dos criterios que van a definir el tiempo y la cantidad de materia prima que será surtida a las estaciones de trabajo. El primero es limitar el número de veces que se pueden hacer requisiciones de cualquier estación de trabajo. El segundo tiene que ver con el tiempo que es necesario para surtir una orden colocada.

El primer criterio no esta considerado dentro de la modelación pero es sencillo incluirlo, para esto podemos incluir la restricción:

$$\sum_j \sum_t \gamma_{jt} \leq \varepsilon, \forall j \in I^1, t \in \{2, \dots, T - \omega\}$$

Donde ε es el número de salidas de materias primas permitidas en el sistema.

Esto es algo muy importante por que damos la pauta para poder satisfacer materias primas al sistema de producción considerando las capacidades de surtido del

almacén. Otra consideración importante es que limitamos el envío de materia prima a sólo una estación de trabajo por cada vez que se programa un envío de materia prima para cada intervalo $t + \omega$.

4.4. Diferentes Tecnologías

Las diferentes tecnologías que se han mencionado en los modelos de esta tesis se refieren a cualesquiera de maquinaria adicional o personal extra asignados a cierta estación de trabajo. Al incluir esto es posible simular varios escenarios con los diferentes cambios de tecnología disponibles en el sistema para las estaciones de trabajo que sean críticas para el sistema.

Con el fin de facilitar la resolución de los problemas, el análisis de los posibles cambios debe hacerse con escenarios de simulación, una vez tomada la decisión si se realizan o no cambios y para qué estaciones de trabajo aplican dichos cambios. Es recomendable no permitir ningún cambio, es decir, modificar que los cambios sean iguales a cero, es decir:

$$\delta_{ij}^k = 0, \forall (i, j) \in \mathbf{A}^1$$

Así, los recursos tecnológicos para resolver los problemas serán concentrados para la programación y asignación de recursos de producción para satisfacer la demanda dada.

Capítulo 5

Experimentación

5.1. Experimentación

En este capítulo se muestran los resultados de la experimentación de los problemas generados de forma aleatoria, se emplea una sistema con configuración de red de tipo de producción simple, donde realizamos análisis en los parámetros T , ω y \bar{x} . Se generó un sistema de red con:

$$I = 17$$

$$|\mathbf{A}| = 22$$

Los problemas generados siempre contaron con las consideraciones siguientes:

- El inventario de materia prima disponible es suficiente para la producción dentro del período de optimización.
- La demanda de producto se concentró en el tiempo T , donde el producto demandado es un poco mayor que la capacidad del sistema.

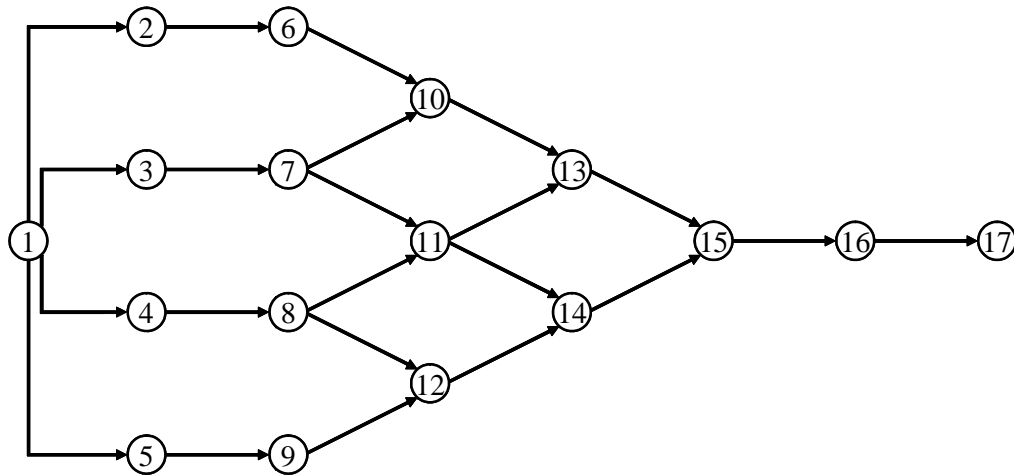


Figura 5.1: Red generada para la experimentación

Las consideraciones anteriores fueron hechas para modelar condiciones de trabajo donde el sistema opere a su máxima capacidad.

El sistema de producción generado se muestra en la Figura 5.1.

Las capacidades de producción se muestran en la Tabla 5.1. En este sistema todos los coeficientes α y β son iguales a uno, es decir, por cada unidad de materia prima que entra a una estación, esta genera una unidad producida.

Es necesario definir una nomenclatura para las corridas realizadas, de esta manera podemos consultar los archivos de forma estructurada. La nomenclatura empleada para los archivos de salida se muestra en la Tabla 5.2. Por ejemplo, la nomenclatura SPSU-T30-I17-A22-W5-X_M200, se refiere a un sistema de tipo de producción simple uniprodueto, donde el horizonte de planeación comprende 30 períodos, existiendo un total de 17 inventarios de producto en proceso con un total de 22 estaciones de trabajo, la materia prima puede ser surtida una vez cada 5 períodos, además de tener la restricción de que en todos los inventarios de producto en proceso no puede haber

A	b_{ij}	A	b_{ij}	A	b_{ij}
(1, 2)	$+\infty$	(6, 10)	20	(11, 14)	26
(1, 3)	$+\infty$	(7, 10)	52	(12, 14)	12
(1, 4)	$+\infty$	(7, 11)	12	(13, 15)	54
(1, 5)	$+\infty$	(8, 11)	28	(14, 15)	60
(2, 6)	34	(8, 12)	42	(15, 16)	100
(3, 7)	21	(9, 12)	28	(16, 17)	150
(4, 8)	70	(10, 13)	84		
(5, 9)	84	(11, 13)	90		

Tabla 5.1: Capacidades de producción del sistema generado

mas de 200 unidades esperando a ser procesadas.

Los resultados de las corridas de los modelos los presentamos en las siguientes Tablas. Los parámetros de costos que se consideraron en el sistema se muestran en la Tabla 5.3. Para los costos de mantener inventario se tomó en cuenta la capacidad máxima instalada del sistema, además de las capacidades de cada estación de trabajo, el vector de costos utilizado se muestra en la Tabla 5.4.

5.2. Resultados

Los resultados presentados en la Tabla 5.5 muestran que a medida que se permite mayor inventario de producto en proceso, es necesario invertir mayor tiempo para encontrar el óptimo, bajo las consideraciones realizadas. Incluir las capacidades de almacenamiento ayuda a reducir el tiempo necesario para resolver los problemas que

SPSU	:	Tipo de sistema
TXX	:	Valor del parámetro T
LXX	:	Número de inventarios
AXX	:	Numero de estaciones de trabajo
WXX	:	Valor del parámetro ω
X_MXXX	:	Inventario máximo

Tabla 5.2: Nomenclatura de los problemas generados

C	:	500
Q	:	100

Tabla 5.3: Parámetros de costos utilizados en la experimentación

se puedan plantear.

La restricción de transferir materia prima hacia producción impacta también en el tiempo de solución, pues al considerar esto se reduce el espacio de exploración donde se busca la solución.

Inventario	Costo (φ)
1	: 0
2	: 0.7
3	: 0
4	: 1.5
5	: 3
6	: 0
7	: 2.05
8	: 0
9	: 1.33
10	: 3
11	: 3
12	: 0
13	: 0.13
14	: 1.14
15	: 0.23
16	: 0.85
17	: 0

Tabla 5.4: Parámetros de costos por mantener inventario utilizados en la experimentación

Nombre	Unidades Incumplidas	Tiempo CPU	Valor objetivo
SPSU-T30-I17-A22-W2-X_M-200	0	38.58	\$17,099
SPSU-T30-I17-A22-W2-X_M-260	0	174.75	\$16,260
SPSU-T30-I17-A22-W2-X_M-320	0	186.48	\$16,260
SPSU-T30-I17-A22-W2-X_M-380	0	242.04	\$16,249
SPSU-T30-I17-A22-W2-X_M-440	0	276.93	\$16,219
SPSU-T30-I17-A22-W2-X_M-+INF	0	249.8	\$16,070
SPSU-T30-I17-A22-W3-X_M-200	134	7.2	\$83,981
SPSU-T30-I17-A22-W3-X_M-260	0	24.07	\$16,913
SPSU-T30-I17-A22-W3-X_M-320	0	51.04	\$16,761
SPSU-T30-I17-A22-W3-X_M-380	0	59.93	\$16,711
SPSU-T30-I17-A22-W3-X_M-440	0	77.19	\$16,692
SPSU-T30-I17-A22-W3-X_M-+INF	0	77.6	\$16,377
SPSU-T30-I17-A22-W4-X_M-200	263	2.83	\$148,515
SPSU-T30-I17-A22-W4-X_M-260	46	12.9	\$41,128
SPSU-T30-I17-A22-W4-X_M-320	0	29.03	\$17,530
SPSU-T30-I17-A22-W4-X_M-380	0	34.64	\$17,306
SPSU-T30-I17-A22-W4-X_M-440	0	44.94	\$17,216
SPSU-T30-I17-A22-W4-X_M-+INF	0	30.64	\$16,616

Tabla 5.5: Resultados de la experimentación

Nombre	Unidades Incumplidas	Tiempo CPU	Valor objetivo
SPSU-T30-I17-A22-W5-X_M-200	383	5.34	\$208,512
SPSU-T30-I17-A22-W5-X_M-260	160	10.84	\$98,315
SPSU-T30-I17-A22-W5-X_M-320	4	11.73	\$21,217
SPSU-T30-I17-A22-W5-X_M-380	0	20.75	\$18,149
SPSU-T30-I17-A22-W5-X_M-440	0	29.66	\$17,516
SPSU-T30-I17-A22-W5-X_M-+INF	0	13.06	\$16,901
SPSU-T30-I17-A22-W6-X_M-200	463	2.88	\$247,988
SPSU-T30-I17-A22-W6-X_M-260	200	5.49	\$117,380
SPSU-T30-I17-A22-W6-X_M-320	64	11.35	\$51,380
SPSU-T30-I17-A22-W6-X_M-380	0	13.99	\$18,495
SPSU-T30-I17-A22-W6-X_M-440	0	19.84	\$18,308
SPSU-T30-I17-A22-W6-X_M-+INF	0	12.56	\$17,236
SPSU-T30-I17-A22-W7-X_M-200	503	3.55	\$267,898
SPSU-T30-I17-A22-W7-X_M-260	240	6.21	\$137,781
SPSU-T30-I17-A22-W7-X_M-320	104	7.45	\$71,343
SPSU-T30-I17-A22-W7-X_M-380	12	10.57	\$24,761
SPSU-T30-I17-A22-W7-X_M-440	0	9.63	\$18,479
SPSU-T30-I17-A22-W7-X_M-+INF	0	8.58	\$17,578

Tabla 5.6: Resultados de la experimentación

Nombre	Unidades Incumplidas	Tiempo CPU	Valor objetivo
SPSU-T50-I17-A22-W2-X_M-440	0	14453.7	\$14,071
SPSU-T50-I17-A22-W2-X_M-+INF	0	14583.8	\$13,545
SPSU-T50-I17-A22-W3-X_M-200	706	43.13	\$368,500
SPSU-T50-I17-A22-W3-X_M-260	93	418.99	\$63,660
SPSU-T50-I17-A22-W3-X_M-320	0	14450.1	\$15,711
SPSU-T50-I17-A22-W3-X_M-380	0	14451.2	\$15,309
SPSU-T50-I17-A22-W5-X_M-200	1315	18.52	\$671,972
SPSU-T50-I17-A22-W5-X_M-260	783	25.28	\$406,898
SPSU-T50-I17-A22-W5-X_M-320	334	141.66	\$184,874
SPSU-T50-I17-A22-W5-X_M-380	20	678.33	\$30,568
SPSU-T50-I17-A22-W5-X_M-440	0	6842.5	\$18,212
SPSU-T50-I17-A22-W5-X_M-+INF	0	14459.5	\$15,884
SPSU-T50-I17-A22-W6-X_M-200	1483	14.31	\$753,834
SPSU-T50-I17-A22-W6-X_M-260	983	24	\$506,438
SPSU-T50-I17-A22-W6-X_M-320	523	30.03	\$279,040
SPSU-T50-I17-A22-W6-X_M-380	143	109.23	\$92,311
SPSU-T50-I17-A22-W6-X_M-440	1	525.02	\$19,901
SPSU-T50-I17-A22-W6-X_M-+INF	0	14456.4	\$16,869

Tabla 5.7: Resultados de la experimentación

Nombre	Unidades Incumplidas	Tiempo CPU	Valor objetivo
SPSU-T50-I17-A22-W8-X_M-200	1715	18.41	\$871,202
SPSU-T50-I17-A22-W8-X_M-260	1283	20.25	\$655,640
SPSU-T50-I17-A22-W8-X_M-320	843	28.57	\$438,220
SPSU-T50-I17-A22-W8-X_M-380	413	46.99	\$226,692
SPSU-T50-I17-A22-W8-X_M-440	128	60.73	\$86,549
SPSU-T50-I17-A22-W8-X_M-+INF	0	319.7	\$18,207
SPSU-T50-I17-A22-W10-X_M-200	1883	12.32	\$953,064
SPSU-T50-I17-A22-W10-X_M-260	1423	17.18	\$725,375
SPSU-T50-I17-A22-W10-X_M-320	1043	15.93	\$537,550
SPSU-T50-I17-A22-W10-X_M-380	653	27	\$347,012
SPSU-T50-I17-A22-W10-X_M-440	379	68.18	\$212,745
SPSU-T50-I17-A22-W10-X_M-+INF	0	205.65	\$19,563
SPSU-T50-I17-A22-W11-X_M-200	1915	8.41	\$970,817
SPSU-T50-I17-A22-W11-X_M-260	1543	12.61	\$785,000
SPSU-T50-I17-A22-W11-X_M-320	1163	22.81	\$597,175
SPSU-T50-I17-A22-W11-X_M-440	442	28.08	\$244,524
SPSU-T50-I17-A22-W11-X_M-+INF	0	131.2	\$20,595

Tabla 5.8: Resultados de la experimentación

Capítulo 6

Conclusiones y Recomendaciones

6.1. Conclusiones

En esta tesis hemos desarrollado un conjunto de modelos para la optimización de sistemas de producción bajo un enfoque de optimización de los inventarios de producto en proceso, además de esto, hemos incluido en la modelación consideraciones que se presentan comúnmente en sistemas productivos, teniendo siempre en mente la necesidad de entregar soluciones en corto tiempo.

Consideramos sistemas de producción con configuración lineales, además de configuraciones de red, soportando la producción simple así como producción de ensamble.

Los coeficientes de transferencia presentados en esta tesis son útiles para reflejar las condiciones actuales del sistema así como para simplificar el modelo y por lo tanto el tiempo de solución. Hemos modelado sistemas donde la premisa de que es muy caro dejar de producir por mala asignación de recursos, lo que resulta mandatorio cumplir con la demanda siempre y cuando sea posible.

6.2. Aportaciones

Dentro de las aportaciones es la creación de cuatro modelos matemáticos que contemplan situaciones del día a día que se presentan en las industrias manufactureras. nuestro enfoque a los inventarios y la estructura de la modelación permiten entregar soluciones en tiempo relativamente corto.

Resultados de este trabajo fueron publicados en:

- Pacheco Leal Samuel David, “Buffers Optimization in Assembly Lines”, volumen editado del Encuentro Internacional de Ciencias de la Computación 2005, Puebla, México. ISBN 968 863 859 5.
- Pacheco Leal Samuel David, “Buffers Optimization in Assembly Lines”, Memorias del Encuentro Internacional de Ciencias de la Computación 2005, Puebla, México.
- Pacheco Leal Samuel David, “Optimización de Inventarios en Líneas de Ensamble”, Memorias del 3er. Congreso internacional de Innovación y Desarrollo Tecnológico, Cuernavaca, Morelos, México.

También se realizaron varias ponencias en:

- Pacheco Leal Samuel David, “Optimización de Buffers en Líneas de Ensamble”, XV Escuela Nacional de Optimización y Análisis Numérico, Morelia, Michoacán, México. Abril 2005.
- Pacheco Leal Samuel David, “Optimización de Buffers en Líneas de Ensamble”, Seminario de Investigación del Programa de Posgrado de Sistemas, Facultad de

Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León, Mayo 2005.

- Pacheco Leal Samuel David, “Buffers Optimization in Assembly Lines”, Taller de Optimización y Meta-heurísticas, Encuentro Internacional de Ciencias de la Computación 2005, Puebla, México, Septiembre 2005.
- Pacheco Leal Samuel David, “Optimización de Inventarios en Líneas de Ensamble”, 3er. Congreso internacional de Innovación y Desarrollo Tecnológico, Cuernavaca, Morelos, México, Septiembre 2005.

6.3. Recomendaciones

La experimentación realizada muestra buenos resultados en cuanto a la calidad de las soluciones y en cuanto al tiempo necesario para entregar dichas soluciones. Sin embargo, es necesaria una exploración mas a fondo con el resto de los modelos desarrollados.

La investigación futura para este trabajo:

- Desarrollar restricciones que ayuden a reducir aun más el tiempo de solución, esto sería de gran ayuda para la optimización de horizontes largos de planeación
- Desarrollar la experimentación para sistemas multiproducto
- Explorar la posibilidad de adaptar la función objetivo y algunas de las restricciones en procedimientos heurísticos

Apéndice A

Corte y Ramificación (Branch and Bound)

La técnica de ramificación-cotas usa un principio básico “divide y conquista” para explorar el conjunto de soluciones enteras factibles, aunque en lugar de explorar todas y cada una de las soluciones factibles, la técnica usa cotas en los costos óptimos para explorar sobre algunas secciones del conjunto de soluciones factibles.

Sea F el conjunto de soluciones factibles al problema

$$\text{mín } \mathbf{c}'\mathbf{x}$$

$$\text{s.t } \mathbf{x} \in F$$

Podemos dividir al conjunto F en una colección finita de subconjuntos F_1, \dots, F_k , y resolver por separado cada uno de estos problemas

$$\text{mín } \mathbf{c}'\mathbf{x}$$

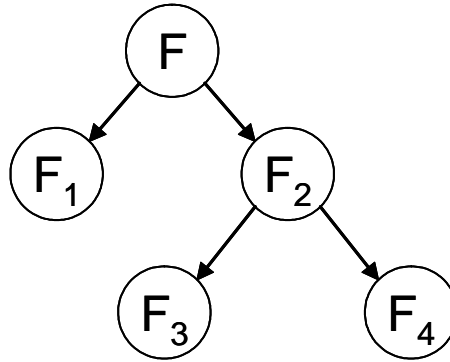


Figura A.1: Un árbol de subproblemas: el conjunto de soluciones factibles, ramificado en F_1 y F_2 ; además F_2 también ramificado en F_3 y F_4

$$\text{s.t } \mathbf{x} \in F_i, i = 1, \dots, k.$$

Después comparamos las soluciones óptimas de los subproblemas y escogemos la mejor de ellas. Cada subproblema puede ser casi tan difícil como el problema original eso nos sugiere intentar resolver cada subproblema por medio del mismo método, esto es seguir dividiendo el problema en subproblemas, este procedimiento es la parte de ramificación del método, creando así un árbol de subproblemas??

Se supone la existencia de un algoritmo eficiente en el que para cada F_i de interés, se calcula una cota inferior $b(F_i)$ del costo óptimo de subproblema correspondiente esto es ,

$$b(F_i) \leq \min_{\mathbf{x} \in F_i} \mathbf{c}'\mathbf{x}$$

La idea principal del método consiste en que mientras que el costo óptimo del subproblema es difícil de calcular óptimamente una cota del mismo puede resultar mas fácil de obtener. Una forma usual de obtener dicha cota es usando el costo óptimo de la relajación lineal del problema. Durante el transcurso del algoritmo ocasionalmente

se resuelven problemas a optimalidad o simplemente se evalúa el costo de ciertas soluciones factibles, esto nos permite mantener una cota superior U en el costo óptimo, la cual puede ser el costo de la mejor solución encontrada hasta el momento.

La esencia del método cae en la siguiente observación. Si la cota inferior $b(F_i)$ correspondiente a un subproblema en particular satisface $b(F_i) \geq U$, entonces este problema no debe ser considerado en las siguientes iteraciones dado que la solución del subproblema no supera la mejor solución factible encontrada hasta ese momento. En resumen el algoritmo a nivel general se describe a continuación, En cualquier punto el algoritmo guarda en memoria un conjunto de los subproblemas activos y el costo U de la mejor la mejor solución factible encontrada hasta ese momento. Inicialmente U toma el valor de ∞ o el costo de alguna solución factible si ésta se encuentra disponible, así la iteración típica del algoritmo es de la siguiente forma

1. Seleccionar un subproblema F_i
2. Si el subproblema es infactible, se elimina, de otra forma se calcula la cota $b(F_i)$ para el subproblema correspondiente.
3. Si $b(F_i) \geq U$ se elimina el subproblema
4. Si $b(F_i) < U$ se puede obtener una solución óptima al subproblema o quebrar este en sus correspondientes subproblemas los que se agregan a la lista de subproblemas activos.

En este algoritmo existen algunos parámetros libres en la mayoría de los casos las mejores opciones son dictadas por la experiencia y el tipo de problema que se trata de resolver. Ejemplos de estos parámetros son:

- (a) Existen diferentes maneras de seleccionar un subproblema de activo, dos de las mas usadas son breath first search y búsqueda del primero más profundo.
- (b) Hay varias formas de determinar la cota inferior $b(F_i)$ del costo óptimo del los subproblemas, la mas común es considerar la relajación lineal del subproblema.
- (c) Existen muchas formas de quebrar el problema en subproblemas.

Como ilustración, suponemos que usamos como cota inferior $b(F_i)$ el costo óptimo de la relajación lineal donde las restricciones de integralidad son ignoradas. Si se obtiene una solución óptima entera del problema relajado ésta automáticamente es una solución óptima para el correspondiente problema entero, eso evita la expansión en los subproblemas. El siguiente paso es actualizar U (si la solución obtenida es mejor que el valor previo de U) y así podemos eliminar el actual subproblema, si la solución \mathbf{x}^* al problema lineal relajado no es entera, entonces se escoge un componente x_i para el que x_i^* no es entero y se crean dos subproblemas agregando las siguientes restricciones

$$x_i \leq \lfloor x_i^* \rfloor, \quad \text{or} \quad x_i \geq \lceil x_i^* \rceil$$

Apéndice B

AMPL

B.1. Lenguaje en AMPL

AMPL es un lenguaje de modelado algebraico para programación matemática: un lenguaje capaz de expresar en notación algebraica problemas de optimización tales como los problemas de programación lineal. Veamos un pequeño ejemplo.

Ejemplo 1.1. Una compañía fabrica tres productos, P_1 , P_2 y P_3 , que precisan para su elaboración dos materias primas, M_1 y M_2 . Las disponibilidades semanales de estas materias son 25 y 30 unidades, respectivamente.

El beneficio neto que proporciona cada unidad de producto, así como las unidades de materia prima que necesita para su elaboración, vienen dados en la siguiente tabla:

	P_1	P_2	P_3
M_1	1	2	2
M_2	2	1	3
Beneficio (u.m.)	2	6	3

Planificar la producción semanal de forma que se maximice el beneficio.

Solución:

Sean x_1, x_2, x_3 (x_i) la cantidad producida de P_1, P_2 y P_3 respectivamente ($P_i, i = 1, 2, 3$).

El problema a resolver sería el siguiente:

$$\begin{aligned} \text{máx} \quad z &= 2x_1 + 6x_2 + 3x_3 \\ \text{s.a.} \quad x_1 + 2x_2 + 2x_3 &\leq 25 \\ 2x_1 + x_2 + 3x_3 &\leq 30 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

El modelo (modelo+datos) escrito en AMPL del ejemplo 1.1 es el siguiente:

```
# FABRICACION DE 3 PRODUCTOS CON 2 MATERIAS PRIMAS

# VARIABLES DE DECISION Y RESTRICCIONES DE NO NEGATIVIDAD

var x1 >= 0;

var x2 >= 0;

var x3 >= 0;

# FUNCION OBJETIVO DEL MODELO

maximize z : 2*x1 + 6*x2 + 3*x3;

# RESTRICCIONES DEL MODELO

subject to restriccion1 : x1 + 2*x2 + 2*x3 <= 25;

subject to restriccion2 : 2*x1 + x2 + 3*x3 <= 30;
```

Tabla B.1: Modelo básico del ejemplo

Es posible escribir un fichero de datos con valores diferentes y resolverlo junto al modelo general (por ejemplo con $n = 4$ y $m = 5$).

B.1.1. Reglas léxicas de AMPL

Los modelos AMPL envuelven **variables**, **restricciones** y **objetivos**, expresados con la ayuda de **conjuntos** y **parámetros**. A todos se les llama **elementos del modelo**. Cada elemento del modelo tiene un nombre alfanumérico (una cadena de uno o más letras, dígitos, y caracteres de subrayado): x_1 , z , restriccion1 . Nota: las letras mayúsculas son distintas de las letras minúsculas en AMPL.

Existen dos tipos de constantes:

- **Constantes numéricas:** un signo opcional, una secuencia de dígitos que pueden contener un punto decimal y un exponente opcional que comienza con una de las letras: d , D , e , E , como en $1.23D-45$. Toda la aritmética en AMPL tiene la misma precisión (sobre la mayoría de las máquinas tiene precisión doble).
- **Constantes literales** son cadenas delimitadas por una comilla simple o por dobles comillas. Si la comilla simple forma parte de la constante literal debe aparecer dos veces seguidas (igual ocurre con la doble comilla).

Cada línea de instrucciones debe ir finalizada con **un punto y coma** (;).

Los comentarios comienzan con $\#$ y se extienden hasta el final de la línea, o se pueden delimitar entre $/^*$ y $*/$, en cuyo caso pueden extenderse más de una línea.

```

# MODELO: EJEMPLO1.MOD

# FABRICACION DE n PRODUCTOS CON m MATERIAS PRIMAS

# PARÁMETROS DEL MODELO

param n >=0, integer;

param m >=0, integer;

# CONJUNTOS DE INDICES

set PRODUCTOS := 1..n;

set MPRIMAS := 1..m;

# VARIABLES DE DECISION Y RESTRICCIONES NO NEGATIVIDAD

var x {j in PRODUCTOS} >= 0;

# MAS PARAMETROS DEL MODELO

param c {i in PRODUCTOS};

param b {j in MPRIMAS};

param a {(i,j) in {MPRIMAS,PRODUCTOS}};

# FUNCION OBJETIVO DEL MODELO

maximize z : sum {j in PRODUCTOS} c[j]*x[j];

# RESTRICCIONES DEL MODELO

subject to restriccion {i in MPRIMAS} :

sum {j in PRODUCTOS} a[i,j]*x[j] <= b[i];

```

Tabla B.2: Modelo general del ejemplo

```
# DATOS: EJEMPLO1.DAT

param n := 3;

param m := 2;

param c:=

1 2

2 6

3 3;

param a : 1 2 3:=

1 1 2 2

2 2 1 3;

param b:=

1 25

2 30;
```

Tabla B.3: Datos para el ejemplo.

B.1.2. Los elementos de un conjunto

Un conjunto contiene cero o más elementos o miembros, cada uno de los cuales es una lista ordenada de una o más componentes. Cada miembro de un conjunto debe ser distinto. Todos los miembros deben tener el mismo número de componentes; a este número común se le conoce como **dimensión del conjunto**.

Un conjunto explícitamente se escribe como una lista de miembros separada por comas, colocados entre llaves: "{...}". Si el conjunto es de dimensión uno, los miembros son simplemente constantes numéricas o constantes de cadena, o cualquier expresión cuyo valor sea un número o una cadena:

$$\{\text{"a"}, \text{"b"}, \text{"c"}\}$$

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\{t, t+1, t+2\}$$

Para un conjunto multidimensional, cada miembro debe escribirse como una lista separada por comas entre paréntesis:

$$\{(\text{"a"}, 2), (\text{"a"}, 3), (\text{"b"}, 5)\}$$

$$\{(1, 2, 3), (1, 2, 4), (1, 2, 5), (1, 3, 7), (1, 4, 6)\}$$

El valor de un miembro numérico es el resultado de reordenar su representación decimal por un valor real de precisión limitada. Los miembros numéricos que parecen diferentes pero que al ser redondeados al valor de precisión limitada son el mismo, tales como 1 y 0.01E2, son considerados iguales.

B.1.3. Expresiones que indexan y subíndices

Muchos elementos de AMPL pueden definirse como colecciones indexadas sobre un conjunto; los miembros individuales son seleccionados al añadir un “**subíndice**” al nombrar el elemento. El rango de posibles subíndices es indicado por **una expresión que indexa** en la declaración del modelo. Operadores de reducción, tales como **sum**, también usan expresiones que indexan para especificar conjuntos sobre los que las operaciones son iteradas.

Un subíndice es una lista de expresiones simbólicas o numéricas, separadas por comas y encerradas entre corchetes, como en:

```
demanda[i]
```

```
costos[j,p[k],“0-”]
```

Cada expresión con subíndices debe evaluar a un número o a una literal. El valor resultante o secuencia de valores debe dar un miembro de un conjunto de índices unidimensional o multidimensional.

Una expresión que indexa es una lista de expresiones de conjunto separadas por comas, seguida opcionalmente por dos puntos “:” y una expresión lógica, todo encerrado entre llaves:

- Expresiones que indexan:

```
{lista de expresiones de conjuntos}
```

```
{lista de expresiones de conjuntos : expresión lógica}
```

- Lista de expresiones de conjuntos:

expresión de conjuntos

miembro-ciego in expresión de conjuntos

lista de expresiones de conjuntos, lista de expresiones de conjuntos

Cada expresión de conjunto puede ser precedida por un **miembro ciego** y la palabra clave **in**. Un miembro ciego para un conjunto unidimensional es un nombre no referenciado, es decir, un nombre no definido hasta ese punto. Un miembro ciego para un conjunto multidimensional es una lista separada por comas, encerrada entre paréntesis, de expresiones o nombres no referenciados; la lista debe incluir al menos un nombre no referenciado.

Un miembro ciego introduce uno o más índices ciegos (nombres no referenciados en sus componentes), cuyo **campo**, o rango de definición, comienza en la siguiente expresión de conjunto; el campo de un índice corre a través del resto de la expresión que indexa, hasta el final de la declaración usando la expresión que indexa, o hasta el final del operando que usa la expresión que indexa. Cuando un miembro ciego tiene una o más expresiones componentes, los índices ciegos del miembro ciego varían sobre una **proyección** del conjunto, es decir, ellos toman todos los valores para que el miembro ciego pertenezca al conjunto.

$\{A\}$	<i># todos los elementos de A</i>
$\{A,B\}$	<i># todos los pares, uno de A, uno de B</i>
$\{i \text{ in } A, j \text{ in } B\}$	<i># el mismo que antes</i>
$\{i \text{ in } A, C[i]\}$	<i># el mismo que antes</i>
$\{i \text{ in } A, (j,k) \text{ in } D\}$	<i># 1 de A y 1 (un par) de D</i>
$\{i \text{ in } A: p[i] > 30\}$	<i># todo i de A tal que p[i] sea mayor que 30</i>
$\{i \text{ in } A, j \text{ in } C[i]: i \leq j\}$	<i># nota: i y j deben ser numéricos</i>
$\{i \text{ in } A, (i,j) \text{ in } D: i \leq j\}$	<i># todos los pares con i en A y i,j en D</i> <i># (mismo valor de i) e i menor o igual que j</i>

El argumento opcional “**:** **expresión lógica**” es una expresión que indexa, selecciona solamente los miembros que verifican la expresión lógica y excluye a los demás. La expresión lógica típicamente envuelve uno o más índices ciegos de la expresión que indexa.

B.1.4. Expresiones aritméticas, lógicas y de conjuntos. Funciones matemáticas

En las expresiones lógicas y aritméticas de AMPL, pueden combinarse varios elementos. A partir de ahora consideraremos que una expresión que puede contener variables se representará como **vexpr**. Una expresión que no puede contener variables se denota como **expr** y algunas veces será llamada “expresión constante”, a pesar de que pueda contener índices ciegos. Una expresión lógica, representada como **lexpr**, puede contener variables sólo cuando es parte de una expresión **if** que produzca una

vexpr. Las expresiones de conjuntos serán denotadas como **sexpr**.

Los operadores aritméticos, lógicos y de conjuntos, en orden de precedencia creciente, se muestran en la tabla 4.

Los valores numéricos que aparecen como expresiones lógicas valen **falso** (false) si es 0, y **verdadero** (true) para cualquier otro valor numérico.

Expresiones Aritméticas

Las expresiones aritméticas son construidas con los operadores aritméticos usuales, con funciones de AMPL y con operadores de reducción aritméticos como **sum**:

```

número

variable

expresión  op.aritmético  expresión (+,-,less,*,/,mod,div,^,**)

unario    expresión

función( lista de expresiones )

if lexpr then vexpr [ else vexpr ]

operador-reducción  expresión que indexa (sum, prod, max, min)

{ expr }
```

Los operadores aritméticos, lógicos y de conjuntos, en orden de precedencia creciente, se muestran en la tabla .

Las funciones aritméticas incorporadas en AMPL se muestran en la tabla 5 y en la tabla 6.

Nombre	Tipo	Notas
if ... then ... else	A, S	A: si no hay else, se supone else 0 S: es obligatorio else sexpr
or,	L	o lógico
exists, forall	L	operadores de reducción lógica
and, &&	L	y lógico
<, <=, =, ==, <>, !=, >=, >	L	operadores relacionales
in, not in	L	pertenencia a un conjunto
within, not within	L	S within T significa $S \subseteq T$
not, !	L	negación lógica
union, diff, symdiff	S	symdiff es la diferencia simétrica
inter	S	intersección
cross	S	producto cartesiano
setof .. by	S	constructor de conjuntos
+, - , less	A	a less b = $\max(a - b, 0)$
sum, prod, min, max	A	operadores de reducción aritmética
*, /, div, mod	A	div cociente entero
+, -	A	más y menos unario
^, **	A	exponenciación

Tabla B.4: Operadores aritméticos (A), lógicos (L) y de conjuntos (S).

Sintaxis	Significado
Beta(a,b)	$x^{a-1}(1-x)^{b-1}/(\Gamma(a)\Gamma(b)/\Gamma(a+b)), x \in [0, 1]$
Cauchy	$1/(\pi(1+x^2))$
Exponential	$e^{-x}, x > 0$
Gamma(a)	$x^{a-1}e^{-x}/\Gamma(a), x \geq 0, a > 0$
Irland224()	Uniforme entera en $[0, 2^{24})$
Normal(μ, σ)	$N(\mu, \sigma)$
Poisson(μ)	$e^{-\mu}\mu^k/k!, k = 0, 1, \dots$
Uniform(m,n)	Uniforme $[m, n)$
Uniform01()	Uniforme $[0, 1)$

Tabla B.5: Funciones de generación de variables aleatorias en AMPL.

Sobre los operadores de reducción aritmética:

- La palabra clave **sum** debe seguirle cualquier expresión que indexa. La expresión aritmética siguiente se evalúa una vez para cada miembro del conjunto de índices, y todos los valores resultantes se suman. El operador **sum** tiene menor precedencia que *****, así podemos escribir:

$$\text{sum}\{i \text{ in ORIG}, j \text{ in DEST}, p \text{ in PROD}\}$$

$$\text{cost}[i,j,p] * \text{Trans}[i,j,p]$$

representa el total de $\text{costo}[i,j,p] * \text{Trans}[i,j,p]$ sobre todas las combinaciones de orígenes, destinos y productos.

- Otros operadores aritméticos iterados son **prod** para la multiplicación, **min** para el mínimo, y **max** para el máximo. Como ejemplo, podríamos usar:

Sintaxis	Significado
<code>abs(x)</code>	valor absoluto
<code>acos(x)</code>	$\cos^{-1}(x)$
<code>acosh(x)</code>	$\cosh^{-1}(x)$
<code>asin(x)</code>	$\sin^{-1}(x)$
<code>asinh(x)</code>	$\sinh^{-1}(x)$
<code>atan(x)</code>	$\tan^{-1}(x)$
<code>atan2(y,x)</code>	$\tan^{-1}(x/y)$
<code>atanh(x)</code>	$\tanh^{-1}(x)$
<code>ceil(x)</code>	entero mayor más cercano
<code>cos(x)</code>	coseno
<code>exp(x)</code>	exponencial
<code>floor(x)</code>	menor entero más cercano
<code>log(x)</code>	$\log_e(x)$
<code>log10(x)</code>	$\log_{10}(x)$
<code>max(x,y,...)</code>	máximo
<code>min(x,y,...)</code>	mínimo
<code>precision(x,n)</code>	x redondeado a n cifras significativas
<code>round(x,n)</code>	x redondeado a n dígitos después del punto decimal
<code>round(x)</code>	x redondeado al entero más cercano
<code>trunc(x,n)</code>	x truncado a n dígitos después del punto decimal
<code>trunc(x)</code>	x truncado a un entero

Tabla B.6: Algunas funciones aritméticas en AMPL.

```
max{i in ORIG} oferta[i,p]
```

para describir la mayor oferta del producto p de todos los orígenes.

Expresiones Lógicas

Las expresiones lógicas aparecen donde se requiera un valor: “verdadero” o “falso”. Por ejemplo, en el comando **check**, en la parte “tal que” de las expresiones que indexan (sigue a los “:”) y en **if lexp then ...**

Las expresiones lógicas pueden ser de la siguiente forma (lexpr):

```
expr
expr oper -- compara expr
lexpr oper -- logico lexp
not lexp
miembro in sexpr
miembro not in sexpr
sexpr within sexpr
sexpr not within sexpr
exists indexado lexp
forall indexado lexp
{ lexp }
```

El operador **exists**, cuando se aplica sobre un conjunto vacío, devuelve falso y el operador **forall** devuelve verdadero.

Expresiones de conjunto

Las expresiones de conjuntos (`sexpr`) que producen conjuntos pueden tener uno de los siguientes formatos:

```
{ [ miembro [ , miembro ... ] ] }
sexpr op-conjunto sexpr (union diff symdiff inter cross )
union indexado sexpr
inter indexado sexpr
expr .. expr [ by expr ]
setof indexado miembro
if lexpr then sexpr else sexpr
( sexpr )
interval
conj-predefinido
indexado
```

Podemos ver el uso del operador **setof**, en el siguiente ejemplo:

```
ampl: set y := setof {i in 1 ..5 } ( i,i ^2);
ampl: display y;
set y := (1,1) (2,4) (3,9) (4,16) (5,25);
```

B.1.5. Declaraciones de elementos del modelo

La declaración de los elementos del modelo tiene el siguiente formato general:

<pre>elemento nombre [alias] [exp. indexada] [cuerpo];</pre>
--

Las palabras claves para los elementos del modelo AMPL pueden ser una de las siguientes:

```

set

param

var

arc

minimize

maximize

subject to, subj to, s.t.

node

```

Si se suprime el nombre del elemento se supone que es **subject to**.

Las declaraciones pueden aparecer en cualquier orden, con la excepción de que cada nombre debe estar declarado antes de ser usado.

Para las declaraciones de variables, restricciones y objetivos, se permite una forma especial de expresión indexada:

```
{if lexpr}
```

Si la expresión lógica `lexpr` es verdad, entonces el resultado es un elemento simple (no indexado); en otro caso el elemento es excluido del modelo.

Declaración de conjuntos

La declaración de conjunto del modelo tiene el siguiente formato general:

```
set nombre [alias] [exp. indexada] [atributos] ;
```

en la que atributos es una lista de atributos opcionales separada por comas. Los cuales pueden ser (sexpre indica una expresión de conjuntos):

```

dimen n

within sexpre

:= sexpr

default sexpr

```

La frase `:=` especifica un valor para el conjunto; esto implica que el conjunto no será definido posteriormente en una línea de instrucciones específica para datos (`:=` y **default** son mutuamente excluyentes). El conjunto vacío se indica con: `{}`.

Existe la función **card(S)** la cual da el número de elementos del conjunto S.

También se pueden realizar operaciones entre conjuntos, como:

```

set A := 1 ..n ;

set B := i..j by k;

set C := A d i f f B;

set D := A union B;

set E := A inter B;

set F := A symdi f f B;

```

Se pueden definir conjuntos con infinitos elementos (Nota: no se puede iterar sobre conjuntos infinitos), los clásicos intervalos cerrados, abiertos o semicerrados, bien de números reales (**interval [a,b]**) o bien de números enteros (**integer [a,b]**). Nota: la palabra `interval` puede omitirse.

Declaración de parámetros

La declaración de un parámetro del modelo tiene el siguiente formato general:


```
param nombre [alias] [exp. indexada] [atributos] ;
```

en la que los atributos es una liste de atributos opcionalmente separa por comas.

Los cuales pueden ser (sexpr indica una expresión de conjuntos):

binary

integer

symbolic

oprel expr

in sexpr

:= expr

default t expr

donde “oprel” puede ser:

< <= = == != <> > >=

El atributo **in** especifica un chequeo para ver que el parámetro se encuentra en el conjunto dado. Los parámetros indexados pueden definirse de forma recursiva. Por ejemplo:

```
param comb 'n sobre k' {n in 0 .. N, k in 0 .. n}
```

```
:= if k=0 or k = n then 1 else comb[n-1,k-1] + comb[n-1,k];
```

Infinity es un parámetro predefinido; al igual que **-Infinity**.

Declaración de variables

La declaración de una variable del modelo tiene el siguiente formato general:

```
var nombre [alias] [exp. indexada] [atributos] ;
```

en la que atributos es una lista de atributos opcionalmente separada por comas.

Los cuales pueden ser (vexpr indica una expresión de variables):

```

binary

integer

>= expr

<= expr

:= expr

default t expr

= vexpr

coeff [ exp.indexada ] restricción vexpr

cover [ exp. indexada ] restricción

obj [ exp. indexada ] objetivo vexpr

```

Las frases `>=` y `=<` especifican cotas, la frase `:=` indica un valor inicial. La frase **default** indica los valores iniciales por defecto, cuyos valores pueden darse en una línea de instrucciones específica para datos.

Las frases **coeff** y **obj** se utilizan para la generación de coeficientes por columnas; estas especifican los coeficientes que serán colocados es la restricción indicada u objetivo indicado, el cual debe ser previamente referenciado usando **to_come**. Una frase **cover** es equivalente a la frase **coeff** pero con vexpr igual a 1.

Declaración de restricciones

La declaración de una restricción de modelo tiene el siguiente formato general:

```
[subject to] nombre [alias] [exp. indexada] [ := dual-inic]
[default dual-inic] [ : expr restriccion];
```

La frase opcional `:= dual-inicial` especifica un valor inicial para la variable dual (multiplicador de Lagrange) asociado con la restricción. La expresión de restricción debe estar en uno de los siguientes formatos:

```
vexpr <= vexpr
vexpr = vexpr
vexpr >= vexpr
expr <= vexpr <= expr
expr >= vexpr >= expr
```

Para permitir la generación de coeficientes por columna para la restricción, una de las `vexprs` puede tener una de las siguientes formas:

```
to come + vexpr
vexpr + to come
to come
```

Los términos de esta restricción que se especifican en una declaración `var` son colocados en la posición de `to _ come`.

Declaración de objetivos

La declaración de un objetivo del modelo tiene el siguiente formato general:

```
maximize nombre [alias] [exp. indexada] [: expresion] ;
minimize nombre [alias] [exp. indexada] [: expresion] ;
```

y puede especificarse una expresión en una de las siguiente formas:

vexpr

to come + vexpr

vexpr + to come

to come

La forma **to_come** permite la generación de coeficientes por columna, como con las restricciones

Notación para sufijos para valores auxiliares

Las variables, restricciones y objetivos tienen una variedad de valores auxiliares asociados, a los cuales se puede acceder añadiendo al nombre uno de los siguientes sufijos dependiendo del tipo de elemento del modelo.

Sufijos para variables

.init	valor actual inicial
.init0	valor inicial inicial (x_j^0)
.lb	cota inferior actual
.lb0	cota inferior inicial (l_j)
.rc	costo reducido menor ($.rc, x_j \geq l_j$)
.slack	menor holgura ($x_j - l_j$)
.rc	costo reducido $- (z_j - c_j)$
.slack	min (lsalck, uslack)
.ub	cota superior actual
.ub0	cota superior inicial (u_j)
.urc	costo reducido superior ($.rc, x_j \leq u_j$)
.uslack	holgura superior ($u_j - x_j$)
.val	valor actual (x_j)

Sufijos para restricciones

.body	valor actual del cuerpo de la restricción ($\mathbf{A}_i\mathbf{x}$)
.dinit	valor inicial actual para la variable dual
.dinit0	valor inicial inicial para la variable dual (w_i^0)
.dual	variable dual actual (w_i)
.lb	cota inferior (rl_i)
.ldual	valor dual menor ($.dual, \mathbf{A}_i\mathbf{x} \geq rl_i$)
.lsalck	holgura menor ($\mathbf{A}_i\mathbf{x} - rl_i$)
.slack	$\min(lsalck, uslack)$
.ub	cota superior (ru_i)
.udual	valor dual superior ($.dual, \mathbf{A}_i\mathbf{x} \leq ru_i$)
.uslack	holgura superior ($ru_i - \mathbf{A}_i\mathbf{x}$)

Sufijos para objetivos

.val valor actual

B.1.6. Especificación de datos

Hay que tener en cuenta que:

- La lectura de datos se inicializa con el comando "data". Por ejemplo:

```
ampl: data diet.dat;
```

lee los comandos de datos de un fichero llamada diet.dat.

- AMPL trata cualquier secuencia de espacios, tabuladores y caracteres claves de una nueva línea como un solo espacio.

- El final de cualquier comando de datos de indica por un punto y coma “;”.

Datos de un conjunto

Conjuntos unidimensionales.

Un conjunto simple se especifica al listar sus miembros.

```
set ORIG := SE MD BA ;
```

```
set DEST := CA CO HU AL JA MA GR ;
```

```
set PROD := plato cuchillo tenedor ;
```

Si un conjunto se ha declarado con el atributo **ordered** o **circular**, debemos de listar sus miembros en orden:

```
set SEMANAS := 27 sep93 04 oct93 11 oct93 18 oct93 ;
```

- Su una cadena de la lista incluye caracteres distintos de letras, dígitos, signo de subrayado (_), punto, + y -, debe ser cerrado entre comillas:

```
set ALMACEN := “A&P” JEWEL VONS ;
```

- También para distinguir cuando un número queremos que sea una cadena (“+1” o “3e4”), éste debe ser cerrado entre comillas.
- Los miembros de un conjunto deben ser todos diferentes; AMPL avisará de la existencia de elementos duplicados. Los números que tienen la misma representación en el ordenador serán considerados como iguales.
- Para una colección indexada de conjuntos, los miembros de cada conjunto de la colección se especificarán individualmente.

```

set PROD ;

set AREA { PROD };

set PROD := plato cuchillo ;

set AREA [ plato ] := este norte ;

set AREA [ cuchillo ] := este oeste export ;

```

- Podemos especificar explícitamente que uno o más de esos conjuntos es vacío, al dar una lista vacía; poniendo el punto y coma justo después del operador “:=”. Si queremos que AMPL suponga que cada conjunto es vacío excepto si se especifica otra cosa, incluyendo una frase **default** en el modelo:

```

set AREA { PROD } default {};

```

En otro caso seríamos avisados de que la especificación de los miembros de un conjunto no se ha realizado.

Conjuntos de dos dimensiones

Para un conjunto de pares, los miembros pueden especificarse de varias maneras:

```

set ORIG ;

set DEST ;

set LINKS within { ORIG , DEST };

```

1. lista de pares


```

set LINKS :=
    (SE,CO ) ( SE,HU ) ( SE,JA ) ( SE,GR ) ( MD,CA)
    (MD,CO ) ( MD,HU ) ( MD,AL ) ( MD,JA ) ( MD,GR)
    (BA,CA ) ( BA,AL ) ( BA,JA ) ( BA,MA ) ;

```

2. lista de pares, sin los paréntesis y las comas

```

set LINKS :=
    SE CO  SE HU  SE JA  SE GR  MD CA
    MD CO  MD HU  MD AL  MD JA  MD GR
    BA CA  BA AL  BA JA  BA MA  ;

```

3. un conjunto de pares puede especificarse en una tabla también de la siguiente forma:

```

set LINKS:  CA  CO  HU  AL  JA  MA  GR  :=
    SE      -  +  +  -  +  -  +
    MD      +  +  +  +  +  -  +
    BA      +  -  -  +  +  +  -  ;

```

Un signo “+” indica un par que está en el conjunto y un signo “-” indica que no está ese par. Normalmente las filas son etiquetadas con la primera componente y las columnas con la segunda. Si se prefiere lo puesto, podemos indicar una tabla transpuesta al añadir (**tr**) después del nombre del conjunto:

```

set LINKS (tr):
    SE MD BA :=
CA - + +
CO + + -
HU + + -
AL - + +
JA + + +
MA - - +
GR + + - ;

```

Las tablas son más convenientes para conjuntos que son relativamente densos. En otro caso la lista de pares va mejor.

4. Otra forma de describir un conjunto de pares es listar todas las segundas componentes que unen cada primer componente:

```

set LINKS:=
(SE,*) CO HU JA GR
(MD,*) CA CO HU AL JA GR
(BA,*) CA AL JA MA ;

```

Se podría hacer listando todas las primeras componentes que unen con cada una de las segundas componentes: (*,CA) MD BA.

Cada comodín * es seguido por una lista, cuyas entradas son sustituidas por el * para generar pares de conjuntos.

Conjuntos multidimensionales

Utilizamos los siguientes ejemplos para ver las distintas formas de definir conjuntos multidimensionales:

```
set RUTAS within {ORIG,DEST,PROD};
```

```
set RUTAS :=
```

```
(SE,HU, cuchillo ) ( SE,JA, cuchillo ) (SE,GR, cuchillo )
```

```
(MD,CA, plato ) ( MD,CA, cuchillo ) (MD,CO, plato )
```

```
... ;
```

```
set RUTAS :=
```

```
SE HU cuchillo SE JA cuchillo SE GR cuchillo
```

```
MD CA plato MD CA cuchillo MD CO plato
```

```
... ;
```

```
set RUTAS :=
```

```
(MD ,*, plato ) CA CO HU JA GR
```

```
(BA ,*, plato ) CA AL JA MA
```

```
... ;
```

```
set RUTAS :=
```

```
(*, CA ,*) MD plato MD cuchillo BA plato
```

```
(*, CO ,*) MD plato MD cuchillo
```

```
... ;
```

```

set RUTAS :=

(*,*, plato ):  CA  CO  HU  AL  JA  MA  GR :=

    SE          -   -   -   -   -   -   -
    MD          +   +   +   -   +   -   +
    BA          +   -   -   +   +   +   -

(*,*, cuchillo ):  CA  CO  HU  AL  JA  MA  GR :=

    SE          -   -   +   -   +   -   +
    MD          +   +   +   +   +   -   -
    BA          -   -   -   -   -   +   - ;

```

También se puede usar la notación (**tr**).

Datos de parámetros

Para un parámetro escalar (no indexado), la asignación de un valor sería:

```
param avail := 40;
```

Parámetros indexados

La forma más simple para dar datos para un parámetro indexado es por medio de una lista. Para un parámetro indexado sobre un conjunto simple:

```

set PROD;

param valor {PROD} > 0;

```

Cada elemento de la lista de datos consta de un miembro del conjunto y de un valor;

```
set PROD:= plato cuchillo tenedor ;
```

```

param valor :=
    plato 200
    cuchillo 140
    tenedor 160 ;

```

equivalentemente:

```

param valor := plato 200, cuchillo 140, tenedor 160 ;

```

A menudo necesitamos datos para varios parámetros que están indexados sobre el mismo conjunto, en esta situación puede escribirse:

```

param valor := plato 200 cuchillo 140 tenedor 160 ;
param benefi := plato 25 cuchillo 30 tenedor 29 ;
param market := plato 6000 cuchillo 4000 tenedor 3500 ;

```

o

```

param:   valor  benefi  market :=
plato    200    25      6000
cuchillo 140    30      4000
tenedor  160    29      3500 ;

```

Los dos puntos después de la palabra clave **param** es obligatoria; indica que damos valores para varios parámetros.

Si se sigue con el nombre del conjunto **PROD** y otros dos puntos:

```

param: PROD:   valor  benefi  market :=
    plato    200    25      6000
    cuchillo 140    30      4000
    tenedor  160    29      3500 ;

```

entonces los elementos del conjunto **PROD** son definidos también con este coman-

do, evitando así la definición con el comando `set PROD`; el efecto es combinar las especificaciones del conjunto y los tres parámetros indexados sobre él.

Parámetros bidimensionales

Los valores de un parámetro indexado sobre dos conjuntos, son generalmente introducidos como:

```

set ORIG;

set DEST;

param costo {ORIG,DEST} >= 0;

data

param costo:  CA  CO  HU  AL  JA  MA  GR :=

SE           39  14  11  14  16  82  8

MD           27  9   12  9   26  95  17

BA           24  14  17  13  28  99  20 ;

```

Las etiquetas en las filas dan el primer índice y las etiquetas de las columnas dan el segundo índice, así por ejemplo, `costo["SE", "CA"]` se ha definido a 39.

Podemos usar la notación (tr):

```

param costo ( tr ):

           SE  MD  BA

CA        39  27  24

CO        14  9   14

... ;

```

Cuando las tablas son grandes pueden utilizarse caracteres de nueva línea en cualquier lugar, o también emplear el siguiente formato:

```

param costo : CA CO HU AL :=
    SE      39 14 11 14
    MD      27 9 12 9
    BA      24 14 17 13

:      JA MA GR :=
    SE      16 82 8
    MD      26 95 17
    BA      28 99 20

```

Los dos puntos indica el comienzo de cada nueva subtabla; cada una tiene las misma etiquetas de filas, pero diferentes etiquetas de columna.

El parámetro no tiene porque estar indexado sobre todas las combinaciones de miembros de ORIG y DEST, sino tan sólo de un subconjunto de esos pares:

```

set LINKS within { ORIG , DEST };

param costo { LINKS } >= 0;

```

Como se vio en la sección anterior, el conjunto LINKS puede definirse como:

```

param costo: CA CO HU AL JA MA GR
    SE . 14 11 . 16 . 8
    MD 27 9 12 9 26 . 17
    BA 24 . . 13 28 99 . ;

```

Donde un “+” indica un miembro del conjunto, la tabla para costo da un valor.

Donde un “-” indica que no pertenece, la tabla contiene un punto “.”. El punto puede aparecer en cualquier tabla AMPL para indicar “valor no especificado aquí”.

Es posible definir un símbolo diferente, por ejemplo `--`, al incluir el siguiente comando en data:

```
defaultsym "--";
```

el cual puede ser desactivado al introducir el comando:

```
nodefaultsym ;
```

También es posible introducir los datos del siguiente modo:

```
param costo :=
    SE CO 14 SE HU 11 SE JA 16 SE GR 8
    MD CA 27 ... ;
```

Cuando un parámetro está indexado sobre un subconjunto poco denso de pares, una lista puede ser más compacta y legible que la representación tabular, la cual estaría formada mayoritariamente por puntos.

Otra ventaja del formato lista es que, como en el caso unidimensional, los datos para varias componentes pueden darse juntos:


```

param: LINKS: costo limit :=
      SE CO  14   1000
      SE HU  11   800
      SE JA  16  1200
      SE GR   8  1100
      MD CA  27  1200
      MD CO   9   600
      MD HU  12   900
      MD AL   9   950
      MD JA  26  1000
      MD GR  17   800
      BA CA  24  1500
      BA AL  13  1400
      BA JA  28  1500
      BA MA  99  1200 ;

```

Esta tabla da simultáneamente los miembros de `LINKS` y los valores para costo, y también los valores para otros parámetros, `limit`, que está también indexado sobre `LINKS`.

Finalmente, la lista de datos para costo puede escribirse más concisamente al organizarla en trozos, como se mencionó para los miembros del conjunto `LINKS` en la sección previa.

```

set LINKS :=
    (SE ,*) CO HU JA GR
    ... ;

param costo :=
    [SE ,*] CO 14 HU 11 JA 16 GR 8
    [MD ,*] CA 27 CO 9 HU 12 AL 9 JA 26 GR 17
    [BA ,*] CA 24 AL 13 JA 28 MA 99 ;

```

Parámetros multidimensionales.

Podríamos introducirlos de las siguientes formas:

```

set ORIG ;
set DEST ;
set PROD ;
set RUTAS within { ORIG,DEST,PROD };
param costo { RUTAS } >= 0;

```

lista simple

```

param costo :=
    MD CO plato 9 MD CO cuchillo 8 MD CA plato 27
    MD CA cuchillo 23 MD GR plato 17 ... ;

```

uso de trozos

```

param costo :=
    [MD, *, plato ] CA 27 CO 9 HU 12 JA 26 GR 17
    [BA, *, plato ] CA 24 AL 13 JA 28 MA 99
    ... ;

```

uso de trozos

```

param costo :=
    [*, CA, *] MD plato 27 MD cuchillo 23 BA plato 24
    [*, CO, *] MD plato 9 MD cuchillo 8
    ... ;

```

uso de trozos 2 dimensiones y tablas

```

param costo :=
    [*,*, plato ]:   CA  CO  HU  AL  JA  MA  GR :=
                    MD 27  9  12  .  26  .  17
                    BA 24  .  .  13  28  99  .
    [*,*, cuchillo ]: CA  CO  HU  AL  JA  MA  GR
                    SE .  .  11  .  16  .  8
                    MD 23  8  10  9  21  .  .
                    BA .  .  .  .  .  81  . ;

```

Se puede emplear la notación (tr).

Otro ejemplo:

```

set PROD ;

set AREA { PROD };

param T > 0;

param renta {p in PROD , AREA [p ] , 1 ..T } >= 0;

data

param T := 4;

set PROD := plato cuchillo ;

set AREA [ plato ] := este norte ;

set AREA [ cuchillo ] := este oeste export ;

param renta :=

[ plato, *,*]:    1    2    3    4 :=
este              25.0  26.0  27.0  27.0
norte             26.5  27.5  28.0  28.5

[ cuchillo, *,*]:  1    2    3    4 :=
este              30    35    37    39
oeste             29    32    33    35
export            25    25    25    28 ;

```

Valores por defecto.

AMPL comprueba que los comandos de datos asignan valores para exactamente todos los parámetros en el modelo. AMPL dará un mensaje de error si damos un valor para un parámetro inexistente, o nos olvidamos de dar un valor a un parámetro que

existe.

Si el mismo valor apareciera muchas veces en un comando de datos, podemos especificar la frase default. Por ejemplo,

```

set ORIG ;

set DEST ;

set PROD ;

param costo { ORIG,DEST,PROD } >= 0;

data

param costo default 9999 :=

[*,*, plato ]:   CA  CO  HU  AL  JA  MA  GR :=

                MD 27  9  12  .  26  .  17

                BA 24  .  .  13 28 99  .

[*,*, cuchillo ]: CA  CO  HU  AL  JA  MA  GR :=

                SE .  .  11  .  16  .  8

                MD 23  8  10  9  21  .  .

                BA .  .  .  .  .  81  . ;

```

Tanto a los parámetros missing (como `costo["SE", "CA", "plato"]`), como a los marcados como omitidos con el uso de un punto (como `costo["SE", "CA", "cuchillo"]`), se les asignará el valor 9999. En total, hay 24 con valor 9999.

La característica default es especialmente útil cuando queremos que todos los parámetros de una colección indexada tengan el mismo valor. Por ejemplo:

```

param oferta { ORIG } >= 0;

param demanda { DEST } >= 0;

data

param oferta default 1 ;

param demanda default 1 ;

```

También, como se explicó en secciones anteriores, una declaración de parámetro puede incluir una expresión default. Por ejemplo:

```

param costo { ORIG,DEST,PROD } >= 0, default 9999;

```

Sin embargo, es mejor poner la frase default en los comandos de datos. La frase default debería ir en el modelo cuando queremos que el valor por defecto dependa de alguna forma de otros datos. Por ejemplo, un costo arbitrariamente grande podría darse para cada producto al especificar:

```

param gran_costo { PROD } > 0;

param costo { ORIG,DEST,p in PROD } >= 0, default gran_costo [p];

```

Datos para variables y restricciones

Opcionalmente podemos asignar valores iniciales a las variables o restricciones del modelo, usando cualquiera de las opciones para asignar valores a parámetros. El nombre de la variable almacena su valor, y el nombre de la restricción el valor de la variable dual asociada.

```

var Tans:  CA  CO  HU  AL  JA  MA  GR :=
SE      100 100 800 100 100 500 200
MD      900 100 100 500 500 200 200
BA      100 900 100 500 100 900 200

```

También con una tabla simple podemos dar valores a parámetros (valor, benefi, market) y variables (Make):

```

param:      valor  benefi  market  Make :=
    plato   200    25      6000    3000
    cuchillo 140    30      4000    2500
    tenedor  160    29      3500    1500 ;

```

Los valores iniciales de las variables (o expresiones que envuelven esos valores iniciales) pueden verse antes de escribir `solve`, usando los comandos **display**, **print** o **printf**.

El uso más común de asignar valores iniciales a variables o restricciones es dar un punto de arranque para resolver un problema de optimización no lineal.

B.1.7. Comandos del lenguaje

La llamada a AMPL normalmente causa la entrada en un entorno de comandos, donde los comandos pueden ser introducidos interactivamente. Las declaraciones del modelo y las instrucciones de introducción de datos son también aceptadas como comandos.

El entorno de comandos de AMPL reconoce dos modos. En modo modelo, reconoce las declaraciones del modelo y todos los comandos que se describirán a continuación. El otro modo es el modo datos, en el cual sólo se reconocen instrucciones referentes a la introducción de datos. El entorno siempre vuelve al modo modelo al encontrar una palabra clave que no comience con la palabra **data**.

Una frase de la forma:

```
include fichero ;
```

trae el contenido del fichero al entorno de comandos de AMPL. Los comandos **include** pueden estar anidados, ellos son reconocidos en modo modelo y en modo datos.

Las secuencias:

```
model; include fichero ;
```

```
data; include fichero ;
```

pueden abreviarse como:

```
model fichero ;
```

```
data fichero ;
```

Los comandos no son parte de un modelo, pero producen que AMPL actúe como se describe en la **tabla 7**. Los comandos distintos a **data**, **end**, **include**, **quit** y **shell** producen que AMPL entre en modo modelo.

Desde la línea de comandos de AMPL podemos escribir, por ejemplo:

```
ampl: include ejemplo1.run ;
```

siendo el fichero ejemplo1.run (tabla 8), un fichero por lotes que almacena la secuencia de comandos necesarios para resolver el ejemplo 1.1.

Los comandos **display**, **print** y **printf** imprimen expresiones arbitrarias. Tienen el siguiente formato:

```
display [conjunto: ] lista-argumentos [redireccion];
```

```
print [conjunto: ] lista-argumentos [redireccion];
```

```
printf [conjunto: ] fmt, lista-argumentos [redireccion];
```



```

# EJEMPL01.RUN

option solver cplex;

model ejemplo1.mod ;

data ejemplo1.dat ;

solve ;

display z;

display x;

display restriccion.slack ;

```

Tabla B.7: Fichero de lotes para el modelo del ejemplo 1.1

Si el conjunto está presente, su campo de acción se extiende hasta el final del comando, y causa una ejecución del comando para cada miembro del conjunto.

La lista-argumentos es una lista de expresiones separadas por comas.

El opcional **redirección** tiene una de las dos formas siguientes:

> fichero

>> fichero

La primera abre por primera vez un fichero para escribir, y la segunda añade al fichero ya creado, aunque > actúa igual que >>, si el fichero está ya abierto.

Con el comando **option** se puede conseguir que la salida que se ha solicitado tenga un formato

específico. Por ejemplo:

```
option display_precision 3;
```

```
option omit_zero_rows 1;
```

La primera especifica la precisión de salida (0 equivale a ninguna) y la segunda

Comandos	Significado
break	termina un bucle for o while
close	cierra un fichero
continue	salta al final del cuerpo del bucle
data	cambia a modo datos; opcionalmente incluye un fichero
display	imprime elementos del modelo y expresiones
delete	elimina un componente previamente declarado
drop	elimina una restricción u objetivo
end	finaliza la entrada del fichero de entrada actual
expand	muestra explícitamente el modelo
fix	fija una variable a su valor actual
for { indx } { cp }	bucle for
if lexpr then { }	comprueba una condición
include	incluye ficheros
let	cambia los valores de los datos (:=)
match(cad,mod)	posición de mod en cad
model	cambia al modo modelo; opcionalmente incluye un fichero
objective	selecciona un objetivo a optimizar
option	define o muestra valores opcionales
print	imprime elementos del modelo sin formatear
write	escribe en un fichero partes de un problema
xref	muestra dependencias del modelo

Tabla B.8: Algunos comandos del entorno AMPL.

omite las salidas con valor cero (por defecto es 0, es decir no omite los valores cero).

Otras opciones son:

```
option solver_msg 0;

option relax_integrality 1;

option presolve 0;

option show_stats 1;

option times 1;

option gentimes 1;

option log_file ' hola.tmp ';

option solution_round 6;

option single_step 1;
```

B.1.8. Optimizadores

AMPL clasifica los optimizadores en diferentes tipos atendiendo al tipo de problema que resuelven y los métodos que usan, estos son:

- **Lineal (símplex)**: objetivo y restricciones lineales, resueltos con alguna versión del método del símplex.
- **Lineal (interior)**: objetivo y restricciones lineales, resueltos con alguna versión de métodos de punto interior (o barrera).
- **Redes**: objetivo lineal y restricciones de flujo en redes, resuelto por alguna versión del método símplex para redes.
- **Cuadrático**: objetivos cuadráticos convexos o cóncavos y restricciones lineales,

resuelto por un método tipo símplex o método tipo punto interior.

- **Convexo:** objetivo convexo o cóncavo pero no todos lineales, y restricciones lineales, resuelto por un método de punto interior.
- **No lineal:** continuo pero no todos los objetivos y restricciones lineales, resuelto por cualquiera de los métodos siguientes: gradiente reducido, cuasi-newton, lagrangiano aumentado y punto interior.
- **Complementariedad:** Lineal o no lineal como antes, pero con condiciones de complementariedad adicional.
- **Lineal entero:** objetivo y restricciones lineales y alguna o todas las variables enteras, resuelto por alguna aproximación de ramificación y acotación que aplica un algoritmo lineal para resolver los subproblemas sucesivos.
- **Entero no lineal:** objetivos y restricciones continuos pero no todos lineales y alguna o todas las variables enteras, resuelto por la aproximación de ramificación y acotación que aplica un algoritmo no lineal para resolver los subproblemas sucesivos.

Bibliografia

- [1] F. Robert, *Software Survey: Linear Programming*. INFORMS OR/MS Today, 32(3), 2005
- [2] M. Pinedo, *Scheduling Theory, Algorithms and Systems*, New Jersey, Prentice Hall, 1995.
- [3] B. Dimitris, Tsitsiklis, John N. *Introduction to Linear Optimization*. Athena Scientific, 1997
- [4] Powell S., Pyke D., *Buffering unbalanced assembly systems*, IIE Transactions, 1998.
- [5] Armin Scholl, *Balancing and Sequencing of Assembly Lines*, Second Edition.
- [6] K. B. Baker, *Introduction to Sequencing and Scheduling*, New York, Wiley, 1974.
- [7] X. Cao and Y. Ho, *Models of Discrete Event Dynamic Systems*, IEEE Control Systems Magazine, Vol. 10, No. 4, pp. 69-76, 1990.
- [8] S. R. Jernigan and S. Ramaswamy and K. S. Barber, *On-line Scheduling Using a Distributed Simulation Technique for Intelligent Manufacturing Systems*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century, Vancouver, Canada, Vol. 3, pp. 2159-2164, 1995.
- [9] H. Cho and J. S. Smith and R. A. Wysk, *An Intelligent Workstation Controller for Integrated Planning and Scheduling of an FMS Cell*, Production Planning and Control, Vol. 8 No. 6, pp. 597-607, 1997.
- [10] P. Gu, A. Naumann, *Real-time Scheduling and Control of Manufacturing Cells*, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century, Vancouver, Canada, Vol. 3, pp. 2148-2152, 1995.
- [11] Y. X. Dai, B. S. Chen, *An Alternate Approach for Real-time Scheduling*, Proceedings of the IEEE International Conference on industrial Technology, Shanghai, China, pp. 189-193, 1996.
- [12] K-H Kim, P. J. Egbelu, *A mathematical Model for Job Shop Scheduling With Multiple Process Plan Consideration Per Job*, Production Planning and Control Magazine, Vol. 9, No. 3, pp. 250-259, 1998.
- [13] P.B. Luh, D. J. Hootomt, *Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique*, ZEEE Transactions on Automatic Control, Vol. 38, pp. 1066-1079, 1993.

- [14] Bazaraa, M. S., J. J. Jarvis y H. D. Sherali, *Programación Lineal y Flujo en Redes*, 2da. edición, Limusa, 1998.
- [15] Benjaafar, S., *Machine Sharing in Cellular Manufacturing Systems, Planning, Design, and Analysis of Cellular Manufacturing Systems*, A. K. Kamrani, H. R. Parasei and D. H. Liles (Editors), Elsevier Science B. V., 1995
- [16] Bertsekas, D. P., *Network Optimization: Continuous and Discrete Models*, Athena Scienti. . . c, Belmont, Massachusetts, 1998.
- [17] Drolet, J. R., *Scheduling Virtual Cellular Manufacturing Systems*, Ph.D. Thesis, School of Industrial Engineering, Purdue University, West Lafayette, Indiana, 1989.
- [18] Flynn, B. B., y F. R. Jacobs, *A Simulation Comparison of Group Technology with Traditional Job Shop Manufacturing*, International Journal of Production Research, 24(5), 1171-1192, 1986.
- [19] Fu, M. C. y B. K. Kaku, *Minimizing Work-in-Process and Material Handling in the Facilities Layout Problem*, IIE Transactions, 29(1), 29-36, 1997.
- [20] Heragu, S. S., *Group Technology and Cellular Manufacturing*, IEEE Transactions on Systems, Man and Cybernetics, 24(2), 203-215, 1994.
- [21] Irani, S.A., *Production Flow Analysis in a Semiconductor Fab*, Proceedings of the Joint National Science Foundation and Semiconductor Research Corporation Workshop on Operational Methods in Semiconductor Manufacturing, University of California at Berkeley, CA, February 19-20, 1997.
- [22] Martin, R. K., *Large Scale Linear and Integer Optimization: A United Approach*, Kluwer Academic Publishers, 1999.
- [23] Sethi, A. K. y S. P. Sethi, 1990, *Flexibility in Manufacturing: A survey*, The International Journal of Flexible Manufacturing Systems, 2(4), 289-328, 1990.
- [24] Suresh, S. C., *Partitioning Work Centers for Group Technology: Analytical Extension and Shop-Level Simulation Investigation*, Decision Sciences, 23, 267-290, 1992.
- [25] Wemmerlöv, U. y L. N. Hyer, *Cellular Manufacturing in the U.S. Industry: A Survey of Users*, International Journal of Production Research, 27(9), 1511-1530, 1989.
- [26] Wolsey, L. A., *Integer Programming*, Wiley-Interscience, 1998.

Ficha autobiográfica

Samuel David Pacheco Leal

Candidato para el Grado de Maestro en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

Optimización de Inventarios en Sistemas de Producción

Nací en Monterrey, N.L. Hijo del Sr. Víctor Pacheco y la Sra. Hilda Leal de Pacheco.

Me gradué como Ingeniero Industrial Administrador en la Facultad de Ciencias

Químicas de la Universidad Autónoma de Nuevo León. Laboré en una empresa

manufacturera como Ing. Ind. y de Procesos y fungí como docente en una escuela

privada. En el año 2004 ingresé al Programa de Maestría en Ciencias en Ingeniería

de Sistemas apoyado por CONACYT. Actualmente laboro en una consultoría.