

MODELAMIENTO PARA PROGRAMAR PRODUCCIÓN A CORTO PLAZO EN
PLANTAS CON LOTES MULTIPROPÓSITO, BASADA EN TIEMPOS
CONTINUOS; DESARROLLADO EN ITALCOL S.C.A. BUCARAMANGA

JENNY ANGÉLICA ANGULO SOLEDAD
RONALD FERNANDO PINEDA BONILLA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA
2007

MODELAMIENTO PARA PROGRAMAR PRODUCCIÓN A CORTO PLAZO EN
PLANTAS CON LOTES MULTIPROPÓSITO, BASADA EN TIEMPOS
CONTINUOS; DESARROLLADO EN ITALCOL S.C.A. BUCARAMANGA

JENNY ANGÉLICA ANGULO SOLEDAD
RONALD FERNANDO PINEDA BONILLA

Trabajo de grado para optar al título de
Ingeniero Industrial

Directora:

Alina Fedossova
P. H. D. Ciencias Físico-Matemática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA
2007

*A nuestros padres, familias,
maestros y colegas,
que nos apoyaron
y comparten nuestro triunfo.*

Jenny y Ronald.

AGRADECIMIENTOS

A *Pedro José Salazar García*, Gerente de Producción; *Carolina Güiza Ojeda*, Asistente de Producción y todo el personal operativo del área productiva de ITALCOL S.C.A Bucaramanga, por su interés y colaboración en la realización de este proyecto.

A la profesora Alina Fedosova, por su asistencia en la dirección.

A la profesora Myriam Leonor Niño, por sus asesorías.

Y a todos aquellos que de una u otra forma cooperaron en la construcción de este proyecto de grado.

CONTENIDO

	pág.
<u>INTRODUCCIÓN</u>	<u>16</u>
<u>PARTE I</u>	<u>17</u>
<u>1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN</u>	<u>18</u>
<u>2. OBJETIVOS</u>	<u>19</u>
2.1 OBJETIVO GENERAL	19
2.2 OBJETIVOS ESPECÍFICOS	19
<u>3. ALCANCE Y LIMITACIONES DEL PROYECTO</u>	<u>20</u>
3.1 ALCANCE	20
3.2 LIMITACIONES	20
<u>4. MARCO TEÓRICO</u>	<u>21</u>
4.1 FUNDAMENTACIÓN TEÓRICA GENERAL	21
4.1.1 Sistemas de producción	21
4.1.2 Programación de operaciones	24
4.1.3 Scheduling	26
4.1.4 Secuenciación	28
4.1.5 Investigación operativa y optimización	29
4.1.6 Modelamiento matemático	30
4.1.7 Programación lineal entera mixta	31

4.2	FUNDAMENTACIÓN EN LITERATURA CONCERNIENTE	34
4.2.1	Modelamiento matemático a través de la PLEM para la producción continua, a corto plazo y de lotes multipropósito	34
4.2.2	Características generales de las formulaciones estudiadas	37
4.2.3	A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. Sundaramoorthy y Karimi	43
4.3	FUNDAMENTACIÓN TEÓRICA EN MÉTODOS DE SOLUCIÓN.	60
4.3.1	Lenguajes algebraicos de modelado	61
<u>5.</u>	<u>AMBIENTE DE DESARROLLO DEL PROYECTO</u>	<u>67</u>
5.1	ITALCOL S.C.A.	67
5.2	ORGANIZACIÓN	67
5.2.1	Misión	67
5.2.2	Objetivo	68
5.2.3	Política de calidad	68
<u>PARTE II</u>		<u>69</u>
<u>6.</u>	<u>METODOLOGÍA</u>	<u>70</u>
6.1	ETAPA 1: OBSERVACIÓN Y CARACTERIZACIÓN DEL PROBLEMA	71
6.1.1	Diagnóstico actual del sistema	72
6.1.2	Caracterización del sistema productivo	72
6.2	ETAPA 2: CONSTRUCCIÓN DEL MODELO MATEMÁTICO	73
6.2.1	Conjuntos	75
6.2.2	Parámetros	75

6.2.3 Variables y ecuaciones	75
6.3 ETAPA 3: SOLUCIÓN Y VALIDACIÓN DEL MODELO	76
6.3.1 Solución	77
6.3.2 Validación	78
<u>PARTE III</u>	<u>80</u>
<u>7. DESARROLLO DE LA METODOLOGÍA PROPUESTA</u>	<u>81</u>
7.1 ETAPA 1: OBSERVACIÓN Y CARACTERIZACIÓN DEL SISTEMA PRODUCTIVO	81
7.1.1 Diagnóstico actual del sistema	81
7.1.2 Caracterización del sistema productivo	87
7.2 ETAPA 2: CONSTRUCCIÓN DEL MODELO MATEMÁTICO	102
7.3 ETAPA 3: SOLUCIÓN Y VALIDACIÓN DEL MODELO	112
7.3.1 Validación	112
7.3.2 Solución	117
<u>8. CONCLUSIONES</u>	<u>123</u>
<u>9. RECOMENDACIONES</u>	<u>124</u>
<u>BIBLIOGRAFÍA</u>	<u>125</u>

LISTA DE FIGURAS

	pág.
Figura 1. Sistema de producción	22
Figura 2. Representación de slots sincrónicos	35
Figura 3. Representación de puntos eventos	36
Figura 4. Ejemplo de representación STN	38
Figura 5. Ejemplo de representación RTN	38
Figura 6. Ejemplo de representación de tarea	40
Figura 7. Ejemplo de representación RD	41
Figura 8. Representación RD	45
Figura 9. Representación de slots	46
Figura 10. Diagrama de receta para el ejemplo 1a	54
Figura 11. Diagrama de receta para el ejemplo 3a	55
Figura 12. Programación de máximo beneficio para el ejemplo 1a tomando $H=8$, $K=4$ y tiempos variable de procesamiento de lote.	57
Figura 13. Programación de máximo beneficio para el ejemplo 3a tomando $H=8$, $K=6$ y tiempos de procesamiento variable.	58
Figura 14. Programación de mínimo makespan para el ejemplo 1a con $K=12$	59
Figura 15. Ilustración del proceso branch and cut en términos de un árbol	66
Figura 16. Etapas metodológicas del proyecto	70
Figura 17. Fases de la observación y caracterización del problema	71
Figura 18. Fases de la formulación y construcción del modelo matemático.	74
Figura 19. Esquema general del proceso de producción de ITALCOL S.C.A. Bucaramanga	86
Figura 20. Diagrama de operaciones y secuencia de producción de la línea de harinas	90
Figura 21. Diagrama de operaciones y secuencia de producción de la línea de peletizados y quebrantados	91
Figura 22. Diagrama de operaciones y secuencia de producción de la línea de extrusados	92
Figura 23. Diagrama de operaciones para la tarea mezclar	94
Figura 24. Diagrama de operaciones para las tareas peletizar y quebrantar	96
Figura 25. Diagrama de operaciones	98
Figura 26. Diagrama de operaciones	97
Figura 27. Interfase gráfica para el modelo ITALCOL 2.2	106
Figura 28. Diagrama RD ITALCOL 1.1	107
Figura 29. Diagrama RD ITALCOL 2.1	108
Figura 30. Duración del slot k	118
Figura 31. Cantidad de material (tamaño de lote) involucrado y descargado en cada tarea i en el slot k	119

Figura 32. Cantidad de material (tamaño de lote) involucrado y descargado en cada tarea i en el slot k	120
Figura 33. Cantidad de material (tamaño de lote) involucrado y descargado en cada tarea i en el slot k	121
Figura 34. Cantidad de los productos demandados en último slot K	122

LISTA DE CUADROS

	pág.
Cuadro 1. Comparativo de similitudes y diferencias en las formulaciones estudiadas	42
Cuadro 2. Límites sobre los tamaños de lote de tareas y coeficientes en las expresiones para tiempos de procesamiento.	55
Cuadro 3. Capacidades de almacenamiento, inventarios iniciales y precios de venta de materiales.	56
Cuadro 4. Soluciones estadísticas del ejemplo 3 para maximizar beneficios y utilizando tiempos de procesamiento de lotes variables.	59
Cuadro 5. Soluciones estadísticas del ejemplo 1a y 3b para minimizar makespan , utilizando tiempos de procesamiento de lotes variables.	60
Cuadro 6. Familias de materias primas.	88
Cuadro 7. Tolvas de almacenamiento de producto en proceso y sus capacidades	89
Cuadro 8. Tiempo de mezcla	93
Cuadro 9. Límites capacidad de unidades	97
Cuadro 10. Productos tipo para el modelo	99
Cuadro 11. Tiempos de procesamiento en cada unidad para cada tarea	101
Cuadro 12. Parámetros de tareas y unidades	109
Cuadro 13. Parámetros de los materiales	111
Cuadro 14. Proceso real para el caso 1	112
Cuadro 15. Solución del modelo para maximización de beneficio, caso 1	113
Cuadro 16. Solución del modelo para minimización de makespan , caso 1	113
Cuadro 17. Proceso real para el caso 2	114
Cuadro 18. Solución del modelo para maximización de beneficio del caso 2	114
Cuadro 19. Solución del modelo para minimización de makespan del caso 2	115
Cuadro 20. Proceso real para el caso 3	116
Cuadro 21. Solución del modelo para maximización de beneficio del caso 3	116
Cuadro 22. Solución del modelo para minimización de makespan del caso 3	117

LISTA DE ANEXOS

- Anexo A. Hipervínculos de acceso electrónico a archivos soporte
- Anexo B. Ejemplo 3a para maximización de beneficio
- Anexo C. Ejemplo 3b para minimización de **makespan**
- Anexo D. Plano general ITALCOL S.C.A. Bucaramanga
- Anexo E. Diagrama general de rutas ITALCOL S.C.A. Bucaramanga
- Anexo F. Diagrama general de líquidos ITALCOL S.C.A. Bucaramanga
- Anexo G. Código fuente ITALCOL 1.1
- Anexo H. Código fuente ITALCOL 2.1
- Anexo I. Código fuente ITALCOL 2.2

GLOSARIO

BUFFER TIME: intervalo de tiempo en el que la tarea produce o consume una cierta cantidad de material desde y para un estado.

DECOPLAMIENTO: separación de la información de tareas y unidades, mediante dos variables binarias de asignación *w-variables*; *y-variables*.

ESTADO: fase en la que el material se encuentra almacenado de manera real o imaginaria; ubicado antes, después o entre dos tareas.

GAP: el criterio **gap**, es un criterio de terminación que se utiliza mediante comparación de soluciones (la mejor hasta el momento y la de un nodo sucesor), el cual establece el momento de finalización de la compilación.

HORIZONTE DE PROGRAMACIÓN: cantidad de tiempo para la cual se determina la programación de producción en una organización.

MAKESPAN: tiempo total de fabricación en el que todos los trabajos completan su ejecución.

NONZEROS: elementos diferentes a cero.

PUNTO EVENTO: puntos de tiempo desconocidos llamados eventos, que representan el comienzo de una tarea en una unidad.

RTN: "Resource Task Network", representación gráfica que describe el flujo de la planta, en la que se consideran todos los procesos como gráficos bipartidos comprimiendo dos tipos de nodos, recursos y tareas.

RD: "Recipe Diagram", representación gráfica que mediante nodos y arcos describen el flujo de planta, donde los nodos son las tareas, los arcos los materiales (material-estado) y la dirección del arco describe la tarea precedente

SLOT: representación del tiempo en términos de bloques ordenados de longitud variables desconocidas.

STN: "State Task Network" representación gráfica que describe el flujo de planta, representando todos los materiales como estados procesados a través de un conjunto de etapas de procesamiento.

SCHEDULING: asignación de recursos sobre el tiempo para realizar un conjunto de tareas.

TITULO: MODELAMIENTO PARA PROGRAMAR PRODUCCIÓN A CORTO PLAZO EN PLANTAS CON LOTES MULTIPROPÓSITO, BASADA EN TIEMPOS CONTINUOS; DESARROLLADO EN ITALCOL S.C.A. BUCARAMANGA*

AUTORES: JENNY ANGÉLICA ANGULO SOLEDAD
RONALD FERNANDO PINEDA BONILLA**

PALABRAS CLAVES: PROGRAMACIÓN DE OPERACIONES, PROGRAMACIÓN LINEAL ENTERA MIXTA, FORMULACIÓN DE TIEMPOS CONTINUOS, LOTES MULTIPROPÓSITO, SLOT.

DESCRIPCIÓN:

Este trabajo comprende la construcción, desarrollo, solución y validación de un modelo matemático que proporciona soluciones factibles y brinda a ITALCOL S.C.A. Bucaramanga, una base cuantitativa sólida para programar su producción a corto plazo.

El estudio del sistema de producción de la organización permite conocer las variables y relaciones matemáticas necesarias para representar lo más real posible su proceso productivo y cursos de acción y en base a un modelo matemático previamente formulado de programación lineal entera mixta, tiempos continuos y lotes multipropósito, se construye un lenguaje propio de programación en el sistema general de modelamiento algebraico (GAMS), se compila y mediante su motor de solución CPLEX, se obtiene: la secuencia de tareas, la cantidad de material a procesar en determinado momento y el tiempo de procesamiento de cada tarea, que tienen lugar en cada puesto de trabajo.

En este sistema de producción, el modelo se diseña y resuelve para dos objetivos de programación, maximizar beneficio y/o minimizar makespan, encontrando de manera rápida soluciones válidas, factibles y favorables para la organización, que apoyan el proceso de toma de decisión al programar las operaciones y desligan la intervención subjetiva del encargado. Mediante análisis y comparación entre registros históricos de producción y soluciones halladas por el modelo, se valida y estima el desempeño hipotético del mismo, concluyendo acerca de las ventajas que presenta al ser aplicado.

Para terminar, se integra al modelo final una interfase gráfica que permite al usuario alimentar y modificar los parámetros necesarios dentro del modelo, realizar una corrida de producción en cualquier momento, obtener solución y tomar decisiones al respecto.

* Trabajo de investigación

** Facultad de ingenierías fisicomecánicas – Escuela de estudios industriales y empresariales – Dirigido por Alina Fedossova

TITLE: MODELING TO PROGRAM SHORT TERM PRODUCTION IN PLANTS WITH MULTIPURPOSE BATCH, BASED IN CONTINUOUS TIMES; DEVELOPED IN ITALCOL S.C.A. BUCARAMANGA*

AUTHORS: JENNY ANGÉLICA ANGULO SOLEDAD
RONALD FERNANDO PINEDA BONILLA**

KEY WORDS: SCHEDULING, MIXED INTEGER LINEAR PROGRAMMING, CONTINUOUS TIME FORMULATION, MULTIPURPOSE BATCHES, SLOT

ABSTRACT:

This work includes the construction, development, solution and validation of a mathematical model that provides feasible solutions and offers a solid quantitative base to program its production in the short term to ITALCOL S.C.A. Bucaramanga,

The study of the production system allows to know the necessary variables and mathematical relations to represent the most real possible, its productive process and action courses and based on a previously formulated mathematical model of mixed integer linear programming, continuous times and multipurpose batches, an own language of programming in the general algebraic modeling system (GAMS), is built, compiled and through its solver CPLEX, the tasks sequences, the material amount to process at certain moment and the processing time of each task, that take place in each job shop, are obtained.

In this production system, the model is designed and solved for two programming objectives, to maximize profit and/or to minimize makespan, rapidly finding valid, feasible and favorable solutions for the organization, which support the decision making process to program operations and also those separate the subjective intervention of the one in charge. Through analysis and comparison between production historical registries and solutions found by the model, its hypothetical performance is valued and estimated, concluding about the advantages that it displays to be applied.

In order to finish, the final model is integrated to a graphical interphase that allows the user to feed and to modify the necessary parameters within the model, to make a production run at any time, to obtain solution and to make decisions on the matter.

* Investigation work

** Physics and Mechanics Engineering Collage – Industrial and Managerial Studies School – Directed by Alina Fedossova

INTRODUCCIÓN

Las organizaciones en procura de ventajas competitivas asumen una búsqueda interminable enfocada al manejo eficiente de sus recursos y a la mejora de sus procesos, siendo uno de los más relevantes el determinar un plan de producción óptimo que pueda generar ahorros significativos mientras se persigue una meta de producción o se pretende satisfacer los requerimientos del mercado.

Con la aplicación de la *programación lineal entera mixta (PLEM)* y el *modelamiento matemático* como herramientas predominantes, este trabajo formula una metodología útil como instrumento para elegir y tomar la mejor decisión en la planeación de un buen programa de producción. Es un método cuantitativo que desliga la intervención subjetiva del encargado proveyéndolo de soluciones factibles que persigan, ya sea, la maximización de la producción (beneficio) y/o la minimización del tiempo de producción (**makespan**).

Para este fin, se investigan y analizan formulaciones existentes en la literatura que proporcionen las bases de competencia necesarias para el diseño, desarrollo y ejecución de un modelo matemático y se aplica la formulación. La construcción del modelo matemático comprende las siguientes etapas: 1) formulación del problema, 2) diseño matemático del modelo, 3) ejecución del modelo, 4) solución del modelo y 5) evaluación; que se aplican al proceso productivo de la empresa ITALCOL S.C.A. Bucaramanga, empresa interesada en el proyecto a través de su Gerente de Producción.

En el transcurso del proyecto se identifican y tratan ciertas asunciones que en términos de precisión y simplificación limitan el problema y mejoran la eficiencia del modelo.

Para dar solución al modelo se emplea el código GAMS/Cplex, paquete de software que se encuentra licenciado y dispuesto para acceso y aprovechamiento, en la Escuela de Estudios Industriales y Empresariales de la UIS.

PARTE I GENERALIDADES DEL PROYECTO

1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

Las organizaciones productivas buscan una buena relación producto-mercado, deben analizar cómo se fabrica, el espacio necesario, la maquinaria y equipo, el prototipo de la producción, entre otros; que presentan cierta complejidad que puede desbordar las capacidades gerenciales u operativas de la organización. La complejidad puede manifestarse en la toma de decisiones subjetivas, intuitivas o improvisadas, con rapidez y sin un panorama amplio de alternativas. Sin embargo, los avances de la ciencia y la tecnología facilitan la construcción de escenarios mas posibles y factibles, que como beneficio consigan en los procesos productivos una mejor relación costo beneficio que se traduce en menor tiempo de producción y mayor utilidad y que contengan un componente humano en la ejecución de las labores administrativas y operativas.

Un buen escenario para la toma de decisiones gerenciales y operativas de manera objetiva, esta basado en los modelos analíticos o matemáticos formulados desde la investigación de operaciones, los cuales son métodos de simulación para facilitar la planeación eficaz y el control administrativo en los procesos de producción.

La empresa ITALCOL S.C.A. ubicada en la ciudad de Bucaramanga, se dedicada a la producción, distribución y venta de alimentos concentrados para animales, basa su modelo administrativo y operativo de carácter cualitativo que en gran parte dificulta la correlación entre las distintas variables y mayoría cualitativas, requiriéndose además de un enlace entre las acciones humanas y los métodos cuantitativos, entre el administrador y el investigador. Con una herramienta como la del modelamiento matemático se pretende facilitar con la ayuda de los computadores y del análisis cuantitativo, la toma de decisiones oportunas con mayor confiabilidad y menor incertidumbre, para mejorar los procesos de producción, disminuir costos – maximizando utilidades, disminuir tiempos – cumpliendo metas, es decir estableciendo un nuevo programa de producción que defina la secuencia de tareas para cada corrida que tiene lugar en cada puesto de trabajo, el material a procesar en determinado momento en cada puesto de trabajo y el tiempo de procesamiento de cada tarea en cada puesto de trabajo.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Construir, desarrollar, solucionar y validar un modelo matemático que contenga las variables y relaciones necesarias para representar lo más real posible el proceso productivo de ITALCOL S.C.A. Bucaramanga y sus cursos de acción, de manera que este proporcione soluciones factibles que le brinde a la organización una base cuantitativa sólida para programar su producción y comprenda: la secuencia de tareas, la cantidad de material a procesar en determinado momento y el tiempo de procesamiento de cada tarea, que tienen lugar en cada puesto de trabajo; basados en el estudio de formulaciones matemáticas para la programación de producción a corto plazo en plantas que trabajan con procesos por lotes, tiempos continuos y equipamiento multipropósito.

2.2 OBJETIVOS ESPECÍFICOS

- Identificar y analizar los procesos, las variables y relaciones que intervienen en las operaciones productivas de ITALCOL S.C.A. Bucaramanga, para generar un diagnóstico actual de dicho proceso que permita conocer y entender sus elementos, características e interrelaciones.
- Construir un lenguaje propio para el modelamiento matemático del proceso productivo de ITALCOL S.C.A. Bucaramanga, a partir del estudio de publicaciones referentes al tema.
- Validar el modelo matemático generado para corroborar que este se ajusta lo suficiente a la realidad y arroja soluciones factibles, que mejoren la eficiencia en la programación de la producción de ITALCOL S.C.A. Bucaramanga, generando ahorros significativos en esta área.

3. ALCANCE Y LIMITACIONES DEL PROYECTO

3.1 ALCANCE

El alcance del proyecto es el estudio, desarrollo y validación de un modelo matemático para programar producción a corto plazo, en ITALCOL S.C.A. Bucaramanga que trabaja con procesos por lotes, tiempos continuos y equipamiento multipropósito; a través de la programación lineal entera mixta (PLEM) y la utilización de la herramienta GAMS/CPLEX.

El desarrollo del modelo para ITALCOL S.C.A. Bucaramanga, involucra la observación, identificación y formulación del problema de programación de producción; la construcción, prueba y validación del modelo, con el fin de obtener soluciones factibles para cada corrida que comprenda: la secuencia óptima de tareas, la cantidad de material a procesar en determinado momento y el tiempo de procesamiento de cada tarea, que tienen lugar en cada puesto de trabajo. Finalmente se motivará la empresa para su implementación.

3.2 LIMITACIONES

Al aplicar la investigación de operaciones como solución para el mejoramiento de la eficiencia en las organizaciones, esto puede traer consigo distintas limitaciones que no satisfagan las expectativas de los responsables de la toma de decisiones dentro de una empresa. Como son:

- El planteamiento del modelo matemático considera sólo un objetivo (maximizar o minimizar), mientras que en la mayoría de las organizaciones se busca el cumplimiento de múltiples objetivos.
- Ninguna formulación matemática completa puede reducir los grados inherentes de libertad que cualquier planta tiene para la asignación de tareas a unidades y puntos eventos. Si alguna tratara de reducirlos, llegaría a soluciones subóptimas.
- A pesar de que el modelo tiene en cuenta todas las variables y restricciones que requiere, es muy difícil incluir situaciones imprevistas que afectan el proceso como: daños de máquinas, ausentismo, fenómenos naturales, entre otras.
- El costo que puede implicar el desarrollo e implementación de soluciones definidas por medio de la investigación de operaciones, en ocasiones puede superar los beneficios potenciales para la organización.

4. MARCO TEÓRICO

4.1 FUNDAMENTACIÓN TEÓRICA GENERAL

Tener bases teóricas en sistemas de producción, investigación de operaciones y métodos de solución para problemas de PLEM, es fundamental para contextualizar el ambiente del proyecto y adquirir las competencias intelectuales necesarias que orientan conceptual y metodológicamente el desarrollo del mismo.

4.1.1 Sistemas de Producción. Según Gavett** un sistema de producción es una colección compleja de elementos físicos (materiales, equipamiento, mano de obra, instalaciones), operada de alguna manera para proporcionar un producto o servicio.

Dice, que más que un arreglo estático el sistema es móvil y dinámico porque muchos de sus elementos aunque se mueven dentro de patrones altamente predecibles y estables sobre rutas fijas y conocidas, con el tiempo pueden cambiar los patrones o las dimensiones del sistema. Tanto el movimiento como las características dinámicas pueden describirse como relaciones cuantificables en *modelos cuantitativos* a través de la abstracción de las relaciones entre las variables del sistema que pueden ser: distancia, tiempo, masas, costos o muchas otras, siempre y cuando proporcionen sus dimensiones.

En cuanto a Administración se refiere, menciona que los responsables están comúnmente enfrentados con *problemas* que son resueltos dentro de los límites organizacionales de autoridad y responsabilidad, para esto requieren de la toma de una o varias decisiones entre los diferentes cursos de acción. El sistema de producción contiene todos los elementos o variables sobre las cuales el encargado tiene control y por lo tanto puede manipular para tomar su decisión. En contraste, el ambiente del sistema contiene variables sobre las cuales el encargado no tiene control pero que son afectadas por y/o afectan su decisión. De esta manera, cuando se habla de un sistema operacional o de producción se refiere específicamente al conjunto de elementos o variables de interés que constituyen el dominio del decisor.

Además sustenta las siguientes características de los sistemas de producción:

1. *Proceso de transformación del material:* proporciona las bases tecnológicas de producción. Aquí se hace referencia a las operaciones, que son los puntos en los

** GAVETT, William J. Production and operations management. USA: Harcourt, Brace & World, 1968. p. 8-13.

cuales los recursos son empleados para generar una transformación específica de material. A una unidad lógica de trabajo representada para un producto o servicio completado. Y al proceso que es la secuencia de operaciones requeridas para completar el trabajo.

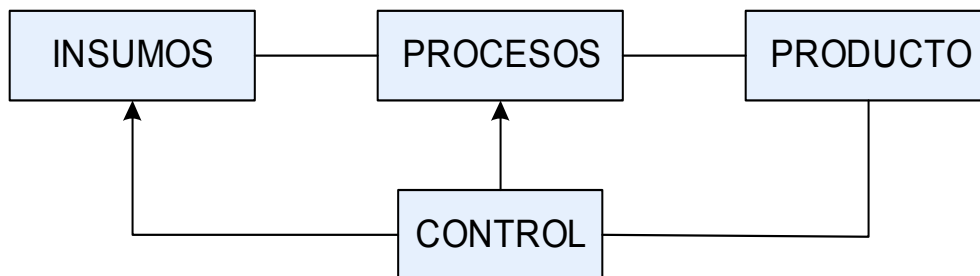
2. *Grado de repetitividad*: generalmente, hay algún patrón fundamental repetitivo en las operaciones de producción aunque con el tiempo este puede cambiar. En todo caso, se tiene en cuenta todas las clases de procesos u operaciones repetitivas bien sea en las entradas, salidas o en la misma naturaleza de la actividad productiva, que tengan como objetivo el cambio de las propiedades de los objetos o materiales físicos.

3. *Proceso de información*: sobrepuesto al sistema físico está un sistema de información. El sistema extrae desde y alimenta al proceso físico con información y datos. Así, un proceso de producción puede ser descrito como un sistema de información acrecentado. Generalmente la información cuantitativa es recogida sobre un periodo de tiempo extenso para luego ser procesada y almacenada en una *base de datos*. Una base de datos es muy importante puesto que es utilizada para propósitos de diseño, planeación y control.

4. *Proceso de flujo de material*: esta es una clase de transformación de material que corresponde al cambio de los objetos en el sistema en cuanto a espacio se refiere. El objeto del flujo puede ser un material primo, alguna pieza de equipamiento, una persona, un documento o un producto.

En la siguiente gráfica se ilustra un sistema de producción simplificado, resaltando su definición y sus elementos principales.

Figura 1. Sistema de producción



Los autores

- *Clasificación de los sistemas de producción.* Según Ibarra^{††}, cada sistema de producción caracterizado esencialmente por su proceso productivo, conlleva un conjunto de implicaciones para la empresa en cuanto al comportamiento apropiado de las diferentes dimensiones de fabricación y empresariales. Según este enfoque, concluye que en la literatura se ha aceptado por lo general la existencia de ocho tipologías de sistemas o configuraciones productivas bien definidas: *Proyecto, Job-Shop, Lotes (Batch), Línea acompañada por Equipo, Línea acompañada por Obrero, Configuración Continua, Just in Time y Sistema Flexible de Fabricación.*

Para los intereses de este trabajo, profundizar en las particularidades de la configuración por lotes, la continua y adicionalmente la multipropósito, será fundamental porque el sistema de producción involucrado contiene una mezcla de dichas características. Santiago Ibarra Mirón las define así:

1. *Configuración por lotes.* El sistema de flujo en lotes produce cierta variedad de producto en volúmenes elevados. El mayor volumen se debe a un aumento de la repetitividad en ciertos artículos que se hacen dominantes. Se requieren más operaciones y éstas son más especializadas, por lo que difícilmente un mismo operario pueda dominarlas todas con una eficiencia aceptable. En tal sentido, el trabajo se divide en diferentes etapas tecnológicas en las cuales los lotes sufren distintas operaciones. Así la instalación se suele dividir en secciones o talleres en los cuales se agrupan los equipos con funciones similares. Se suele emplear una combinación de **layouts** celulares y funcionales, utilizados cuando es efectivo en cuanto a costos, disponer el equipo en células para producir familias de productos. Como hay muchos productos, el equipamiento y utillaje son mayormente flexibles y de propósito general. El flujo de material es fraccionado aunque regular, variable de un pedido a otro. No obstante existen pautas de flujo para familias de productos y para grandes lotes. Es el sistema más utilizado.

2. *Configuración de flujo continuo.* Cada máquina y equipo están diseñados para realizar siempre la misma operación y preparados para aceptar de forma automática el trabajo suministrado por la máquina precedente. Está diseñado para fabricar un producto o una familia limitada de productos en volúmenes muy elevados. El diseño del producto es muy estable que a menudo es un producto genérico. El flujo del material es continuo, sincronizado e integrado a través de toda la instalación como si fuera un gran proceso tecnológico. Este rígido sistema, se basa en un proceso muy automatizado, costoso y especializado en la obtención de un producto estándar donde la homogeneidad es total y absoluta, funcionando

^{††} IBARRA MIRÓN, Santiago. Configuraciones productivas: conceptos y tipologías fundamentales [en línea]. Disponible en <<http://www.monografias.com/trabajos16/configuraciones-productivas/configuraciones-productivas.shtml>> [citado 26 de julio de 2006].

continuamente con mínima intervención del personal de línea. Generalmente precisa laborar las 24 horas para procurar ser un sistema costeable y eficiente.^{##}

William J. Gavett sostiene las siguientes características para la configuración de lotes multipropósito,

3. *Configuración de lotes multipropósito*. Este tipo de producción se caracteriza por fabricar una diversidad de productos así como una gran variedad de volúmenes. El sistema debe tener la flexibilidad para todo tipo de productos y volúmenes variables aun cuando los lotes de gran tamaño de productos estandarizados pueden ser procesados de una manera diferente. Cada producto puede tener un *camino* diferente, saltando o modificando el orden en que se utilizan las etapas de fabricación. En estos procesos cualquier cambio entre productos de la misma familia requiere una preparación previa de la maquinaria. La preparación supone un tiempo en que la línea de producción estará parada lo que implica a su vez un coste valorable en términos de producción no hecha, que se debe recuperar con la producción de lotes de muchas unidades y así distribuir este entre ellas. La productividad y eficiencia económica en plantas que trabajan con este tipo de producción se ven fuertemente influenciadas por el planteamiento de la producción que se ponga en práctica.

Los productos o partes de ellos son elaborados intermitentemente en lotes tal que ninguna máquina es utilizada exclusivamente para la producción de un solo producto. Hay así un uso compartido de máquinas por productos diferentes.^{§§}

4.1.2 Programación de Operaciones. Carolina Maldonado presenta su definición y características al respecto:

La programación de operaciones tiene como función determinar que operaciones se van a realizar sobre los distintos pedidos, durante cada momento del horizonte de planificación (HP), en cada centro de trabajo (CT), de forma que con la capacidad disponible en cada uno de ellos se cumplan las fechas de entrega planificadas, empleando el menor volumen de recursos e inventarios posibles. El horizonte de planificación dependerá de las características del proceso productivo y su entorno, variando entre unas horas y varias semanas.

Si la ruta de obtención de las partes es fija, el problema se centra en la determinación de la secuencia de pasos de los pedidos, aunque ahora se ha de determinar para cada equipo. Si por el contrario, algunas partes

^{##} Ibid.

^{§§} GAVETT, Op. Cit., p.483.

tienen rutas alternativas de forma que una operación puede realizarse en varios equipos diferentes, será necesario proceder a una asignación concreta antes de establecer la secuencia de paso, así:

1. *Carga de talleres*, asignación de los pedidos a los CT indicando qué operaciones se realizarán en cada uno de ellos.
 2. *Secuenciación*, establecimiento de la prioridad de paso de los pedidos en los diferentes CT para cumplir las fechas de entrega planificadas con la menor cantidad de inventarios y recursos.
 3. *Programación detallada*, determinación de los momentos de comienzo y fin de las actividades en cada CT, así como de las operaciones de cada pedido para la secuenciación realizada.^{***}
- *Programación a Corto Plazo*. Maldonado^{†††} sostiene que este tipo de programación se refiere a aquella que tiene en cuenta el tiempo de las operaciones y toma como base la programación semanal, diaria o por hora. Es usada para optimizar el funcionamiento total de la programación de la producción y puede hacerse de dos maneras:
 1. Carga del Centro de Trabajo.
 2. Secuenciación del Trabajo.

Además anota que la esencia de la programación a corto plazo es interrelacionar el tamaño de las órdenes de trabajo, las fechas deseadas de realización y la capacidad de producción de las máquinas, siendo necesario:

1. Evaluar y controlar los pedidos que se van a fabricar.
2. Establecer las prioridades entre los pedidos o trabajos a realizar.
3. Rastrear la evolución de los pedidos en curso.
4. Controlar el desarrollo de las operaciones.
5. Controlar la capacidad de cada centro de trabajo.
6. Proporcionar retroalimentación al sistema de planificación y control de capacidad.

Sostiene que los objetivos de este tipo de programación son:

^{***} MALDONADO, Carolina. Planificación, programación y control de las operaciones [en línea]. Disponible en <<http://www.monografias.com/trabajos18/direccion-operaciones/direccion-operaciones.shtml>> [citado 26 de julio de 2006].

^{†††} Ibid.

1. Minimizar el tiempo de finalización.
2. Maximizar la utilización (lo que hace efectivo el uso del personal y el equipamiento).
3. Minimizar el inventario del trabajo en curso.
4. Minimizar el tiempo de espera de los clientes.

Y por último señala que para su planificación existen diversas técnicas, de acuerdo a las características, necesidades y políticas de la organización como son:

1. Regla de Jonson
2. Procedimientos de prueba y error
3. Reglas de prioridad
4. Teoría de colas
5. Sistemas OPT, entre otros.

4.1.3 Scheduling. Para Kenneth R. Baker, el scheduling presenta la siguiente definición y características:

[Scheduling] es la asignación de recursos sobre el tiempo para realizar un conjunto de tareas. Esta definición general converge en dos significados. Primero, al determinar la programación como una función para tomar decisión y segundo, como un cuerpo de teoría con principios, modelos, técnicas y conclusiones lógicas que proporcionan visión dentro de la función.

Los objetivos que se pretenden maximizar al hacer un correcto manejo del **scheduling** y la secuenciación son:

1. Los trabajos se completan cuando vencen o el retraso de entrega es minimizado.
2. La utilización de máquinas, instalación y mano de obra es maximizada.
3. El inventario de producto en proceso es llevado a un nivel deseable.
4. Los tiempos guías son llevados a un mínimo deseable.
5. Las oportunidades de incentivo de los empleados se igualan.
6. El tiempo fuera de servicio de las máquinas debido a cambios y arreglos es minimizado.

• **La función de programación.** Para encontrar solución al problema práctico de asignación de recursos sobre el tiempo al realizar un conjunto de tareas se establece una aproximación del sistema, el cual apoya la toma de decisiones respecto al **scheduling** y comprende cuatro etapas: formulación, análisis, síntesis y evaluación. En la primera, el problema es identificado y los criterios que deben guiar la toma de decisión son

determinados. En el análisis, el proceso del examen de elementos e interrelaciones de un problema es detallado; aquí se identifican las variables de decisión y también se especifica la relación entre ellas y las restricciones que deben obedecer. La síntesis es el proceso de construcción de soluciones alternativas al problema, su papel es caracterizar las opciones factibles que están disponibles. Finalmente, la evaluación es el proceso que compara esas alternativas factibles y selecciona un curso deseable de acción. Esta selección por supuesto está basada en el criterio que fue desarrollado al principio.

Los modelos y métodos de **scheduling** son inherentes en la aproximación del sistema y proporciona las bases directas para la toma de decisiones, si dicha aproximación es utilizada. Por cierto, cuando un modelo es realmente una representación confiable de la realidad, esto puede ser una parte integral de la función del **scheduling**. Uno de los modelos más simples y más ampliamente utilizado es el diagrama de Gantt, que es una representación gráfica de las relaciones de **scheduling** entre la asignación de recursos y el tiempo.

- **Teoría del scheduling.** La teoría del **scheduling** involucra los modelos matemáticos que relacionan la función del **scheduling** y el desarrollo de modelos y técnicas útiles que han sido continuamente el vínculo entre la teoría y la práctica. La perspectiva teórica es predominantemente una aproximación cuantitativa, una de ellas que se esfuerza por capturar la estructura del problema es el modelo matemático. En particular, este comienza con una traducción de las metas dentro de la especificación de una función objetivo y unas restricciones al tomar la decisión dentro de las restricciones explícitas.

Idealmente, la función objetivo debería involucrar todos los costos en el sistema que dependen de las decisiones del **scheduling**. En la práctica, tales costos son difíciles de medir o aún más de identificar. De hecho, los mayores costos de operación (y los más identificables) son determinados por la función de planeación, mientras los costos a corto plazo son difíciles de separar y tienden a aparecer fijos. No obstante, tres tipos de metas de decisión parecen prevalecer en el **scheduling**: la utilización eficiente de los recursos, la respuesta rápida a la demanda y la conformidad en las fechas de entrega preestablecidas.

Dos tipos de restricciones factibles son comúnmente encontradas en los problemas del **scheduling**. Primero, hay límites de capacidad de los recursos disponibles y segundo, las restricciones tecnológicas en el orden en que las tareas pueden ser realizadas.

La esencia de los problemas del **scheduling** dan respuesta a: 1) las decisiones de asignación y 2) las decisiones de secuenciación.

Los elementos vitales en los modelos de **scheduling** son recursos y tareas. Los recursos generalmente se caracterizan en términos de sus capacidades cualitativas y cuantitativas, así el modelo describe el tipo y cantidad de cada recurso. Una tarea individual es descrita en términos de información tal como su requerimiento de recurso, duración, momento en el cual puede comenzar y el momento de finalización. Adicionalmente, un conjunto de tareas puede ser algunas veces descrito en términos de restricciones tecnológicas (restricciones de precedencia) que existen entre sus elementos.

La teoría del **scheduling** también incluye una amplia variedad de técnicas que son útiles para la solución de los problemas de **scheduling**. La selección de una técnica apropiada depende de la complejidad del problema, la naturaleza del modelo y la selección de criterios, entre otros. De esta manera, la teoría del **scheduling** es quizás tanto el estudio de metodologías como el estudio de modelos.^{###}

4.1.4 Secuenciación. William J. Gavett señala que:

El problema de secuenciación es un problema especializado del problema del **scheduling** en el que una orden de trabajo determina completamente una programación.

La secuenciación de trabajos establece la regla de prioridad del sistema para seleccionar un trabajo entre un conjunto de trabajos que esperan para ser utilizados por una máquina. Si hay un lote finito de trabajos para ser procesados en una sola máquina, entonces hay una combinación de $N!$ posible de secuencias u órdenes de trabajos a través de la máquina. Una regla de prioridad designará una secuencia entre un conjunto de $N!$ posibles secuencias. Una vez un trabajo es escogido de un conjunto de trabajos en espera, la regla de prioridad puede ser cambiada, así que más de una regla puede ser aplicada para el conjunto de N trabajos. Se puede escoger de un gran número de reglas de prioridad para gobernar la selección de trabajos compitiendo por el tiempo de procesamiento en una máquina dada. Aquí hay algunos ejemplos:

1. El uso de una regla aleatoria tal que cada trabajo tenga la misma probabilidad de ser seleccionada.

^{###} BAKER, Kenneth R. Introduction to sequencing and scheduling. USA: John Wiley & Sons, 1974. p. 2-7.

2. El uso de la regla primero en entrar primero en salir; tal que la prioridad más alta es dada al trabajo que llegue primero a la máquina.
3. La prioridad más alta se asigna al trabajo que requiera la menor cantidad de tiempo de procesamiento en la máquina.
4. La prioridad más alta se asigna al trabajo que tiene la fecha de entrega más temprana.
5. Se selecciona el trabajo que requiera la menor cantidad de tiempo de instalación relativo al trabajo que ha acabado de ser completado en la máquina.
6. Se selecciona el trabajo que tiene el menor número de operaciones restantes para su finalización.

Esta lista puede expandirse fácilmente para incluir una gran variedad de esquemas de prioridad.^{§§§}

4.1.5 Investigación operativa y optimización. Como dice Linares^{****}, la investigación operativa se puede definir como la aplicación de métodos científicos en la mejora de la efectividad en las operaciones, decisiones y gestión. Utilizando otra definición más extensa puede decirse que es la aplicación por grupos interdisciplinarios del método científico, a los problemas complejos producidos en la dirección y gestión de grandes sistemas de hombres, máquinas, etc. La principal característica consiste en construir un modelo científico del sistema del cual se puede predecir y comparar los resultados de diversas estrategias, decisiones, incorporando medidas del azar y del riesgo. El objetivo es ayudar a los responsables a determinar su política y actuaciones en forma científica.

La optimización es una parte relevante dentro del proceso de la investigación operativa y consiste en la selección de una mejor alternativa en algún sentido que las demás alternativas posibles. Ciertas técnicas propias de la investigación operativa se conocen bajo el nombre de optimización o programación matemática. Los problemas de optimización se componen generalmente de:

1. Función objetivo
2. Variables
3. Restricciones

^{§§§} BAKER, Op. cit., p. 540-541.

^{****} LINARES, Pedro et al. Modelos matemáticos de optimización [en línea]. p. 5. (octubre 2001). Disponible en <www.gams.com/docs/contributed/modelado_en_gams.pdf> [citado 29 de julio de 2006].

Resolver un problema de optimización como estos, consiste en encontrar el valor que deben tomar las variables para hacer óptima la función objetivo satisfaciendo el conjunto de restricciones.

A grandes rasgos los métodos de optimización se clasifican en: métodos clásicos y métodos metaheurísticos. Dentro de los primeros se encuentran la optimización lineal, lineal entera mixta, no lineal, estocástica, dinámica, etc. En el segundo grupo se incluyen los algoritmos evolutivos (genéticos entre otros), el método del recocido simulado (**simulated annealing**) o las búsquedas heurísticas (método de tabú, búsqueda aleatoria, etc.). De forma muy general y aproximada se puede decir que los métodos clásicos buscan y garantizan un óptimo local mientras los métodos metaheurísticos tienen mecanismos específicos para alcanzar un óptimo global, aunque no garantizan su alcance.

4.1.6 Modelamiento matemático. Linares^{†††} argumentan que:

Un modelo matemático se define como una descripción desde el punto de vista de las matemáticas de un hecho o fenómeno del mundo real, desde el tamaño de la población hasta fenómenos físicos como la velocidad, aceleración o densidad. El objetivo del modelo matemático es entender ampliamente el fenómeno y tal vez predecir su comportamiento en el futuro. Un modelo en definitiva es una herramienta que ayuda a la toma de decisiones, por esta razón sus resultados deben ser inteligibles y útiles.

Un modelo matemático comprende principalmente tres conjuntos básicos de elementos. Estos son: (1) variables y parámetros de decisión, (2) restricciones y (3) funciones objetivo.

1. *Variables y parámetros de decisión*: las variables son las incógnitas (o decisiones) que deben determinarse resolviendo el modelo. Los parámetros son los valores conocidos que relacionan las variables de decisión con las restricciones y la función objetivo. Desde el punto de vista funcional se pueden clasificar en variables *independientes* o *principales* o *de control* y variables *dependientes* o *auxiliares* o *de estado*, aunque matemáticamente todas son iguales.

2. *Restricciones*: son igualdades o desigualdades que tienen en cuenta las limitaciones tecnológicas, económicas y otras del sistema; que restringen las variables de decisión a un rango de valores factibles.

3. *Función objetivo*: es una función matemática de las variables de decisión que define la medida de efectividad del sistema, permitiendo conocer el nivel de logro

^{†††} Ibid., p. 11-14.

de los objetivos.

Un modelo siempre debe ser menos complejo que el problema real, es una aproximación abstracta de la realidad con consideraciones y simplificaciones que hacen más manejable el problema y permiten evaluar eficientemente las alternativas de solución.

Menciona las etapas que componen el ciclo de un modelo y las expresa así:

1. *Identificación del problema.* Consiste en recolectar y analizar la información relevante para el problema.

2. *Especificación matemática y formulación.* Los problemas reales suelen estar definidos en términos vagos e imprecisos, se debe hacer la tarea de traducción o interpretación en frases precisas, convertibles en ecuaciones matemáticas, definiendo sus variables, su función objetivo y sus restricciones. Aquí se analiza el tamaño del problema, la estructura de la matriz de restricciones y el tipo (PL, PEM, PNL).

3. *Resolución.* Se trata de implantar un algoritmo de obtención de la solución numérica (muy próxima a la matemática), óptima o cuasióptima, que puede ser de propósito general o específico.

4. *Verificación, validación y refinamiento.* Conlleva la eliminación de los errores en la codificación, es decir, conseguir que el modelo haga lo que se desea (depurar y verificar). Es necesario comprobar la validez de las simplificaciones realizadas, a través de los resultados obtenidos (validar).

5. *Interpretación y análisis de los resultados.* Consiste en proponer soluciones por medio de un análisis de sensibilidad.

6. *Implantación, documentación y mantenimiento.* La mayor parte del ciclo de vida de un modelo no está en el desarrollo sino en el uso y mantenimiento. Esto incluye la tarea de formación para los usuarios del modelo.

4.1.7 Programación lineal entera mixta. En el documento PROGRAMACIÓN LINEAL entera mixta⁺⁺⁺ se establece que:

La *programación lineal* (PL) es un procedimiento matemático utilizado para determinar la asignación óptima de recursos escasos y consta de los

⁺⁺⁺ Programación lineal entera-mixta [en línea]. Disponible en <departamentos.unican.es/macc/personal/profesores/castillo/Libro/Chap2.pdf> [citado 29 de julio de 2006]

siguientes elementos: datos, variables, restricciones y función a optimizar. El objeto de la programación lineal es optimizar (minimizar o maximizar) una función lineal de n variables sujeto a restricciones lineales de igualdad o desigualdad. Más formalmente, se dice que un problema de programación lineal consiste en encontrar el óptimo de una función lineal en un conjunto que puede expresarse como la intersección de un número finito de hiperplanos y semiespacios en \mathbb{R}^n .

La forma más general de un *problema de programación lineal* (PPL) consiste en minimizar o maximizar

$$Z = f(x) = \sum_{j=1}^n c_j x_j$$

Sujeto a:

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad i = 1, 2, \dots, p-1$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = p, \dots, q-1$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = q, \dots, m$$

Donde p , q y m son enteros positivos tales que

$$1 \leq p \leq q \leq m$$

Los problemas lineales presentan propiedades que hacen posible garantizar el óptimo global:

1. Si la región factible está acotada, el problema siempre tiene una solución (ésta es una condición suficiente pero no necesaria para que exista una solución).
2. El óptimo de un problema de programación lineal es siempre un óptimo global.
3. Si x y y son soluciones óptimas de un problema de programación lineal, entonces cualquier combinación (lineal) convexa de lo mismos también es una solución óptima.

4. La solución óptima se alcanza siempre, al menos, en un punto extremo de la región factible.

Dado que un PPL puede plantearse de diversas formas, para unificar su análisis es conveniente transformarlo en lo que normalmente se llama forma estándar. Para describir un PPL en forma estándar son necesarios los tres elementos siguientes:

1. Un vector $c \in \mathbb{R}^n$
2. Un vector no negativo $b \in \mathbb{R}^m$
3. Una matriz $m \times n$, A

Con estos elementos el problema lineal asociado y en forma estándar tiene la siguiente forma. Minimizar:

$$Z = c^T x$$

Sujeto a:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

Donde $c^T x$ indica un producto escalar de los vectores c y x , Ax es el producto de la matriz A y el vector x , y $x \geq 0$ hace que todas las componentes de los vectores factibles sean no negativas. Los problemas de programación lineal se estudian normalmente en esta forma. Típicamente, n es mucho mayor que m . En resumen, un problema de programación lineal se dice que está en forma estándar si y solo si:

1. Es de minimización.
2. Solo incluye restricciones de igualdad.
3. El vector b es no negativo.
4. Las variables x son no negativas.

Sus principales características son: la facilidad relativa del método de solución, la gran disponibilidad de paquetes de software de PL y la amplia gama de aplicaciones que incluye casi todas las facetas de los negocios, desde la publicidad hasta la planificación de la producción. Además, la PL brinda una excelente oportunidad para presentar la idea del análisis de hipótesis ya que se han desarrollado herramientas poderosas para el análisis de post optimalidad para el modelo de PL.

Existen decisiones que no pueden ser representadas de forma adecuada

mediante variables continuas. En este caso se utilizan las variables discretas o binarias mediante la programación lineal entera mixta (PLEM), donde los problemas son lineales y algunas o todas las variables son enteras. Los problemas lineales con variables enteras se pueden clasificar en: programación entera pura (PEP) si todas las variables son enteras, programación entera binaria (PEB) si todas son binarias y programación entera mixta (PEM) si algunas son enteras o binarias y el resto continuas.

Un caso particular pero muy frecuente de variables enteras son las variables binarias (0/1), ya que permiten modelar condiciones de asignación o condiciones lógicas.

4.2 FUNDAMENTACIÓN EN LITERATURA CONCERNIENTE

4.2.1 Modelamiento matemático a través de la PLEM para la producción continua, a corto plazo y de lotes multipropósito.

• *Conceptos generales.* Sundaramoorthy y Karimi^{§§§§} sostienen que, los esfuerzos investigativos utilizando representación de tiempos continuos en la programación de procesos con lotes han optado por clasificarlos en dos:

1. Las formulaciones basadas en **slot**: representación del tiempo en términos de bloques ordenados de longitud variables desconocidas.
2. Las formulaciones basadas en puntos eventos: utilización de puntos de tiempo desconocidos llamados eventos que representan el comienzo de una tarea en una unidad.

Representaciones basadas en **slots** como en puntos eventos son clasificadas dentro de dos tipos:

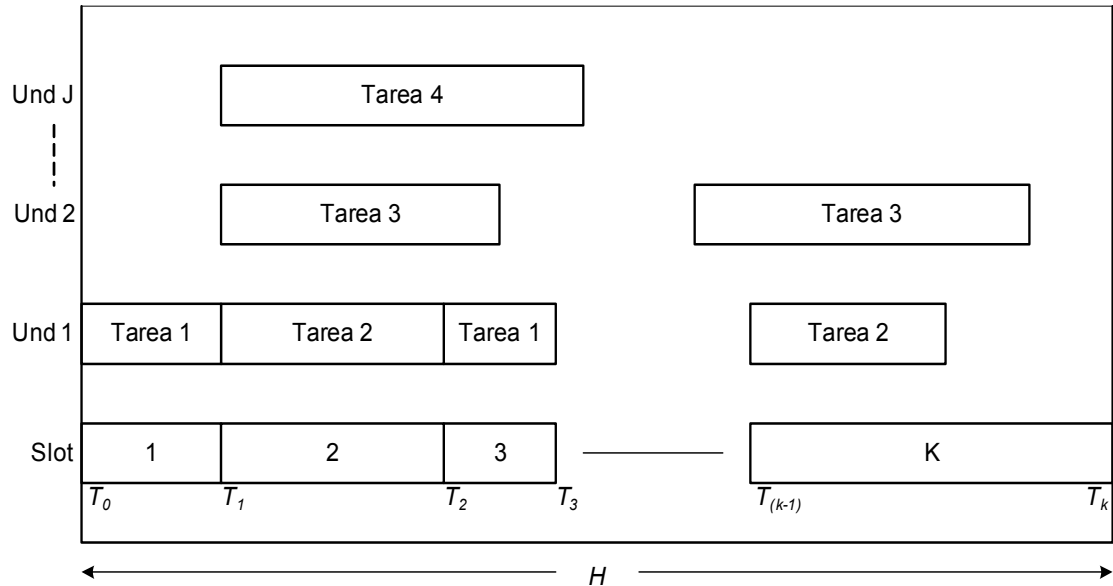
1. Sincrónicos o común: **slots** o puntos eventos están sincronizados o comunes a través de todas las unidades en una planta.
2. Asincrónicos o no común: **slots** o puntos de eventos difieren de una unidad a otra.

En la figura 2 se muestra el diagrama Gantt para la representación de tiempos

^{§§§§} SUNDARAMOORTHY, Arul and KARIMI, I.A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *En*: Chemical Engineering Science [base de datos en línea]. Vol. 60 (may., 2005); p. 2679. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

continuos mediante **slots** sincrónicos que Sundaramoorthy y Karimi^{*****} plantearon en su formulación en el caso de J unidades, cuatro tareas, K **slots** y un horizonte de planificación H .

Figura 2. Representación de **slots** sincrónicos



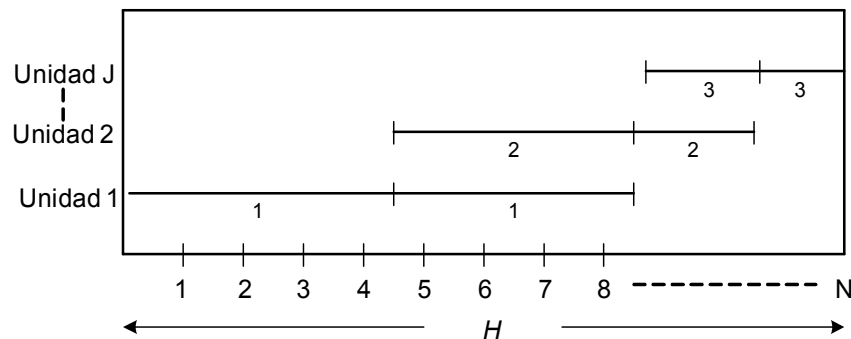
Arul Sundaramoorthy and I.A. Karimi

A continuación en la figura 3 se ilustra la representación de tiempos continuos mediante puntos eventos asincrónicos que Ierapetritou y Floudas^{††††} diseñaron en su formulación para el caso de J unidades, tres tareas, N puntos eventos y un horizonte de planificación H .

***** Ibid., p. 2683.

†††† IERAPETRITOU, M. G. and FLOUDAS C. A. Effective continuous-time formulation for short-term scheduling: multipurpose batch plant. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 37, No. 11 (oct., 1998); p. 4349. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

Figura 3. Representación de puntos eventos



M.G. Ierapetritou and C.A. Floudas

Aunque ambas representaciones en principio pueden manejar recursos compartidos tales como materiales, la representación asincrónica hace más fácil y natural dicho manejo.

- **Estado del arte.** La programación de la producción a corto plazo en plantas con lotes multipropósito es un problema cambiante para el cuál existen formulaciones en la literatura que han recibido atención considerable en la última década.

Maravelias y Grossmann^{####} cuentan que los primeros ensayos utilizaron formulaciones de *programación lineal entera mixta* (PLEM) basadas en la representación uniforme de tiempo discreto e introdujeron el concepto de **State Task Network** (STN). Luego en 1995 se propuso la formulación y representación alternativa de **Resource Task Network** (RTN), en 1996 fue desarrollado un modelo de PLEM y tiempos continuos basados en la representación RTN y un nuevo algoritmo **branch & bound** que ramifica tanto variables discretas como continuas. Entre 1996 y 1999 se formuló una representación de tiempos continuos con *programación no lineal entera mixta* (PNLEM) para la programación de procesos continuos por lotes. Mientras tanto, Ierapetritou y Floudas en 1998 presentaron una nueva formulación asincrónica de PLEM basada en puntos eventos para la programación de plantas con lotes multipropósitos y tiempos continuos, pero que trajo una discrepancia en el balance de material.

A pesar de las formulaciones desarrolladas, de los algoritmos especializados y los recientes desarrollos computacionales en hardware y software de optimización, la

MARAVELIAS, Christos T and GROSSMANN, Ignacio E. A new general continuous-time state task network formulation for short-term scheduling of multipurpose batch plants. *En: Industrial and Engineering Chemistry Research* [base de datos en línea]. Vol. 42 (mar., 2003); p. 3056-3057. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

programación a corto plazo en plantas con lotes multipropósito mediante representación STN de tiempos continuos aún tiene dificultades para resolverse. Así, en un esfuerzo por reducir el tamaño del problema y el tiempo computacional durante los años 2000 y 2001, diversos autores plantearon diferentes métodos donde se hicieron suposiciones específicas que por un lado guiaron a las formulaciones a ser más compactas, pero por otro lado solo se dirigieron a casos específicos del problema general de programación a corto plazo. Algunas de las suposiciones más comunes que resultaron significantes en la reducción del tamaño del problema son: (a) ningún lote es dividido y (b) no hay ninguna restricción de recurso excepto aquellas que se refieren a las unidades de equipamiento.

Sundaramoorthy y Karimi^{§§§§§} comentan que para contrarrestar los inconvenientes presentados hasta ese momento, Giannelos y Georgiadis en 2002 desarrollaron una formulación donde utilizaron puntos eventos sincrónicos con algunas restricciones extras de **timing**/secuencia y un concepto de holgura de tiempo. Sin embargo, su método apuntó a soluciones subóptimas pues se presentó una dificultad en la duración óptima de las tareas. Maravelias y Grossmann en el 2003 utilizaron puntos de tiempo sincrónicos en su formulación para la programación a corto plazo de plantas con lotes y evitaron las restricciones extras de **timing**/secuencia que Ierapetritou y Floudas en 1998 y Giannelos y Georgiadis en 2002 la utilizaron explícitamente. Actualmente Arul Sundaramoorthy e I. A. Karimi (2005) propusieron una nueva formulación de *programación lineal entera mixta* (PLEM) para la programación a corto plazo de plantas con lotes multipropósito mediante el uso de **slots** sincrónicos y demuestran que su formulación es más sencilla, más eficiente, y puede potencialmente ser más concreta que las formulaciones existentes.

4.2.2 Características generales de las formulaciones estudiadas

- ***Effective continuous-time formulation for short-term scheduling in multipurpose batch processes. Ierapetritou y Floudas^{*****}***

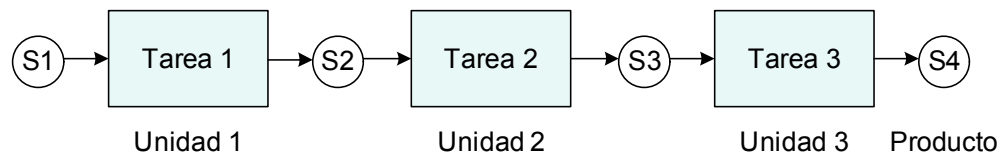
Esta formulación se basa en la representación de tiempos continuos y puntos eventos y resulta un problema de *programación lineal entera mixta* (PLEM). Presenta nuevos elementos con respecto a publicaciones anteriores, como son: 1) Disociación de tareas y unidades, generando dos variables binarias para las restricciones de asignación $wv_{(i,n)}$ y $yv_{(i,n)}$; para todo i que pertenece al conjunto de tareas, j que pertenece al conjunto de unidades y n que pertenece al conjunto de puntos eventos; 2) la definición y utilización de un nuevo conjunto de restricciones

§§§§§ SUNDARAMOORTHY, Op.cit., p. 2680.

***** IERAPETRITOU, Op. cit., 4341-4359.

llamadas restricciones de secuencia de tiempo, las cuales proveen una conexión entre el comienzo y final de tiempos y las variables binarias de asignación; 3) el tiempo de procesamiento variable es considerado en esta publicación y varía de acuerdo a la cantidad de material siendo procesado en una tarea específica; 4) la representación gráfica utilizada es el **state-task-network** (STN), que describe el flujo de planta representando todos los materiales como estados procesados a través de un conjunto de etapas de procesamiento, como se ve a continuación:

Figura 4. Ejemplo de representación STN

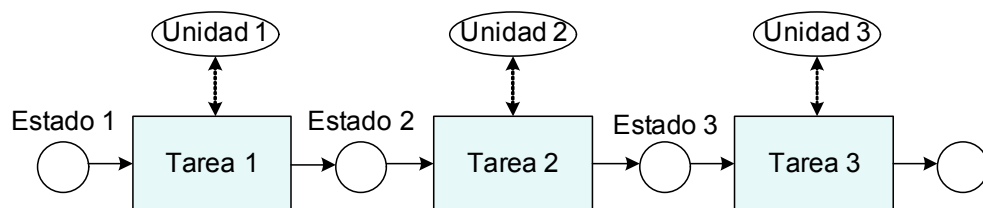


M.G. Ierapetritou y C.A. Floudas

- **An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants.** Castro, Barbosa-Póvoa y Matos⁺⁺⁺⁺⁺.

La representación gráfica que utiliza esta formulación es el **resource-task-network** (RTN), en la que se consideran todos los procesos como gráficos bipartidos comprimiendo dos tipos de nodos, recursos y tareas (figura 5) y así, no realiza la separación entre tareas y unidades. Aquí el concepto de recurso abarca todas las entidades que comprende el proceso (materiales, procesamiento, equipos de almacenamiento y utilidades varias).

Figura 5. Ejemplo de representación RTN



P. Castro, A.P.F.D. Barbosa-Póvoa y H. Matos

+++++ CASTRO, P; BARBOSA-PÓVOA, A. P. F. D. and MATOS, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 40, No. 9 (mar., 2001); p. 2059-2068. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

La representación del tiempo basada en puntos eventos, adopta una red de tiempo común para todos los recursos, es decir, se basa en puntos eventos sincrónicos. Debido a esto, no se hace necesario la utilización de restricciones adicionales de secuencia necesarias cuando cada unidad tiene una red de tiempo no uniforme, simplificando el problema y haciéndolo más seguro. Aunque la diferencia entre resultados de función objetivo comparados con formulaciones anteriores es insignificante, en términos del modelo matemático resultan importantes.

La característica fundamental de esta formulación se basa en políticas de espera de material en equipos de procesamiento y así, se presenta una disminución en el número de puntos eventos. Debido a esto el modelo necesita restricciones adicionales para ajustar un problema que considere políticas de cero esperas.

Esta formulación también busca la asignación óptima de recursos a tareas sobre el horizonte de programación, tanto como para garantizar un objetivo predefinido operacional y/o económico. Así, el modelo matemático en cuestión tiene que: caracterizar la ocurrencia potencial de eventos (comienzo o terminación de tareas) en cada punto de tiempo, los intervalos de tiempo más considerados y adoptar el tamaño más grande del modelo matemático.

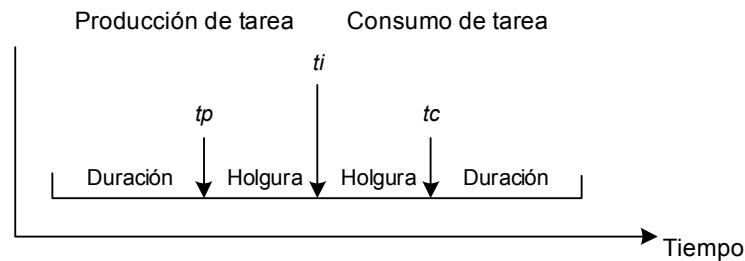
- ***A Simple New Continuous-Time Formulation for Short-Term Scheduling of Multipurpose Batch Processes. Giannelos y Georgiadis^{#####}.***

Para esta formulación la representación gráfica utilizada es la **state-task-network** (STN), la cual es representada para una red de tiempo no uniforme con un número de puntos eventos n prepostulados, definidos hacia el final de una tarea y con características de sincronía. Dado esto, una tarea que puede ser realizada en j unidades (equipamiento compartido) se asume como la realización de j tareas, es decir, las tareas i contienen implícitamente la información de las unidades.

Con respecto a la duración de tareas, los autores presentan en su trabajo la utilización de un término de relajación adicional llamado **buffer-time**. Entonces la duración total de la tarea se compone de un término constante de procesamiento, un término variable dependiente del tamaño del lote y el **buffer time**, siendo este, un intervalo de tiempo en el que la tarea produce o consume una cierta cantidad de material desde y para un estado. El **buffer** de tiempo para el caso de material producido puede ser cero cuando se utilizan políticas de cero esperas de material.

GIANNELOS, Nikolaos F. and GEORGIADIS, Michael C. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 41, No. 9 (abr., 2001); p. 2178-2184. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

Figura 6. Ejemplo de representación de tarea



Nikolaos F. Giannelos y Michael C. Georgiadis

Para la figura 6 se considera tp como el tiempo de producción, ti como el tiempo de intercambio de material y tc el tiempo de consumo de la tarea.

Otra característica de la formulación es la eliminación de puntos eventos innecesarios a través del uso de una red de tiempo no uniforme y una sencilla relajación de la duración de tareas. Para las interacciones entre estados y tareas, esta formulación requiere el uso de restricciones adicionales de secuencia que junto a las restricciones de duración transforman efectivamente la red de tiempo no uniforme a una uniforme, asegurando que el problema de la conservación de material y tamaño de lote este bien formulado a través del dominio del tiempo continuo.

• ***A new general continuous-time state task network formulation for short-term scheduling of multipurpose batch plants. Maravelias y Grossmann***^{§§§§§§}.

En esta formulación, el modelo propuesto utiliza la representación **state-task-network** (STN), para una red de tiempo común, teniendo en cuenta restricciones de recursos no solo para unidades, tiempos de procesamiento y tamaño de lotes variable y diferentes políticas de inventario. Las características claves del modelo propuesto son:

1. Se utiliza una representación de tiempos continuos, común para todas las unidades.
2. Las restricciones de asignación son expresadas mediante el uso de variables binarias que son definidas únicamente por las tareas, no por las unidades.
3. Los tiempos de inicio de las tareas son eliminados, así, las restricciones de ajuste de tiempo son utilizadas únicamente para los tiempos finales de las tareas.

^{§§§§§§} MARAVELIAS, Op. cit., p. 3056-3074.

4. Se incluyen una nueva clase de desigualdades válidas para mejorar la relajación de programación lineal en programación lineal entera mixta.

5. Para reducir el número de variables binarias usan la idea de disociación de tareas para la asignación de unidades.

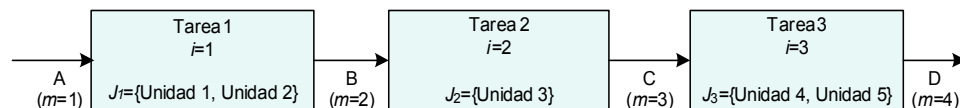
Esta formulación comparada con las anteriores, resultó ser más rápida y en comparación con las que manejan eventos, es más general por lo que con frecuencia se obtienen mejores soluciones en menor tiempo computacional.

• ***A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants.***
Sundaramoorthy y Karimi^{*****}.

Esta formulación basada en **slots** sincrónicos y tiempos continuos divide el tiempo horizonte en k **slots** fijando un número inicial para k y aumentando gradualmente este número hasta obtener la cantidad de **slots** adecuada. Además, no separa tareas de unidades como lo hacen otras publicaciones (ej. Ierapetritou y Floudas, 1998) y para esto define una sola variable de asignación con tres subíndices $y(i,j,k)$, representando i tareas, j unidades y k **slots**. Sin embargo, el uso de estos dos tipos de variables no difieren en el número total de variables de asignación utilizadas en un problema específico, dado que variables 2 índices (i,n) contienen implícitamente el subíndice j detrás de i . El número de variables binarias por punto evento tiene que ser el mismo; de otra manera, la formulación no dará la solución óptima. Proponen un nuevo concepto de balances (tiempo, recursos, materiales, etc.) en la formulación del problema de programación. Esta formulación también es igualmente efectiva para los dos objetivos de programación (maximización de beneficios y minimización de **makespan**) comúnmente usados en la literatura para este problema.

La representación gráfica utilizada, como se ve en la figura 7, es el diagrama de receta (RD), que mediante nodos y arcos describen el flujo de planta, donde los nodos son las tareas, los arcos los materiales (material-estado) y la dirección del arco describe la tarea precedente:

Figura 7. Ejemplo de representación RD



Arul Sundaramoorthy y I.A. Karimi

***** SUNDARAMOORTHY, Op. cit, p. 2679-2702.

Para resumir todo lo anterior, en el siguiente cuadro se presentan los aspectos relevantes, comunes (color verde), diferentes (color azul) y particulares (colores amarillo y naranja) para las cinco formulaciones estudiadas en profundidad.

Cuadro 1. Comparativo de similitudes y diferencias en las formulaciones estudiadas

I&F (1998)	C, BP &M (2001)	G&G (2002)	M&G (2003)	S&K (2005)
Tiempos continuos	Tiempos continuos	Tiempos continuos	Tiempos continuos	Tiempos continuos
PLEM	PLEM	PLEM	PLEM	PLEM
Decopla tareas de unidades	No decopla tareas de unidades	No decopla tareas de unidades	Decopla tareas de unidades	No decopla tareas de unidades
Restricciones adicionales de secuencia	No restricciones adicionales de secuencia	Restricciones adicionales de secuencia	No restricciones adicionales de secuencia	No restricciones adicionales de secuencia
Tiempos de procesamiento variable	Tiempos de procesamiento variable	Tiempos de procesamiento variable adicionando un tiempo de holgura	Tiempos de procesamiento variable	Tiempos de procesamiento variable
Representación STN	Representación RTN	Representación STN	Representación STN	Representación RD
Puntos eventos asincrónicos	Puntos eventos sincrónicos	Puntos eventos asincrónicos	Puntos de tiempo sincrónicos	Slots sincrónicos
Desarrollada para maximizar beneficio	Desarrollada para maximizar beneficio	Desarrollada para maximizar beneficio	Desarrollada para minimizar makespan y maximizar beneficio	Desarrollada para minimizar makespan y maximizar beneficio
			Diferentes políticas de inventario	No utiliza restricciones gran-M
			División y mezcla de lotes	Nuevo concepto de balances de material, tiempo e inventario

Los autores

De acuerdo al análisis y características de las formulaciones, se decide aplicar la formulación de Sundaramoorthy y Karimi (S&K) por la sencillez y otras ventajas que presenta con respecto a las demás. En el siguiente apartado dicha formulación es presentada al detalle.

4.2.3 A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. Sundaramoorthy y Karimi^{†††††††}.

- *Notación*

- *Indices*

i tarea
 j unidad
 k/n slot/ o punto evento
 m material

- *Superíndices*

L límite inferior
 U límite superior

- *Conjuntos*

I Tareas
 I_j Tareas que la unidad j puede realizar
 J Unidades
 J_i Unidades que pueden realizar la tarea i
 II_m Tareas que consumen material m
 K/N Slots/ o puntos eventos
 M Materiales
 M_i Materiales que son o producidos o consumidos por la tarea i
 OI_m Tareas que producen material m

- *Parámetros*

B_{ij}^L Tamaño mínimo de lote de i en j
 B_{ij}^U Tamaño máximo de lote de i en j
 D_m Demanda fijada de material m
 g_m Utilidad neta o beneficio por unidad (kg or mu) de m
 H Horizonte de programación
 I_m^U Máxima capacidad de almacenamiento disponible para m
 I_{m0} Nivel de Inventario inicial de m

††††††† Ibid.

- *Símbolos griegos*

- α_{ij} Tiempo fijo de procesamiento de la tarea i en j
- β_{ij} Tiempo variable de procesamiento de la tarea i en j
- μ_i Material primario que la tarea i produce
- σ_{mi} Coeficiente de materiales
- τ_{ij} Tiempo constante de procesamiento de la tarea i en j

- *Variables*

- Continuas

- b_{ijk} Cantidad de lote que existe en la unidad j justo antes del comienzo de una nueva tarea en T_k
- B_{ijk} Tamaño de lote de la tarea i que la unidad j comienza en T_k
- BE_{ijk} Tamaño de lote descargado por la tarea i en su finalización en T_k
- I_{mk} Nivel de inventario de m en T_k
- SL_k Longitud de slot k que se extiende desde $T_{(k-1)}$ a T_k
- T_{jk} Tiempo restante en T_k para completar la tarea en progreso en el slot k en la unidad j
- T_k Tiempo en el cual el slot k finaliza

- Binarias

- Y_{ijk} 1 si la unidad j comienza la tarea i en el momento T_k

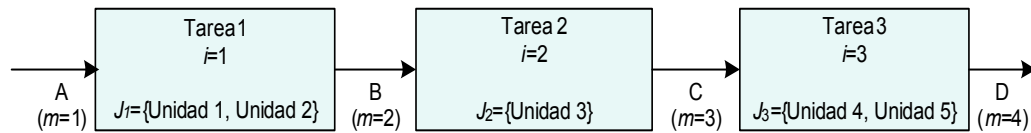
- Continuas 0–1

- y_{ijk} 1 si la unidad j está comprometida con la tarea i en el momento T_k
- YE_{ijk} 1 si la unidad j descarga material(es) de la tarea i en el momento T_k
- Z_{jk} 1 si la unidad j comienza una tarea en el momento T_k

- **Formulación del problema.** Para describir el flujo de producción se utilizan los “diagramas de receta” (RDs). En un RD, los nodos representan las tareas, los arcos los diferentes materiales y la dirección de los arcos la tarea precedente. Aquí el término material significa en general una combinación material-estado. De esta manera, el RD utiliza únicamente un conjunto de nodos que denota tareas y evita la necesidad de usar un segundo conjunto para representar estados de material. Una planta en general puede incluir uno o más RD’s por separado. Se utiliza RD únicamente para describir lo que a recursos se refiere (unidades de producción y materiales). Para mayor claridad, se muestra el siguiente ejemplo. La figura 8 muestra el RD para una planta multipropósito que al producir un producto, requiere de dos intermedios. $m_4=D$ denota el producto final, $m_2=B$ y $m_3=C$ intermedios y $m_1=A$ la materia prima. Cada rectángulo representa una tarea i y

muestra información acerca de las unidades j que pueden procesar la tarea i . Como se mencionó, se utilizan las flechas y sus direcciones para representar materiales y precedencia de tareas, respectivamente.

Figura 8. Representación RD



Arul Sundaramoorthy y I.A. Karimi

Inicialmente para cada tarea i se propone el balance general de material así:

$$\sum_{m \in M_i} \sigma_{mi} (\text{Material } m) = 0$$

Donde σ_{mi} es análoga al coeficiente estequiométrico de una clase en una reacción química que puede estar en unidades de kg/kg en lugar de mol/mol. $\sigma_{mi} < 0$, si la tarea i consume material $m \in M_i$ y $\sigma_{mi} > 0$ si esta produce $m \in M_i$. Lo que quiere decir que la suma total de materiales m que la tarea i consume, es igual a la suma total de material m que produce dicha tarea.

Para la programación a corto plazo de cualquier planta con lotes multipropósito se necesita determinar:

1. La secuencia óptima y el horario de diferentes tareas en cada unidad.
2. El tamaño del lote de cada lote de cada tarea en cada unidad en diferentes momentos utilizando:
 - a) RD para la planta con requisitos de material y unidad.
 - b) Las unidades convenientes (proceso y almacenaje) para tareas, sus límites de capacidad y la información del tiempo de procesamiento de los lotes.
 - c) El tiempo horizonte H para la maximización de beneficio o demandas de producto fijas D_m para la minimización del **makespan** (tiempo).
 - d) Utilidad en lo posible neta, del producto final.

Para esta formulación se asume lo siguiente:

1. La transferencia de material y los tiempos de preparación de la maquinaria se incluyen dentro del tiempo del procesamiento de tareas del lote.

2. El tiempo de procesamiento del lote de la tarea i en la unidad j es o bien una constante τ_{ij} o varía linealmente con su tamaño de lote como $\tau_{ij} + \beta_{ij} \cdot (\text{tamaño de lote})$, donde τ_{ij} y β_{ij} , son conocidas.

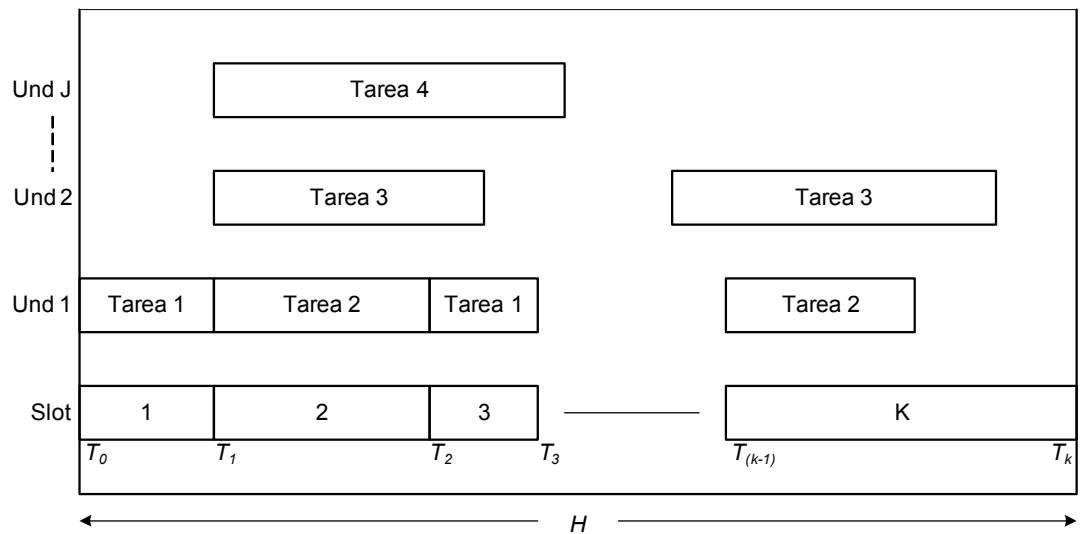
3. La utilidad del producto justifica los costos variables de producción.

- **Modelo matemático.** Para iniciar se divide el horizonte de programación H en K ($k=1, 2, \dots, K$) **slots** de longitudes variables SL_k como se muestra en la figura 9. $k=0$ denota el **slot** anterior al momento cero. Los **slots** son comunes o sincrónicos a todas la unidades ($j=1, 2, \dots, J$) y K se fija antes o se incrementa gradualmente hasta que se adecuan los **slots**. T_k representa el momento en el cual el **slot** k finaliza. $T_0=0$ representa el inicio del horizonte, mientras que $T_k \leq H$ ocurre antes de H . Normalmente cada T_k corresponde al inicio de una tarea en una o más unidades pero necesariamente esto no es así. Una tarea iniciando en T_k puede terminar antes, en, o después $T_{(k+1)}$, como se muestra en la figura 9.

Para el **slot** k que va desde $T_{(k-1)}$ hasta T_k , se tiene que:

$$T_k = T_{(k-1)} + SL_k \tag{1}$$

Figura 9. Representación de **slots**



Arul Sundaramoorthy y I.A. Karimi

Además, la suma de todas las longitudes de **slots** no debe exceder el horizonte H , así:

$$\sum_{k=1}^K SL_K \leq H \quad (2)$$

El inicio de cada nueva tarea en cualquier unidad acciona un **slot**, aunque un nuevo **slot** puede empezar aún sin el inicio de una tarea. Similarmente, diferentes tareas pueden comenzar al mismo tiempo en unidades diferentes. Este tipo de representación permite acomodar los **slots** extra y/o redundantes cuando se sobreestima K . Además, se define una tarea cero ($i=0$) para modelar el estado ocioso de las unidades y para ocupar los **slots** extra y/o redundantes. Entonces se dice que hay $(I+1)$ tareas, I real ($i=1, 2, \dots, I$) y una tarea ociosa ($i=0$).

Con este preámbulo se procede a decidir qué tareas comienzan o finalizan, en qué momentos, en cuales unidades y en cuántas cantidades. Con este fin, se define una variable binaria para el comienzo de una tarea como sigue:

$$Y_{ijk} = \begin{cases} 1 & \text{si la unidad } j \text{ comienza la tarea } i \text{ en el momento } T_k, \\ 0 & \text{de otra manera,} \end{cases}$$

$$i \in I_j, \quad 0 \leq k < K$$

Si una tarea i solo continua en la unidad j en T_k , entonces $Y_{ijk} = 0$. Basados en la binaria Y_{ijk} , se define una variable continua 0–1 Z_{jk} para saber si una unidad j comienza una tarea nueva en T_k :

$$Z_{jk} = \begin{cases} 1 & \text{si la unidad } j \text{ comienza una tarea (incluyendo } i = 0) \text{ en el momento } T_k, \\ 0 & \text{de otra manera,} \end{cases}$$

$$0 \leq k \leq K$$

Si $Z_{jk} = 0$, la tarea actual en la unidad j en T_k continua. Como máximo una sola tarea puede comenzar en una unidad j en T_k , esto se puede escribir así:

$$Z_{jk} = \sum_{i \in I_j} Y_{ijk}, \quad 0 \leq k \leq K \quad (3)$$

Se asume que todas las tareas terminan en T_k o antes y se supone que una tarea nueva comenzará en todas las unidades en T_k , es decir, $Z_{jk} = 1$ para todas las unidades j en la formulación.

Habiendo modelado el inicio de las tareas, se procede ahora a modelar sus tamaños y tiempos. Siendo B_{ijk} ($i \in I_j, 0 \leq k$) el tamaño de lote de la tarea i que comienza en la unidad j en el momento T_k , es decir, la cantidad real de material

primario μ_i que la tarea i involucra en el arranque del lote en T_k en la unidad j . Si la tarea i no comienza en T_k , entonces $B_{ijk} = 0$, y viceversa. Por consiguiente, se tiene:

$$B_{ij}^L Y_{ijk} \leq B_{ijk} \leq B_{ij}^U Y_{ijk}, \quad i > 0, \quad (4)$$

Ahora, la formulación gira alrededor de cuatro balances involucrando unidad, tiempo e inventarios en unidades y almacenamiento.

1. Estado de una unidad de procesamiento.
2. Tiempo de procesamiento de una tarea en una unidad.
3. Cantidad de material que reside en una unidad.
4. Inventario de cada almacenamiento de material.

○ *Balance 1: estado de una unidad.*

A continuación se define las siguientes variables continuas:

$$y_{ijk} = \begin{cases} 1 & \text{si la unidad } j \text{ está continuando la realización de la tarea } i \text{ en el momento } T_k, \\ 0 & \text{de otra manera,} \end{cases}$$

$$i \in I_j, \quad 0 \leq k \leq K$$

$$YE_{ijk} = \begin{cases} 1 & \text{si la unidad } j \text{ termina la tarea } i \text{ y descarga su lote en el momento } T_k, \\ 0 & \text{de otra manera,} \end{cases}$$

$$i \in I_j, \quad 0 \leq k \leq K$$

Un balance sobre el estado de la unidad j simplemente significa:

$$y_{ijk} = y_{ij(k-1)} + Y_{ij(k-1)} - YE_{ijk}, \quad 0 < k < K \quad (5)$$

Donde y_{ijk} es cero cuando la tarea comienza, finaliza o no es asignada. y_{ijk} se hace uno únicamente después de que una tarea ha comenzado y se vuelve cero exactamente cuando la tarea termina.

Se tiene que el conjunto $y_{ijk} = 0$, asegura que ninguna tarea puede continuar más allá del último **slot** K . De manera análoga, para una planta vacía al comienzo, se tiene: $y_{ij0} = YE_{ij0} = 0$.

Es claro que la unidad j no puede comenzar una tarea nueva a menos que finalice la inmediatamente precedente. Dada la tarea ociosa, se asume que una nueva tarea siempre comienza en una unidad al final de cada tarea, esto es:

$$Z_{jk} = \sum_{i \in I_j} YE_{ijk}, \quad 0 < k < K \quad (6)$$

Note que la formulación no obliga la ecuación para $k = K$ y $k = 0$

Ahora, si una tarea i en la unidad j tiene que descargar su lote en T_k , entonces $YE_{ijk} = 1$ y se favorecería el objetivo. Similarmente, se deduce que una unidad j puede comenzar una nueva tarea, únicamente si no está continuando ninguna tarea, entonces:

$$\sum_{i \in I_j} y_{ijk} + Z_{jk} = 1, \quad 0 < k < K \quad (7)$$

Por último, una unidad puede terminar una tarea i y comenzar la misma al mismo tiempo, pero no puede continuarla y terminarla o continuarla y empezarla al mismo tiempo. Lo que se escribe así:

$$y_{ijk} + Y_{ijk} \leq 1, \quad 0 < k < K \quad (8a)$$

$$y_{ijk} + YE_{ijk} \leq 1, \quad 0 < k < K \quad (8b)$$

Las Ecuaciones (3), (5) – (8) obligan a y_{ijk} y YE_{ijk} a ser únicamente 0 o 1 (aunque ellas sean continuas 0-1), mientras que Y_{ijk} es binaria. De hecho es fácil ver que las ecuaciones 3, 6, y 7 hacen las ecuaciones (8a y 8b) redundantes. También la ecuación 5 hace una de las ecuaciones 3, 6, o 7 redundante.

o *Balance 2: tiempo de procesamiento de tarea.*

Se toma t_{ik} como el tiempo que le falta a T_k para completar la tarea que estaba en progreso durante el **slot** k en la unidad j . Al ir de T_k a T_{k+1} , este tiempo o permanecerá constante (tarea no en progreso durante el **slot** k) o disminuye (tarea en progreso) por una cantidad igual a la longitud del **slot**.

Así, un balance de tiempo en T_{k+1} da:

$$t_{j(k+1)} \geq t_{jk} + \sum_{i \in I_j} (\alpha_{ij} Y_{ijk} + \beta_{ij} B_{ijk}) - SL_{(k+1)}, \quad k < K \quad (9)$$

La desigualdad le permite a una tarea continuar en una unidad incluso después de su realización. A menos que una tarea este en progreso en el momento cero, se tiene $t_{j0} = 0$; y también, si no se permite que una unidad continúe procesando más

allá del **slot K**, también se tiene $t_{jK} = 0$. En cualquier momento, cuando una tarea completa su duración requerida en una unidad, t_{jK} tiene que ser cero.

○ *Balance 3: cantidad de lote en una unidad.*

Se tiene b_{ijk} como la cantidad de material primario μ_i (tamaño de lote) que reside en la unidad j justo antes de T_k y BE_{ijk} como la cantidad que la tarea i descarga en su finalización en T_k . Un sencillo balance de masa en torno a la unidad j da:

$$b_{ijk} = b_{ij(k-1)} + B_{ij(k-1)} - BE_{ijk}, \quad i > 0, k > 0 \quad (10)$$

Si la unidad j está vacía en el momento cero, entonces $b_{ij0} = BE_{ij0} = 0$. En T_K , se desea que todas las unidades estén vacías, por lo que $b_{ijK} = 0$. Similarmente, si en cualquier momento una unidad j no está realizando una tarea i en T_k , entonces b_{ijk} tiene que ser cero y viceversa. En otras palabras,

$$B_{ij}^L y_{ijk} \leq b_{ijk} \leq B_{ij}^U y_{ijk}, \quad i > 0, 0 < k < K^* \quad (11)$$

Además, una unidad solo puede descargar un lote en T_k si su tarea termina, es decir, $YE_{ijk} = 1$ y viceversa. En otras palabras:

$$B_{ij}^L YE_{ijk} \leq BE_{ijk} \leq B_{ij}^U YE_{ijk}, \quad i > 0, 0 < k < K^* \quad (12)$$

Finalmente, t_{jk} puede ser cero únicamente cuando y_{ijk} y b_{ijk} lo son. Por lo tanto,

$$t_{jk} \leq \sum_{i \in I_j} (\alpha_{ij} y_{ijk} + \beta_{ij} b_{ijk}), \quad 0 < k < K \quad (13)$$

Note que la ecuación anterior no viola el valor máximo posible para t_{jk} , que es el tiempo de procesamiento del lote de la tarea en progreso en la unidad j . Además, esto asegura que una tarea no termina, hasta que su duración se haya completado.

○ *Balance 4: inventario de material*

La ejecución de una tarea consumirá y producirá material. Se asume que:

* Ecuaciones corregidas de la formulación original por los autores del proyecto; aprobadas mediante comunicación directa vía e-mail con el autor de ella, I. A. Karimi.

1. Todos los materiales son acumulados en un almacenamiento (imaginario o real).
2. El tiempo para transferir material a o desde este almacenamiento es despreciable o esta incluido en el tiempo de procesamiento.
3. Una tarea a su inicio consume los materiales requeridos desde el almacenamiento.
4. A su final, una tarea transfiere los materiales hacia el almacenamiento.

Entonces, el balance de inventario para un material m en T_k es:

$$I_{mk} = I_{m(k-1)} + \sum_{i \in OI_m, i \neq 0} \sum_{j \in J_i} (\sigma_{mi} / \sigma_{\mu_i}) BE_{ijk} + \sum_{i \in II_m, i \neq 0} \sum_{j \in J_i} (\sigma_{mi} / \sigma_{\mu_i}) B_{ijk}, \quad (14)$$

Donde I_{mk} es el inventario de material m en T_k , OI_m es el conjunto de tareas que producen material m , II_m es el conjunto de tareas que consume material m , y σ_{mi} es el rendimiento del coeficiente estequiométrico de material m en el balance de material de la tarea i , el cual es negativo para los materiales consumidos por la tarea i y positivo para los producidos. La ecuación 14 asegura que una tarea nunca es realizada a menos que los materiales requeridos estén presentes en sus inventarios respectivos. Indirectamente, esto también gobierna la precedencia de tareas en las diferentes unidades.

Finalmente se imponen límites, superior e inferior, a todas las variables en la formulación, así se reducen los nodos en la solución del problema de PLEM. Por consiguiente, se establecen para SL_k , t_{jk} , I_{mk} , y otros:

$$SL_k \leq \max_j \left[\max_{i \in I_j} (\alpha_{ij} + \beta_{ij} B_{ij}^U) \right] \quad (15)$$

$$t_{jk} \leq \max_{j \in I_j} (\alpha_{ij} + \beta_{ij} B_{ij}^U) \quad (16)$$

$$I_{mk} \leq I_m^U \quad (17)$$

$$B_{ijk}, b_{ijk}, BE_{ijk} \leq B_{ij}^U \quad (18)$$

Donde I_m^U es la capacidad de almacenaje máxima para material m . En adición, todas las variables continuas son no negativas,

$$Z_{jk}, y_{ijk}, YE_{ijk}, SL_k, t_{jk}, B_{ijk}, b_{ijk}, BE_{ijk}, I_{mk} \geq 0. \quad (19)$$

A parte de los límites (ecuaciones 15-19), las ecuaciones 2-6 y 9-14 son todas las restricciones necesarias para nuestro problema de producción.

o *Función objetivo*

Al asumir que una planta puede vender todos los productos que ella produce, la utilidad neta o beneficio de vender los inventarios de producto terminado al final del horizonte es

$$P = \sum_m g_m I_{mk} \quad (20)$$

Donde g_m es la utilidad neta o beneficio por kg o um (unidad de masa) de producto. Para este objetivo de maximización de ventas o beneficio, la formulación comprende las ecuaciones 2-6, 9-14, 20 y las ecuaciones de límites de 15-19.

Para la *minimización del makespan*, la formulación se modifica ligeramente.

En este caso H deja de ser un parámetro dado. La planta debe satisfacer algunas demandas mínimas de productos. Si D_m denota la demanda para el material m , entonces se tiene:

$$I_{mK} \geq D_m \quad (21)$$

En este caso la ecuación (2) no es necesaria. De esta manera, el **makespan** se minimiza así:

$$MS = \sum_{k=1}^{NK} SL_k \quad (22)$$

Ahora, para este objetivo las ecuaciones necesarias son: 3-6, 9-14, 21, 22 y las de límites 15-19.

- **Ventajas.** Esta formulación tiene características notables que la hacen distinta y de cierta manera, superior a las anteriores. Sus autores la evalúan mediante comparación con las dos formulaciones desarrolladas por: Giannelos y Georgiadis (2002), quienes en su momento concluyeron que su trabajo tenía ventajas significativas respecto al tamaño del modelo, a las soluciones y al tiempo de procesamiento computacional, con respecto a los trabajos anteriores y con la desarrollada por Maravelias y Grossmann (2003a), quienes también encontraron su formulación ventajosa, comparando su modelo con los existentes en la literatura excepto con el de Giannelos y Georgiadis (2002).

Con el fin de reducir el número total de variables binarias, estas dos formulaciones disocian tareas de unidades, reemplazando la variable binaria de asignación y , de tres índices (tareas i , unidades j y **slots** k) por dos variables binarias cada una con

dos índices: w (tareas - punto evento o **slot**) y v (unidades - punto evento o **slot**). Los autores señalan que esto no garantiza un tiempo de solución mejorado y prueban que el número de variables binarias es el mismo por ambos métodos, ya que de otra manera la formulación no llegaría a soluciones óptimas. La única diferencia es que en el método de disociación las w -variables de dos índices despliegan explícitamente la información unitaria en términos de j , mientras que después oculta la misma detrás de i ; lo que quiere decir que la separación de tareas de unidades incrementa el número de binarias para incrementar el número de tareas y al mismo tiempo las disminuye por eliminación de las v -variables, así el efecto de red de aquellas dos acciones es cero en el número total de variables binarias.

Por otro lado, en contraste con los modelos de tiempos continuos existentes de Ierapetritou y Floudas (1998), Giannelos y Georgiadis (2002) y Maravelias y Grossmann (2003), esta formulación no utiliza restricciones gran-M; los autores creen que esto es significativo, porque la experiencia muestra que la eliminación de las restricciones gran-M generalmente mejora las formulaciones de PLEM. En general, las formulaciones en la literatura basadas en eventos, han utilizado restricciones gran-M en abundancia. Maravelias y Grossmann (2003) utilizaron también numerosas restricciones gran-M en el modelamiento de restricciones de duración, tiempo final y en los ajustes de tiempo. Además creen que la carencia de restricciones gran-M da al modelo un avance computacional, como lo muestran en la evaluación realizada.

Por otro lado afirman que el modelo es mucho más sencillo que los anteriores. Como lo demostraron en la evaluación realizada mediante la comparación de las formulaciones mencionadas, este modelo tiene notablemente menos restricciones y elementos diferentes a cero (**nonzeros**) y también utiliza menos variables binarias. Por ejemplo, Maravelias y Grossmann (2003) utilizaron variables binarias para el inicio, final y continuación de tareas, mientras que los autores solo las emplean para el inicio de tareas.

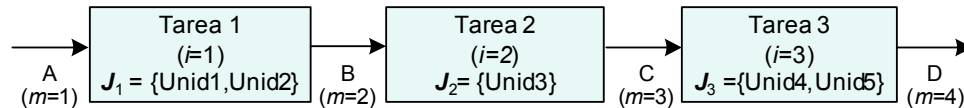
Para una comparación justa los autores (S&K) manipularon su modelo, el de Maravelias y Grossmann 2003 (M&G) y Giannelos y Georgiadis 2002 (G&G): mediante el empleo de GAMS, solucionando numerosos ejemplos existentes en la literatura y otros de invención propia utilizando en los tres modelos un DELL GX 270 (Pentium IV 2.8 GHz CPU con 1GB de RAM) corriendo Cplex 8.1.0 en GAMS 21.2. Evaluaron los tres modelos para ambos objetivos de programación para obtener una mejor idea de cual modelo es fundamentalmente el mejor. Además, resolvieron cada ejemplo para diferentes escenarios y así de manera empírica demostraron la veracidad de su formulación.

- *Evaluación.* Dos ejemplos son expuestos y evaluados para ambos objetivos de programación; el primero (ejemplo 1a en la formulación) es un ejemplo sencillo

de un proceso lineal que solo involucra tres tareas, cinco unidades, cuatro materiales, un horizonte de programación de 8 horas para maximizar beneficio y una demanda de 2000 um (unidades de masa) para un solo producto en la minimización de **makespan**.

La siguiente figura muestra el diagrama RD para este ejemplo.

Figura 10. Diagrama de receta para el ejemplo 1a



Arul Sundaramoorthy y I. A. Karimi

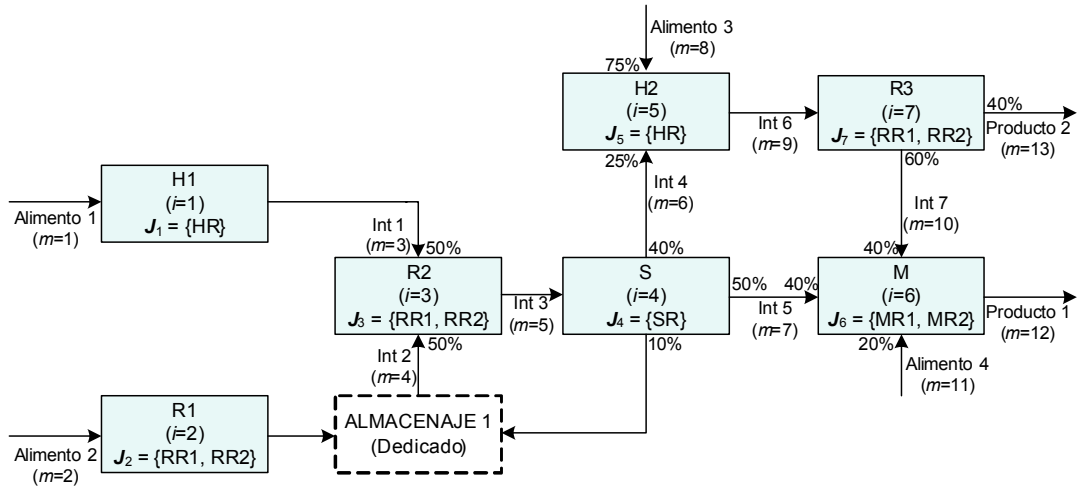
En el segundo ejemplo se consideran dos escenarios (ejemplos 3a y 3b en la formulación). Este es el ejemplo más complejo desarrollado por los autores porque involucra el mayor número de unidades ($J=6$), de tareas ($I=7$) y de materiales ($M=13$). En el primer escenario se toma, un horizonte de programación de 8 horas al maximizar beneficio y una demanda de 100 y 200 um para dos productos respectivamente al minimizar **makespan**. En el segundo escenario se toma, un horizonte de programación de 12 horas para maximizar beneficio y una demanda de 250 um para dos productos al minimizar **makespan**. Aquí, se consideran las características más comunes de una planta multipropósito a saber:

1. Una unidad puede trabajar tareas múltiples.
2. Una tarea puede ser trabajada en unidades múltiples.
3. Numerosas tareas aptas para un conjunto de unidades.
4. Únicamente una tarea apta para una unidad.

Además, se asumen inventarios iniciales para ciertos materiales intermedios con $I_{6,0} = I_{7,0} = 50$ um. Se requiere una tarea de almacenamiento para imitar la mezcla de material (Int 2) reciclado de la tarea S y producido por la tarea R1. Además se considera que los tiempos de transferencia hacia y desde esta tarea de almacenamiento son insignificantes.

La siguiente figura muestra el diagrama RD para este ejemplo.

Figura 11. Diagrama de receta para el ejemplo 3a



Arul Sundaramoorthy y I. A. Karimi

El cuadro 2 y 3 muestran la información referente a tareas, unidades y materiales.

Cuadro 2. Límites sobre los tamaños de lote de tareas y coeficientes en las expresiones para tiempos de procesamiento.

TAREA	i	Unidad	j	τ_{ij}	α_{ij}	β_{ij}	B^L_{ij}	B^U_{ij}
Ejemplo 1a								
Tarea 1	1	Unidad 1	Und1	2	1.333	0.01333	---	100
		Unidad 2	Und2	2	1.333	0.01333	---	150
Tarea 2	2	Unidad 3	Und3	1.5	1.000	0.00050	---	200
Tarea 3	3	Unidad 4	Und4	1	0.667	0.00455	---	150
		Unidad 5	Und5	1	0.667	0.00455	---	150
Ejemplo 3a y 3b								
Calentar-1	H1	Calentador	HR	1	0.667	0.00667	---	100
Calentar-2	H2	Calentador	HR	1.5	1.000	0.01000	---	100
Reacción-1	R1	Reactor 1	RR1	2	1.333	0.01333	---	100
		Reactor 2	RR2	2	1.333	0.00889	---	150
Reacción-2	R2	Reactor 1	RR1	1	0.667	0.00667	---	100
		Reactor 2	RR2	1	0.667	1.00445	---	150
Reacción-3	R3	Reactor 1	RR1	2	1.333	0.01330	---	100
		Reactor 2	RR2	2	1.333	0.00889	---	150
Separar	S	Separador	SR	3	2.000	0.00667	---	300
Mezclar	M	Mezclador 1	MR1	2	1.333	0.00667	20	200
		Mezclador 2	MR2	2	1.333	0.00667	20	200

Arul Sundaramoorthy y I. A. Karimi

Cuadro 3. Capacidades de almacenamiento, inventarios iniciales y precios de venta de materiales.

Material m	Capacidad de almacenamiento (um)	Inventario Inicial (um)	Precio de venta (\$/um)
Ejemplo 1a			
1	IL	DP	0
2	200	0	0
3	250	0	0
4	IL	0	5
Ejemplo 3a y 3b			
1	IL	DP	0
2	IL	DP	0
3	100	0	0
4	100	0	0
5	300	0	0
6	150	50	0
7	150	50	0
8	IL	DP	0
9	150	0	0
10	150	0	0
11	IL	DP	0
12	IL	0	5
13	IL	0	5

IL = ilimitada; DP = disponible como y cuando es requerida

Arul Sundaramoorthy y I. A. Karimi

La programación y solución del ejemplo 3a en su primer escenario para maximizar beneficio y el ejemplo 3b, segundo escenario para minimizar **makespan** mediante el uso de la herramienta GAMS/CPLEX puede notarse en los anexos mencionados en cada caso. El anexo se divide en tres partes a saber: La primera corresponde al código de programación, la segunda al procesamiento iterativo y solución y la tercera a las estadísticas del modelo.

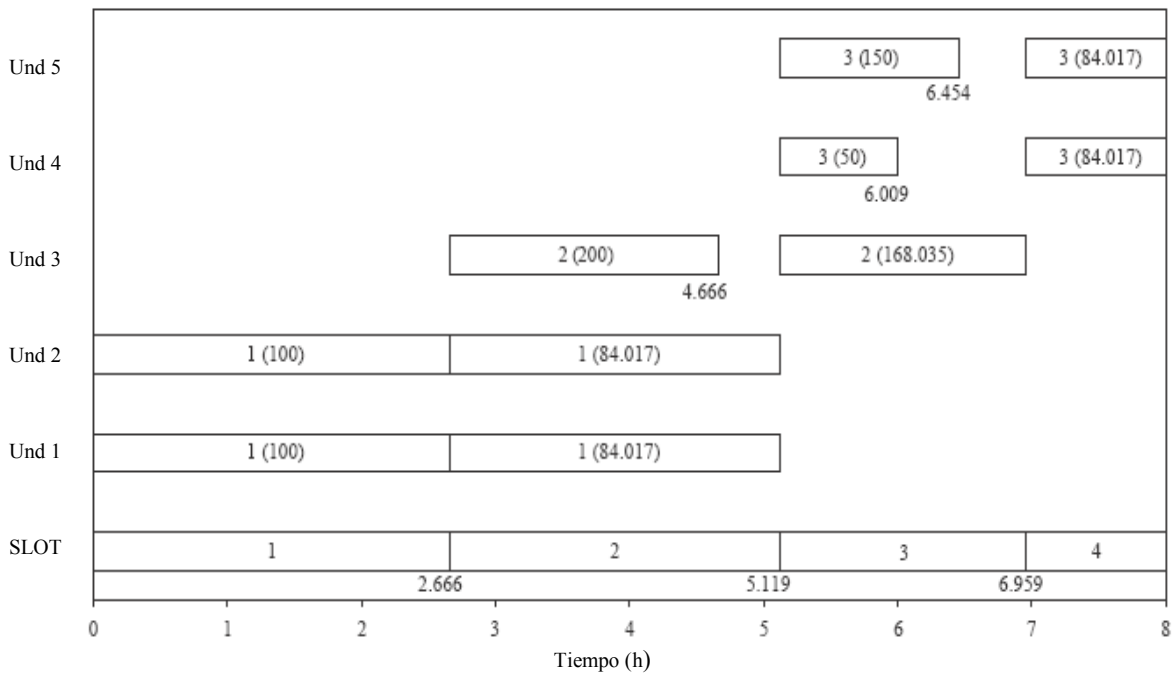
Si es de interés para el lector analizar todos los ejemplos con sus diferentes escenarios y objetivos de programación, diríjase al Anexo A, donde cada ejemplo se encuentra relacionado y listo para su ejecución. Los diferentes escenarios se hallan realizando simples cambios en la programación.

1. Maximización de beneficio.

a. Ejemplo 1a: se considera un horizonte de programación de 8 horas. Para definir K , se incrementa desde 4 a 5 y se observa que ninguna mejora es notoria en el objetivo de la PLEM, así, se asume trabajar con $K=4$ para la

solución óptima. La figura 12 muestra la programación óptima que arroja el modelo. Un rectángulo denota un **slot**. Existen tareas que requieren más de un **slot** para su finalización, en este caso los **slots** se unen para formar un solo rectángulo que denota la duración completa de la tarea. La etiqueta dentro de cada rectángulo representa la tarea que la unidad realiza y el número dentro del paréntesis denota la cantidad de lote (um) de esta tarea. Más aún, si una tarea finaliza antes, entonces su último **slot** termina cuando se muestra su fin exacto bajo el rectángulo.

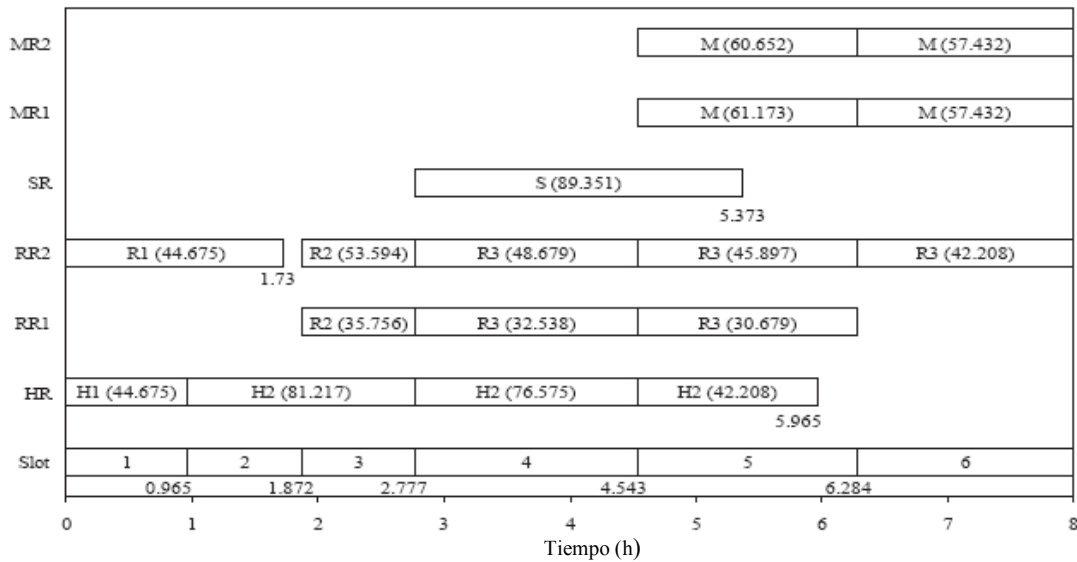
Figura 12. Programación de máximo beneficio para el ejemplo 1a tomando $H=8$, $K=4$ y tiempos variable de procesamiento de lote.



Arul Sundaramoorthy y I. A. Karimi

b. Ejemplo 3a y 3b: se considera un horizonte de programación de 8 horas. Al igual que en el ejemplo anterior, se prueba con $K=5, 6$ y 7 ; y se observa que el objetivo no mejora desde $K=6$ hasta 7 . Sin embargo, como observó CASTRO (2001), el mismo valor objetivo para dos K 's sucesivas no significa optimalidad. Por lo tanto, se prueba para $K=8$ y 9 ; y se nota que el valor objetivo no mejora, así, se toma con seguridad la programación con $K=6$ como la programación de máximo beneficio. La figura 13 representa la solución a este ejemplo mediante el diagrama de Gantt.

Figura 13. Programación de máximo beneficio para el ejemplo 3a tomando H=8, K=6 y tiempos de procesamiento variable.



Arul Sundaramoorthy y I. A. Karimi

El cuadro 4 muestra las estadísticas de solución para los ejemplos expuestos anteriormente y los Anexos A/Arul y Karimi_eje1a.gms, Anexo B y Anexo A/Arul y Karimi_eje3b.gms, muestran los códigos de programación y la solución en GAMS, de los mismos.

En el primer ejemplo se observa que los valores objetivos PLEMR y PLEM son idénticos para los tres modelos y el tiempo de solución es también casi el mismo. Estas observaciones no son extrañas al tratarse de un ejemplo tan sencillo y trivial del que nada se puede concluir. Por esto, para verificar la eficacia y superioridad computacional de esta formulación debe tratarse con ejemplos de mayor complejidad porque en este caso (ejemplo 1a) los tres modelos se desempeñan igualmente bien.

Para el ejemplo 3a cuando $K=7$, el modelo S&K (formulación aplicada) es casi dos veces más rápido que el M&G (32974s vs. 67157s). El modelo G&G da una solución subóptima.

En el segundo escenario (ejemplo 3b), se solucionaron los tres modelos para $H=12h$ y se utilizó $K=7$ para el modelo de S&K y $N=8$ y 7 para los modelos M&G Y G&G, respectivamente. El modelo de estudio es casi tres veces más rápido (140s vs. 462s) que el modelo M&G. Otra vez, el modelo G&G da valores pobres en PLEMR Y PLEM. Con este ejemplo se puede ver que el modelo maneja los problemas más complejos mejor que los otros modelos de la literatura.

Cuadro 4. Soluciones estadísticas del ejemplo 3 para maximizar beneficios y utilizando tiempos de procesamiento de lotes variables.

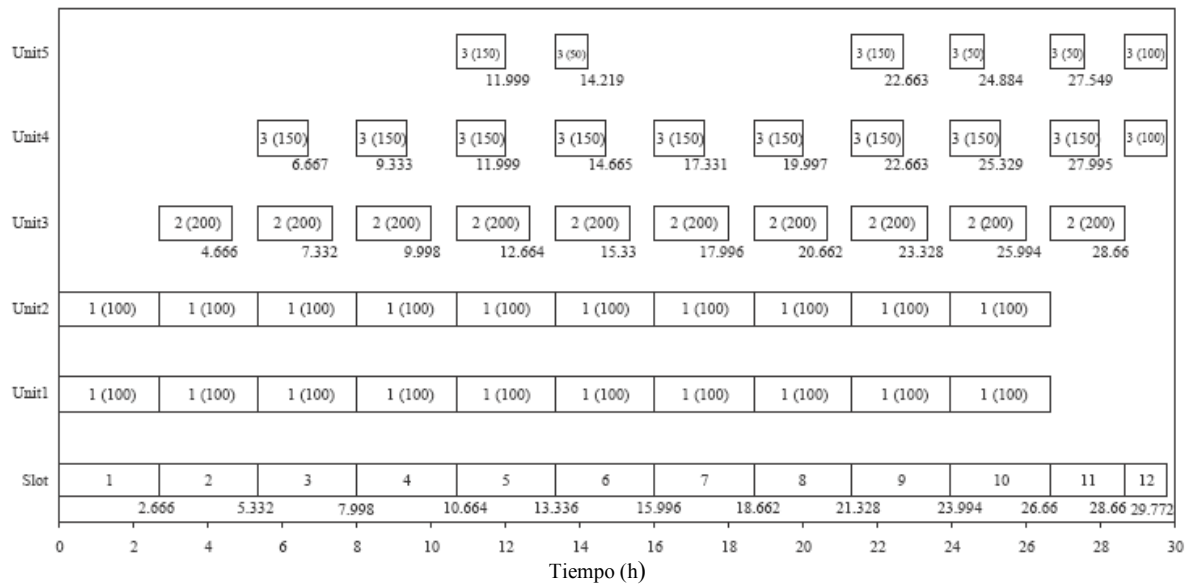
Modelo	K/N	Tiempo CPU	Nodos	PLEMR (\$)	PLEM (\$)	Variables binarias	Variables Continuas	Restricciones	No ceros
E. 1a (H = 8)									
S&K	4	0.06	12	2000.0	1840.2	40	216	192	643
M&G	5	0.09	1	2000.0	1840.2	50	221	613	1798
G&G	4	0.08	0	2000.0	1840.2	20	81	141	393
E. 3a (H = 8)									
S&K	5	37.2	49765	2100.0	1283.1	85	502	597	1816
M&G	6	84.9	71616	2100.0	1283.1	132	691	1627	5432
G&G	4	0.05	0	1571.9	1150.0	44	198	376	1183
E. 3b (H = 12)									
S&K	7	140	42013	3464.0	2867.2	119	688	859	2598
M&G	8	462	75195	3464.0	2867.2	176	921	2159	7909
G&G	7	28.6	298927	3465.6	2443.2	77	336	643	2098

Arul Sundaramoorthy y I. A. Karimi

2. Minimización de *makespan*.

a. Ejemplo 1a: se asume una demanda para el material final de 2000 um. Como se espera, para este caso trivial los tres modelos también se desenvuelven bien de igual manera. La figura 14 presenta la solución óptima a este ejemplo.

Figura 14. Programación de mínimo *makespan* para el ejemplo 1a con K=12



Arul Sundaramoorthy y I. A. Karimi

b. Ejemplo 3b: se toma una demanda para el material 12 y 13 $D_{12} = D_{13} = 250$ um.

El cuadro 5 muestra las soluciones estadísticas de los ejemplos expuestos anteriormente y los Anexos A/Arul y Karimi_eje1a_MS.gms y Anexo C, muestran los códigos de programación y la solución en GAMS, de los mismos.

Cuadro 5. Soluciones estadísticas del ejemplo 1a y 3b para minimizar makespan, utilizando tiempos de procesamiento de lotes variables.

Modelo	K/N	H	Tiempo CPU	Nodos	PLEMR (\$)	PLEM (\$)	Variables binarias	Variables Continuas	Restricciones	No ceros
E. 1a (H = 8)										
S&K	12	---	1.00	217	27.126	29.772	120	616	627	2078
M&G	13	50	4.94	1237	27.126	29.772	130	574	1569	6641
G&G	12	50	0.08	0	27.126	29.772	60	234	409	1202
E. 3b (H = 8)										
S&K	10	---	3042	337408	15.205	17.705	170	967	1264	3782
M&G	11	100	5000	316800	15.205	17.786	242	1267	2970	12427
G&G	10	100	27.2	30428	12.871	19.842	110	475	923	3036

Arul Sundaramoorthy y I. A. Karimi

Se establece un criterio de terminación del 3% **gap** y 5000 s para los tres modelos uniformemente. El criterio **gap** refiere a que si en cierto nodo se encuentra la mejor solución hasta el momento y en un nodo sucesor, otra solución no difiere a la anterior por más del porcentaje establecido en el criterio de terminación, la compilación finaliza tomando como optima la primera de estas dos. Para el ejemplo 3b, nuevamente el modelo S&K se desarrolla más rápido (136s para 2.69% **gap**) que los otros dos modelos. El modelo M&G muestra un **gap** de 2.92% después de 799 s, mientras que el modelo G&G se desarrolla de manera más pobre (5000 s para un **gap** de 24.16%). Para terminar, el objetivo PLEMR es también inferior a los otros dos modelos.

4.3 FUNDAMENTACIÓN TEÓRICA EN MÉTODOS DE SOLUCIÓN.

Linare^{#####} dice que para el desarrollo de modelos de optimización actualmente se conocen tres tipos de lenguajes de modelado a saber:

1. *Lenguajes de programación de propósito general* (C, C++, Java, Visual Basic, FORTAN 90) que llaman a una biblioteca de optimización.
2. *Lenguajes o entorno de cálculo numérico o simbólico* (hojas de cálculo, lenguajes para cálculo numérico intensivo, como MATLAB, o para cálculo

LINARES, Op. cit.

simbólico como **Maple** o **Mathematica**, etc.).

3. *Lenguajes algebraicos de modelado.*

4.3.1 Lenguajes algebraicos de modelado. Acerca del tema Pedro Linares et al., dan la siguiente definiciones y características:

Los lenguajes algebraicos de modelado son lenguajes de alto nivel que han sido diseñados específicamente para el desarrollo e implementación de modelos de optimización de forma más directa para los programadores y más inteligible para los usuarios. En consecuencia, el campo de actuación y utilidad de los modelos de optimización se ha ampliado tremendamente al utilizar estos lenguajes.

Entre los lenguajes de modelado más conocidos se puede mencionar: GAMS y AMPL de origen estadounidense y AIMMS y XPRESS-MP de origen europeo, entre otros. GAMS es el lenguaje más ampliamente difundido comercialmente mientras que AMPL se está potenciando mucho en las universidades estadounidenses.

Entre sus *características* y *ventajas* principales se destacan las siguientes:

1. Proporciona una formulación sencilla de modelos grandes y complejos.
2. Facilitan sobremanera el desarrollo de prototipos.
3. Mejoran sustancialmente la productividad de los modeladores al permitir dedicar más tiempo al diseño, ejecución del modelo y análisis de los resultados y menos a la codificación del mismo.
4. Estructuran los buenos hábitos de modelado al exigir una representación concisa y exacta de los parámetros/variables y sus relaciones.
5. Recogen simultáneamente la estructura del modelo y su documentación.
6. Separan de manera natural los datos de la estructura del modelo y ésta de los algoritmos de solución.
7. La formulación del problema es independiente del tamaño. Permiten el uso del modelo para diferentes casos.

8. Los optimizadores pueden ser intercambiados sin dificultad, se pueden probar nuevos optimizadores, nuevos métodos o nuevas versiones.

9. Permiten la realización de cambios en el modelo de manera sencilla y segura, es decir, se puede afrontar un refinamiento continuo en la formulación del problema.

10. Para cualquier tipo de problemas de programación lineal, no lineal, flujos en redes o mixta complementaria resulta muy fácil implantar su formulación.

11. Permiten la implantación de algoritmos avanzados, que incluyan varias llamadas al optimizador o procedimientos específicos para el problema (como por ejemplo los métodos de descomposición).

12. Permiten la portabilidad de los modelos entre plataformas y sistemas operativos.

Como *desventajas* principales se pueden mencionar las siguientes:

1. No son adecuados para la resolución de problemas de tamaño pequeño por parte de usuarios esporádicos por la barrera de entrada que supone el aprendizaje de un nuevo lenguaje.

2. No pueden utilizarse para la resolución directa de problemas gigantescos cuya formulación completa incluso no se puede realizar (por ejemplo, a partir de un millón de restricciones y/o variables).

3. En la ejecución se incluye un tiempo de creación del modelo y de interfaz con el optimizador que retarda la obtención de la solución, por lo tanto no es recomendable cuando el tiempo de ejecución es un factor crítico.

Los campos de acción de estos lenguajes son tan amplios como los de optimización propiamente dicha.

- *Modelado en GAMS.* El *sistema de modelado algebraico general* (GAMS), puede dar solución a problemas matemáticos de programación a través de herramientas dispuestas para tal fin y para este proyecto en particular, se utilizará el solucionador CPLEX, diseñado para resolver problemas de gran tamaño, alta dificultad, rápidamente y con la intervención mínima del usuario.

GAMS fue uno de los primeros sistemas de modelamiento desarrollado en los años 80's, como respuesta al desarrollo rápido de algoritmos y códigos

computarizados, para analizar y resolver problemas grandes de programación matemática.

El diseño de GAMS ha incorporado ideas de la teoría de base de datos relacionada y programación matemática, intentado combinarlas para satisfacer las necesidades y facilidades para el modelamiento. La teoría de base de datos relacionada proporciona un marco estructurado para desarrollar la organización general de datos y capacidades de transformación. La programación matemática proporciona un modo de describir un problema y una variedad de métodos para solucionarlo. De esta manera se dice que GAMS está diseñado para:

1. Proporcionar un lenguaje algebraico de alto nivel para la representación compacta de modelos grandes y complejos.
2. Permitir hacer cambios en las especificaciones del modelo, de manera sencilla y segura.
3. Permitir relaciones algebraicas por medio de declaraciones no ambiguas.
4. Proporcionar un ambiente donde el desarrollo del modelo es facilitado por el subíndice fundamentado en la expansibilidad y así permitir al modelador empezar con un conjunto pequeño de datos y luego verificar su extensión correcta hacia un contexto mucho más amplio.
5. Estar autodocumentado, permitiendo el uso de muchas más variables, ecuaciones e índices, así también como comentarios, definición de datos, etc. GAMS está diseñado entonces para que la estructura del modelo, asunciones y cualquier procedimiento de cálculo usado en el reporte escrito sea documentado como un subproducto del ejercicio modelado en un archivo autocontenido.
6. Ser un sistema abierto que facilite la interfaz con los más nuevos y mejores solucionadores. Mientras es un solucionador independiente, permite que diferentes solucionadores sean utilizados en un problema dado.
7. Automatizar el proceso de modelamiento incluyendo:
 - a. Permiso del cálculo de datos;
 - b. Verificación de la exactitud de la declaración algebraica del modelo;
 - c. Revisión de la formulación para las fallas obvias;
 - d. Conexión con un solucionador
 - e. Permiso del uso de la solución para el reporte escrito.

8. Permitir la portabilidad de la formulación de un modelo entre los sistemas del computador habilitando el uso de una variedad de computadores que van de PCs a estaciones de trabajo a super computadores.

9. Cambiar de solucionadores de manera sencilla, mediante una frase que indique la modificación o reemplazando el uso de PL por el uso de PNL.

10. Facilitar el importe y exporte de datos hacia y desde otros paquetes del computador.

11. Permitir el uso por grupos de especializaciones variadas.

12. Suministrar modelos ejemplo que pueden ayudar a los modeladores a través de la provisión de una biblioteca de modelo.

GAMS tiene disponible los solucionadores CPLEX, OSL, XA y XPRESS para problemas PL y PEM, MINOS y CONOPT para problemas PNL, DICOPT para problemas PNLEM y MILES y PATH para problemas PCM (problemas complementarios mixtos).

Como es visto y ya se ha mencionado, nuestro problema será solucionado mediante la combinación de las capacidades de alto nivel del modelado en GAMS y el poder de optimización de CPLEX.

- **CPLEX.** Es uno de los solucionadores de GAMS que (sujetos a licencia) da algoritmos de solución a problemas lineal, restringido cuadráticamente y lineal entero mixto. Mientras numerosas opciones de solución están disponibles, GAMS/CPLEX automáticamente calcula y agrupa la mayoría de las opciones en los mejores valores, para problemas específicos.

Antes de la frase de solución, debe indicarse la siguiente frase para que GAMS haga uso exclusivo de ese solucionador:

Option LP = Cplex; {o QCP, MIP, MIQCP, RMIP o RMIQCP}

Si CPLEX fue especificado como solucionador por defecto, entonces la frase anterior no es necesaria.

Los métodos utilizados para resolver problemas de programación entera pura y programación entera mixta requieren dramáticamente más cómputo matemático que aquellos programas lineales puros de tamaño similar. Muchos modelos de programación entera relativamente pequeños toman

enormes cantidades de tiempo para ser resueltos.

Para problemas con variables enteras, CPLEX usa un algoritmo **branch and cut** (B&C) el cual resuelve una serie de subproblemas de PL. Porque un solo problema entero mixto genera muchos subproblemas, incluso los problemas entero mixtos pequeños pueden ser de computación muy intensiva y requieren grandes cantidades de memoria física.

GAMS y GAMS/CPLEX apoyan los conjuntos de orden especial de tipo 1 y tipo 2 así como las variables semi-continuas y semi-enteras.

CPLEX también puede resolver problemas de GAMS de tipo MIQCP (problemas restringidos cuadráticamente enteros mixtos). Como en el caso continuo, si el modelo base es un QP el método simplex puede ser usado y los duales estarán disponibles en la solución. Si el modelo base es un QCP, únicamente el método de barrera puede usarse para los nodos y sólo los valores primales estarán disponibles en la solución^{§§§§§§§§}.

- **Algoritmo branch and cut (rama y corte).** Mitchell^{*****} dice que el algoritmo **branch and cut** es un método de optimización combinatoria para resolver programas lineales enteros, es decir, problemas de programación lineal donde algunos o todos los valores son desconocidos y restringidos a ser enteros. El método es un híbrido entre los métodos **branch and bound** (rama y atadura) y **cutting plane** (hiperplanos cortantes). Estos métodos trabajan para la solución de una secuencia de relajaciones de programación lineal del problema de programación entera. Los métodos **cutting plane** mejoran la relajación del problema para acercarse más al problema de programación entera y algoritmos **branch and bound** continúan con una división sofisticada hasta lograr dicho acercamiento para solucionar problemas.

El método soluciona el problema lineal sin la restricción de número entero que utiliza el algoritmo simplex. Cuando se obtiene una solución óptima y esta solución tiene un valor no entero para una variable que se supone entera, un algoritmo **cutting plane** es usado para encontrar las restricciones lineales que satisfacen a todos los puntos enteros factibles, pero que en el momento están siendo violadas por la solución corriente fraccionaria. Si tal desigualdad es encontrada, es añadida al programa lineal tal que la resolución de ello producirá una solución diferente que se espera sea *menos fraccionaria*. Este proceso es repetido hasta que una solución entera es encontrada (la cual es candidata para ser la óptima) o hasta que no más hiperplanos cortantes son encontrados.

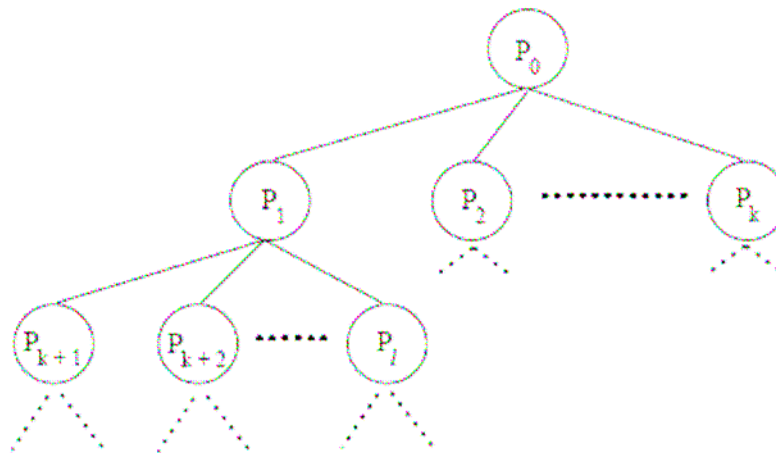
§§§§§§§§ Ibid.

***** MITCHELL, John E. Branch and cut algorithms for combinatorial optimization problems. [en línea] Disponible en <www.rpi.edu/~mitchj/papers/bc_hao.ps> [citado 27 agosto de 2006].

En este momento, la parte del algoritmo **branch and bound** es iniciada. El problema es dividido en dos versiones, una con la restricción adicional de que la variable es mayor que o igual al siguiente número entero mayor del resultado intermedio y otra donde esta variable es menor que o igual al siguiente número entero menor. De este modo nuevas variables son presentadas en la base según el número de variables básicas que son no enteras en la solución intermedia, pero que son números enteros de acuerdo a las restricciones originales. Los programas lineales nuevos entonces son resueltos usando el método simplex y el proceso se repite hasta que una solución que satisfaga todas las restricciones de enteros sea encontrada.

Lucena⁺⁺⁺⁺⁺⁺ representa el proceso del algoritmo **branch and cut** como sigue en la figura y de ello dice:

Figura 15. Ilustración del proceso **branch and cut** en términos de un árbol



Abilio Lucena y John E. Beasley

Un nodo en el árbol de la figura 15, se denota como k . La raíz del árbol es el nodo $k = 0$, tan pronto como se avanza hacia abajo en el árbol de búsqueda las regiones P_k comienzan a ser cada vez más pequeñas. A través del procedimiento algorítmico, puede notarse que la solución óptima puede estar o no contenida en un P_k dado. Para esto es más conveniente trabajar con las relajaciones del problema de programación dado que con el original.

++++++ LUCENA, Abilio y BEASLEY John E. Branch and cut algorithms. [en línea], p. 188. Disponible en <people.brunel.ac.uk/~mastjbb/jeb/book/chapter5.pdf> [citado 27 agosto de 2006]

5. AMBIENTE DE DESARROLLO DEL PROYECTO

5.1 ITALCOL S.C.A.

ITALCOL^{#####} sostiene que:

Fue constituida en 1970 como una empresa dedicada a la fabricación, distribución y venta de alimentos concentrados para animales. Cuenta con plantas de producción en Bogotá, Cali, Bucaramanga, Barranquilla y Medellín, en Colombia y en Panamá.

Los orígenes de sus importaciones son principalmente:

- Maíz - Harina de soya - Fríjol Soya: Estados Unidos, Argentina y Paraguay.
- Harina de pescado: Ecuador y Perú
- Vitaminas y Aditivos: Unión Europea y China

Estas importaciones entran al país a través de los puertos de Buenaventura y Barranquilla. Su capacidad de producción de alimentos concentrados para animales a nivel nacional, es de más de 50.000 toneladas mensuales. Sus ventas totales anuales de alimentos concentrados y materias primas ascienden a 900.000 toneladas anuales. Sus líneas básicas son: las harinas, los peletizados y los extruídos.

Dentro de las líneas de producto que esta empresa ofrece, se encuentran: avicultura, porcicultura, acuicultura, ganadería, equinos, mascotas, y materias primas y aditivos.

5.2 ORGANIZACIÓN

ITALCOL^{#####} presenta las siguientes generalidades:

5.2.1 Misión.

ITALCOL existe para satisfacer la necesidad de alimentación, en busca permanente de una mejor nutrición, manteniendo y desarrollando una oferta

ITALCOL S.A. Historia. [en línea] Disponible en <<http://www.italcol.com/corporativa.htm>> [citado 27 de diciembre de 2006].

Ibid.

de productos, social, económica y ambientalmente sostenibles, procurando el mejoramiento continuo del nivel de vida de nuestra comunidad.

5.2.2 Objetivo.

La más alta calidad.

La calidad es nuestro compromiso con la satisfacción de todos los clientes. Un compromiso que se demuestra en la investigación y desarrollo continuo de nuevos productos. Cumplimos con todas las especificaciones requeridas para la producción de alimentos concentrados, además de óptimas condiciones de precio y servicio. Con responsabilidad, respeto y profesionalismo continuaremos liderando el mercado de alimentos concentrados en Colombia.

5.2.3 Política de calidad.

En ITALCOL entendemos la calidad como el compromiso de todo el grupo humano con la satisfacción del cliente. Este compromiso se demuestra en la investigación y mejoramiento continuo de nuestros procesos.

Es así como atendemos las necesidades de nuestros clientes; produciendo y suministrando alimentos concentrados para animales, según las especificaciones requeridas, las cuales tienen óptimas condiciones de precio y servicio que permiten alcanzar los resultados zootécnicos para cada especie.

Esperamos consolidar nuestro liderazgo en el mercado y con responsabilidad, respeto y profesionalismo, constituirnos como el mejor aliado de nuestros clientes, haciendo de nuestro nombre un sinónimo de calidad, cumplimiento y confianza.

PARTE II METODOLOGÍA PROPUESTA

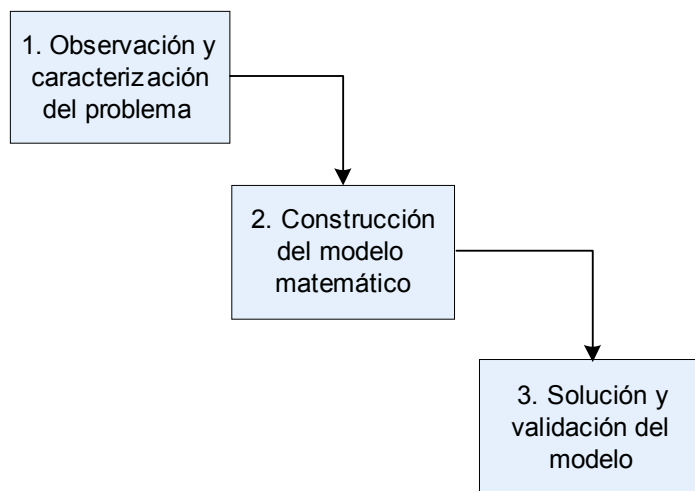
6. METODOLOGÍA

Por la naturaleza de este trabajo de grado, la metodología o método a emplear debe generalizar el proceso de diseño, elaboración y validación de un modelo matemático para programar la producción de ITALCOL S.C.A. Bucaramanga, así mismo, debe ser una herramienta sencilla que permita poner en marcha procedimientos de observación adecuados, haciendo posible una correcta recopilación de datos y mediante análisis se pueda identificar, definir y formular el problema claramente hasta llegar a soluciones y conclusiones que se ajusten a lo esperado.

Por las consideraciones expuestas anteriormente y el estudio de las formulaciones, la metodología más apropiada para aplicar en este trabajo es el método científico, ya que visto como un método de respuesta sigue unos pasos sencillos en los que primero se observa la situación, se formula una pregunta que se convierte en una hipótesis demostrable, se predice el resultado del experimento, se desarrolla el experimento, se analizan los resultados y finalmente se evalúa la hipótesis.

La figura 16 muestra las etapas metodológicas a desarrollar en este proyecto.

Figura 16. Etapas metodológicas del proyecto



Los autores

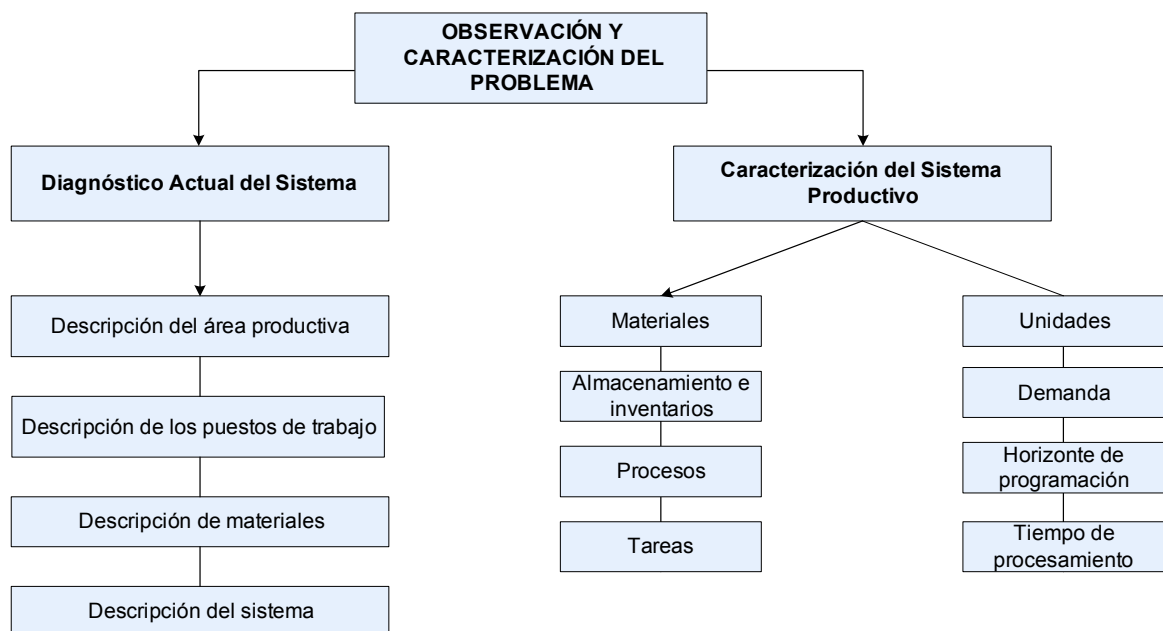
6.1 ETAPA 1: OBSERVACIÓN Y CARACTERIZACIÓN DEL PROBLEMA

En esta etapa inicial se realiza la observación y análisis del proceso productivo de la empresa, desarrollando un resumen bien claro del problema que se va a analizar. El proceso requiere anotación de lo observable, ordenamiento, tabulación y selección de los datos obtenidos. Este minucioso análisis lleva a establecer diferencias entre la situación actual y la situación objetivo o deseada, como también permite la determinación de los objetivos específicos apropiados, las restricciones sobre lo que se va a hacer, las variables involucradas, los diferentes cursos de acción posibles, los límites de tiempo para tomar una decisión, etc.

Seguidamente el problema se considera así: a) identificación de los componentes y variables controlables y no controlables del sistema; b) identificación de los posibles cursos de acción, determinados por las componentes controlables; c) definición del marco de referencia dado por las componentes no controlables; d) identificación de las exigencias importantes entre las diferentes partes del sistema y así encontrar las restricciones que existen.

Para facilitar el cumplimiento de estos propósitos, los esfuerzos de desarrollo se realizan mediante una serie de fases que tienen metas definidas, bajo un enfoque sistemático-secuencial. La siguiente figura muestra las fases de esta etapa.

Figura 17. Fases de la observación y caracterización del problema



Los autores

6.1.1 Diagnóstico actual del sistema. Esta fase comprende la observación global y análisis del sistema productivo mediante el desarrollo de diagramas, tablas de tabulación y descripciones escritas que faciliten el entendimiento del sistema y el reconocimiento de todos sus componentes.

- *Descripción del área productiva.* Consta de la identificación física, descripción y ubicación de las áreas que involucra el proceso productivo, de manera que se reconozca el espacio dispuesto dentro de la planta.
- *Descripción de los puestos de trabajo.* Se lista cada puesto de trabajo y se mencionan sus principales características, teniendo en cuenta los parámetros que de cada uno son necesarios.
- *Descripción de los materiales.* Particularidades y localización de los materiales que son involucrados en el proceso productivo.
- *Descripción del sistema.* Conocido el espacio y actividad general, los puestos de trabajo y los materiales implicados, se realiza un resumen general de toda la actividad productiva con una secuencia bien definida que ubique al lector dentro del proceso y se observe como un sistema.

6.1.2 Caracterización del sistema productivo. Esta fase enmarca toda la información requerida para alimentar el modelo matemático y extiende de manera más detallada el entendimiento del proceso productivo.

Con base en la formulación, plataforma de este proyecto, se identifican y estudian las variables clave que conforman las ecuaciones del modelo matemático y los parámetros de entrada necesarios para su construcción. Para su análisis, se solicita información a la empresa mediante entrevistas informales a personas expertas y encargadas, además de observar, anotar y tabular. Se profundiza en: las implicaciones y requerimientos que trae consigo el proceso y los elementos que se toman en cuenta para la utilización adecuada de los recursos, todo para satisfacer la demanda requerida.

- *Materiales.* Comprende el estudio de materias primas, productos en proceso y productos terminados, sus características y especificaciones, como también su relevancia dentro del proceso productivo.
- *Almacenamiento e inventarios.* Es importante conocer algunos niveles de inventarios intermedios y capacidad máxima de almacenamiento relevantes para la construcción del modelo.

- **Procesos.** Se identifican y exponen las diferentes líneas de procesamiento.
- **Tareas.** Se estudian las operaciones que se llevan a cabo para la consecución de cada producto. Cabe anotar que una tarea es aquella que implica el mismo material de entrada y salida, en las mismas condiciones. Esta, puede ser realizada en una o más unidades pero nunca de manera simultánea. Conocer acerca de las tareas implica saber que unidad las puede realizar, que material consumen, que material producen; entre otros.
- **Unidades.** Se refiere a los puestos de trabajo o maquinaria comprometida en el proceso productivo. Una unidad puede realizar una o más tareas pero nunca de manera simultánea. Además de identificarlas es necesario caracterizarlas y conocer sus límites capacidad, sus requerimientos y saber que tareas pueden ser realizadas en cada unidad.
- **Demanda.** El mercado es una variable fundamental para toda organización. En este caso es necesario conocer la demanda de los productos mas relevantes, pues aparte de ser un parámetro de entrada, es un buen criterio para seleccionar cuales de ellos se involucran en el desarrollo del modelo y poder cumplir con los requerimientos establecidos.
- **Horizonte de programación.** Referido al tiempo que toma la consecución de un plan de producción definido.
- **Tiempos de procesamiento.** La observación detallada y organizada del proceso productivo debe estar acompañada con la medición del tiempo de procesamiento de cada tarea en cada unidad, estudio de su comportamiento y definición de tiempos de procesamiento constante y/o variable para cumplir con los tiempos de entrega.

6.2 ETAPA 2: CONSTRUCCIÓN DEL MODELO MATEMÁTICO

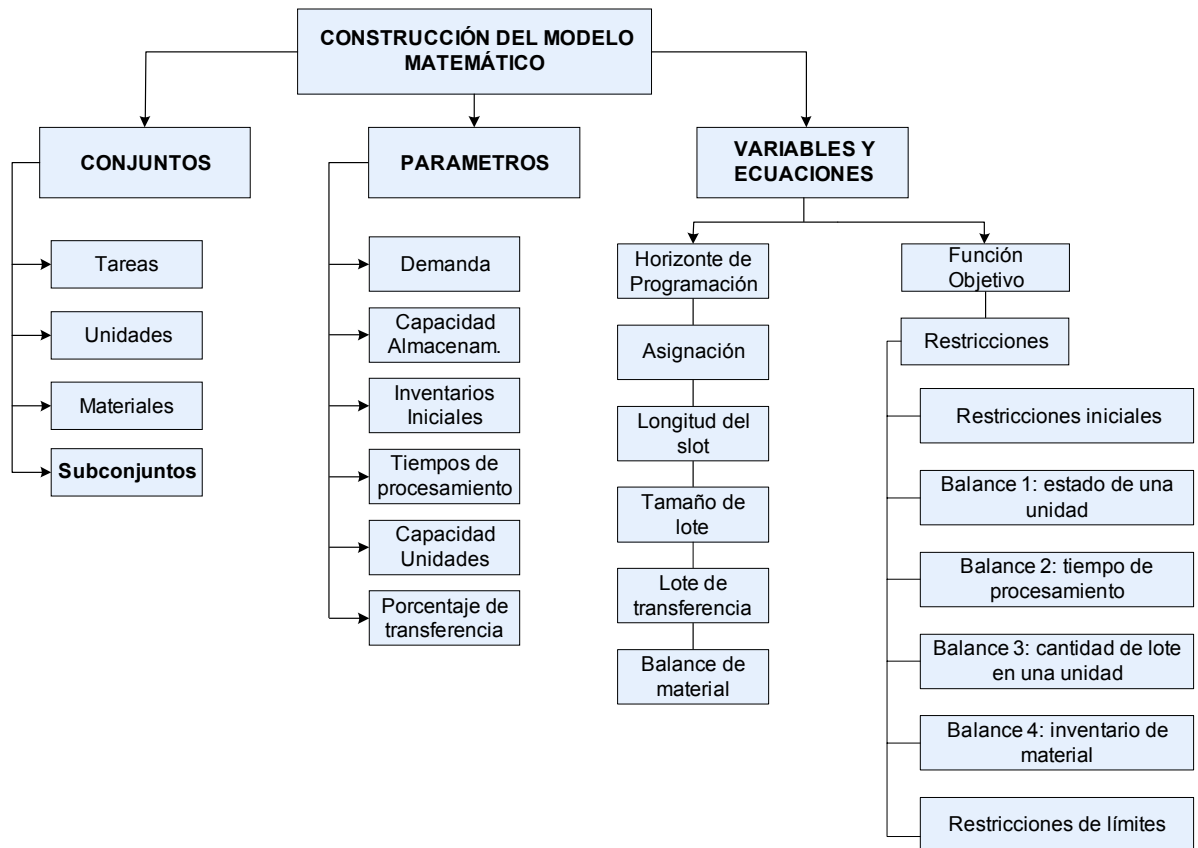
El problema se formula de manera conveniente para la construcción de un modelo matemático, que permita al encargado experimentar sobre este. Evitar hacerlo sobre el proceso real y como soporte técnico en el proceso de toma de decisión, debe generar ahorros significativos mediante el aprovechamiento del componente cuantitativo que arroja la solución del mismo. Para su construcción es necesario en primera medida definir las variables, identificar los parámetros y construir los conjuntos apropiados necesarios para el desarrollo y aplicación a este proceso productivo. Luego, se procede a determinar matemáticamente cada una de las dos partes que constituyen un modelo: a) la medida de efectividad que permite

conocer el nivel de logro de los objetivos, por lo general es una función (ecuación) llamada función objetivo; b) las limitantes del problema llamadas restricciones, que son un conjunto de igualdades o desigualdades que constituyen las barreras y obstáculos para la consecución del objetivo.

Simultaneo a la construcción, es importante realizar pruebas al modelo para corroborar la consistencia en las dimensiones de las variables, parámetros y ecuaciones y en la programación algebraica del mismo; de manera que al realizar las modificaciones pertinentes se corrijan los errores y el modelo arroje resultados razonablemente válidos. Para esto el modelo se compila tantas veces como sea necesario hasta que GAMS/CPLEX notifique que no hay errores de programación.

La siguiente figura muestra las fases de esta etapa.

Figura 18. Fases de la formulación y construcción del modelo matemático.



Los autores

6.2.1 Conjuntos. Es el encabezado del modelo, en ellos se identifican los elementos principales del proceso productivo e intersecciones que existen entre ellos. Son la base para definir por medio de subíndices, las relaciones entre las variables y con los parámetros, además de controlar las ecuaciones de manera ordenada y concisa. Entre estos encontramos: tareas, unidades, materiales, **slots** y los subconjuntos: unidades que puedes realizar ciertas tareas, tareas que consumen ciertos materiales y tareas que producen ciertos materiales.

6.2.2 Parámetros. Son los valores que alimentan el funcionamiento del modelo, cuantifican el cambio máximo y/o mínimo de las variables, y por ende las relaciones entre ellas. De su consecución depende el buen funcionamiento del modelo con respecto al proceso productivo real, su estudio se hace de manera cuidadosa de tal forma que la validación arroje resultados favorables y se acerque lo más posible a lo real. Entre estos encontramos: demanda, capacidades de almacenamiento, inventarios iniciales, tiempos de procesamiento, capacidad de unidades, porcentajes de transferencia y para el caso de maximización de beneficio, el tiempo u horizonte de programación.

6.2.3 Variables y ecuaciones. Son el cuerpo del modelo, representan de manera matemática el funcionamiento real del proceso por medio de sus relaciones y son definidas por el tipo de programación y técnica a ser aplicada. Con las variables se mide la efectividad del mismo, se hacen relevancias, se analizan las que de una u otra forma son de interés para la consecución del objetivo planteado y se concluye al respecto. A grandes rasgos son variables definidas para representar: horizonte de programación, asignación, longitud del **slot**, tamaño de lote, lote de transferencia y balance de material.

Las ecuaciones que corresponden a la escritura matemática de las relaciones entre variables y parámetros, son:

- ***Función objetivo.*** Es pertinente seleccionar cuál de los dos objetivos comúnmente utilizados en la literatura de programaciones se persigue: minimización de **makespan** o maximización de beneficio. Se define cual es el más conveniente para la organización; previo análisis de las características y políticas que intervienen en el sistema de producción de la organización. ...Ver página 52...
- ***Restricciones.*** Analizadas las características del sistema, conocidas todas sus variables y definida la función objetivo a utilizar, en este momento queda delimitar el campo de acción del sistema para la consecución del objetivo, mediante la definición del conjunto de ecuaciones y desigualdades que relacione variables y parámetros y describa lo más real posible el proceso de producción, de manera que las soluciones puedan ser aplicables a la realidad y de alguna manera beneficien a la organización. En este punto una de las ventajas y novedades de la

formulación a emplear, es que el autor define las restricciones basándose en balances de recursos.

1. *Restricciones generales.* Permiten que la programación se realice dentro del tiempo horizonte, hacen referencia a cual tarea comienza/finaliza, en que momento y en cual unidad y estipulan los límites inferior y superior del tamaño del lote para una tarea, en una unidad, en un momento dado. ...Ver en páginas 46-48...

2. *Balance 1: estado de una unidad.* Estas ecuaciones revelan la condición de una unidad (si comienza o finaliza o continúa una tarea) en el momento T_k y asegura que una unidad no empiece una tarea nueva a menos que haya finalizado la inmediatamente precedente. ...Ver en páginas 48-49...

3. *Balance 2: tiempo de procesamiento de una tarea.* Es fundamental definir el tiempo de procesamiento de las operaciones y declararlo como una función lineal, que se compone de una parte constante y otra variable que depende del tamaño del lote o de una de ellas. ...Ver en páginas 49-50...

4. *Balance 3: cantidad de lote en una unidad.* Se refiere a un balance de material en la unidad en un momento determinado. La cantidad de material se limita a la capacidad de almacenamiento de la unidad. Por otro lado, se garantiza que una unidad solo pueda descargar material en T_k si su tarea termina y que un lote de tarea no termina hasta que su duración haya terminado. ...Ver en página 50...

5. *Balance 4: inventario de material.* Este es uno de los balances más importantes y de mayor complejidad a la hora de formular y modelar. Muestra que la cantidad de inventario de material m en un momento T_k está dada por la relación entre el inventario inicial, el material consumido y el material producido. De esta manera también se asegura que una tarea nunca es realizada a menos que las materias primas requeridas estén presentes en sus inventarios respectivos. ...Ver en página 50-51...

6. *Restricciones de límites.* Finalmente se limitan las variables de la formulación para que no excedan sus valores máximos y mínimos posibles y así el modelo se acerque lo más posible a la realidad. ...Ver en páginas 51...

6.3 ETAPA 3: SOLUCIÓN Y VALIDACIÓN DEL MODELO

6.3.1 Solución. Se procede a utilizar el motor GAMS/Cplex para dar solución al modelo, encontrando los valores de las variables dependientes asociadas a las componentes controlables del sistema con el propósito de optimizar, si es posible, o cuando menos mejorar la eficiencia o la efectividad del sistema dentro del marco de referencia que se fijaron los objetivos y las restricciones del problema.

Dado el tamaño y complejidad del problema de programación lineal aquí presentado, el motor Cplex utiliza para su solución el método más apropiado entre sus métodos posibles, siendo en este caso concreto el método **branch and cut**.

- *Aplicación del algoritmo branch and cut (B&C)*. Como ya se ha venido mencionando, el algoritmo base para la solución del modelo es el branch and cut, que Cplex utiliza como método de solución. Mitchell^{*****} manifiesta que el algoritmo parte de un problema de PL formulado de manera estándar y en general, sigue las siguientes etapas:

1. *Inicialización*: se denota el problema inicial de programación lineal entera por PLE^0 y establece el conjunto de nodos activos $L=\{PLE^0\}$. Se implanta el límite superior para $\bar{z} = +\infty$. Se establece $\underline{z}_l = -\infty$ para un problema $l \in L$.

2. *Terminación*: si $L = \emptyset$, entonces la solución x^* la cual produjo el valor objetivo a cargo de \bar{z} es óptima. Si tal x^* no existe (por ejemplo, $\underline{z}_l = -\infty$) entonces el PLE es infactible.

3. *Selección del problema*: se selecciona y elimina un problema PLE^l de L .

4. *Relajación*: se soluciona la relajación de la programación lineal de PLE^l . Si la relajación es infactible, establece $\underline{z}_l = +\infty$ y va hacia la etapa 6. Indica \underline{z}_l como el valor objetivo óptimo de la relajación si este es finito y deja que x^{RI} sea una solución óptima; de otra manera establece $\underline{z}_l = -\infty$.

5. *Añadir hiperplanos cortantes*: de ser deseado, se busca los hiperplanos cortantes que son violados por x^{RI} , si ninguno es encontrado, los añade a la relajación y regresa al paso 4.

6. *Comprensión y recorte*:

- a. Si $\underline{z}_l \geq \bar{z}$, va al paso 2.

***** MITCHELL, Op. cit., p. 5-8

b. Si $\underline{z}_i < \bar{z}$ y x^{Rl} es factible entero, actualiza $\underline{z}_i = \bar{z}$, borra de L todos los problemas con $\underline{z}_i \geq \bar{z}$ y va al paso 2.

7. *División*: Deja $\{S^{lj}\}_{j=1}^{j=k}$ ser una división del conjunto de restricciones S^l del problema PLE^l . Añade problemas $\{PLE^{lj}\}_{j=1}^{j=k}$ a L , donde PLE^{lj} es PLE^l con la región factible restringida a S^{lj} y \underline{z}_{lj} para $j=1, \dots, k$ es unido al valor \underline{z}_i para el problema raíz. Va al paso 2.

Es necesario aclarar que L es el conjunto de nodos activos en el árbol **branch and cut**. El valor del punto factible mejor conocido para (PLE) es \underline{z}_i , el cual proporciona un límite superior en el valor óptimo de (PLE). Además, \underline{z}_i es el límite inferior en el valor óptimo del subproblema corriente bajo consideración. El valor de la relajación del subproblema de PL puede ser usado para actualizar \underline{z}_i . En algunas situaciones, un gran número de hiperplanos cortantes violados son encontrados en el paso 5, en tal caso es común clasificar los hiperplanos de alguna manera (puede ser por violación) y adherir solo un subconjunto. Los subproblemas formados en el paso 7 son llamados subproblemas menores, con el problema previo de PLE^l siendo el subproblema raíz. Usualmente la división toma la forma de una variable disyuntiva $x_i \leq a$ versus $x_i \geq a+1$ para alguna variable x_i y una entera a .

Las relajaciones pueden ser resueltas usando cualquier método para problemas de programación lineal. Normalmente, la relajación inicial es resuelta utilizando el método simplex. Las relajaciones siguientes son resueltas utilizando el método dual simplex, así la solución dual para la relajación del subproblema raíz es aún factible en la relajación del subproblema menor. Además, cuando hiperplanos cortantes son añadidos en el paso 5, la iteración dual corriente es todavía factible, entonces otra vez la relajación modificada puede ser resuelta usando el método dual simplex. También es posible usar un método de punto interior y este puede ser una buena opción si las relajaciones de programación lineal son extensas.

6.3.2 Validación. En este momento es útil examinar de nuevo la formulación del problema y compararla con el modelo, realizar diferentes corridas para corroborar que todas las expresiones matemáticas son consistentes con las dimensiones de las unidades que emplean y verificar que los resultados arrojados por él mismo garantizan valores acordes con la realidad y que cumplen con el objetivo considerado.

Además, puede obtenerse un mejor conocimiento de la validez del modelo variando los valores de los parámetros de entrada y/o de las variables de decisión y comprobando que los resultados del modelo se siguen comportando de una

manera factible.

Un enfoque más sistemático para la prueba del modelo consta en emplear una prueba retrospectiva. En esta prueba se utilizan datos históricos y se reconstruye el pasado para determinar si el modelo y la solución resultante hubieran tenido un buen desempeño, de haberse usado. La comparación de la efectividad de este desempeño hipotético con lo que el modelo arroja, indica si el uso del modelo tiende a dar mejoras significativas sobre la práctica actual. Puede también indicar áreas en las que el modelo tiene fallas y requiere modificaciones. Lo que es más, al emplear las alternativas de solución y estimar sus desempeños históricos hipotéticos, se pueden reunir evidencias en cuanto a lo bien que el modelo predice los efectos relativos de los diferentes cursos de acción.

En general lo que se busca en esta etapa es que la correlación entre la predicción del modelo y lo que ocurra en la vida real sea alta

**PARTE III
DESARROLLO DE
LA
METODOLOGÍA
PROPUESTA**

7. DESARROLLO DE LA METODOLOGÍA PROPUESTA

En este capítulo se desarrollará paso a paso la metodología expuesta en la sección anterior, centrando todos los esfuerzos en el sistema de producción de la empresa ITALCOL S.C.A. Bucaramanga.

7.1 ETAPA 1: OBSERVACIÓN Y CARACTERIZACIÓN DEL SISTEMA PRODUCTIVO

7.1.1 Diagnóstico actual del sistema. Hablar del sistema actual de producción es describir el tipo de configuración, los recursos, los materiales, otros elementos involucrados, sus relaciones y el sistema general de producción.

- *Descripción del área productiva.* Dentro del área productiva de ITALCOL se identifican 4 secciones importantes a saber: ...Ver Anexo D...

1. *Recibo de materia prima*, áreas dispuestas para el descargue de materiales harinosos empacados en lonas y granulados directamente de los contenedores trasportadores. Para los primeros el área se define como una bahía de recepción manual de materiales adjunta a una bodega para su almacenamiento; para los segundos se tienen tres áreas compuestas cada una por un par de cárcamos de recepción de 35 ton.

2. *Almacén de materias primas*, que comprende el almacenamiento de granos en ocho silos de diferente capacidad así: 3 de 2000 ton (silos 4-5-6), 3 de 1000 ton (silos 1-2-3) y 2 de 105 ton (7-8) y el almacenamiento de harinas en una bodega de 3600 m² dispuesta para ello.

3. *Producción*, distribuida en cuatro niveles e interconectada con las demás áreas a través de elevadores, arrastres y trasportadores entre si. En el nivel cero, ubicado en el sótano se encuentran: dos molinos, una mezcladora, tres quebrantadoras y dos enfriadoras. En el nivel uno se dispone de la oficina de producción, el dosificado que se compone de una consola de mando, una tolva-báscula de materiales y una tolva-báscula de líquidos; una enfriadora, tres empacadoras automáticas y una manual, un extruder, una secadora para extrusados, una secadora y una prelimpiadora para materias primas, una caldera, cinco tanques para almacenamiento y tres para consumo de líquidos. En el nivel dos se hallan, tres peletizadoras, la oficina múltiple de costo y producción y el laboratorio de calidad. Por último, el nivel tres se compone de las tolvas de almacenamiento de materia prima, producto en proceso y producto terminado, además de tres zarandas, una aceitadora y una prelimpiadora de producto en proceso.

4. *Almacén de producto terminado*, es el área donde el producto empacado, revisado y apilado, esta listo para su comercialización. Aquí se ubican dos bandas transportadoras móviles, una de ellas inclinada con pendiente positiva para el acopio en las partes más altas.

• *Descripción de los puestos de trabajo*. Se identifican los siguientes puestos de trabajo:

1. *Recibo de materias primas*: compuesto por una bahía de 40 m² y tres pares de cárcamos; dos de ellos llenados con un pala accionada de manera mecánica con la ayuda de operarios y otro llenado de manera hidráulica por la inclinación de 45° del camión, para que el grano caiga por efecto de su propio peso.

2. *Acondicionamiento de materia prima*: conformado por una prelimpiadora y una secadora; donde la materia prima que no cumple especificaciones de humedad e impurezas, es tratada.

3. *Vaceo*: este puesto de trabajo dispone de dos arrastres ubicados en bodega, en los cuales de forma manual se vacían las materias primas harinosas empacadas en lonas que van rumbo a producción.

4. *Molinos*: dos molinos se hallan en este puesto de trabajo, encargados de reducir el tamaño del grano o de pulverizar el mismo (especificaciones previamente instaladas), bien sea una materia prima o ciertos productos intermedios que necesitan de este proceso.

5. *Dosificado*: este puesto de trabajo se asume como el corazón del proceso productivo. Aquí, de manera computacional se realiza la dosificación formulada de las materias primas en una tolva-báscula, por medio de un programa que se encuentra en red con todas las plantas del país. Las materias primas caen a una mezcladora donde se compactan y forman la mezcla de los diferentes productos que ofrece la empresa.

6. *Extrusión*: en él se encuentra un extruder, para transformar y dar cocción a las diferentes mezclas procesadas en esta línea, a través de un tornillo sin fin y la adición de vapor. Una secadora y un puesto de empaque manual lo complementan. La secadora se encuentra articulada a él por medio de un tubo transportador y un extractor, por lo que ambas máquinas se comportan como una sola unidad.

7. *Peletizado*: la maquinaria que comprende este puesto de trabajo corresponde a tres peletizadoras que con la adición de vapor dan cocción al producto y lo moldean según requerimientos preestablecidos de tamaño de grano.

8. *Quebrantado*: tres quebrantadoras lo componen, donde al producto peletizado que lo necesite, se le reduce el tamaño por medio de dos cilindros móviles separados una distancia muy pequeña a través de la cual pasa el material.

9. *Zarandas*: tres zarandas son las encargadas de realizar una separación del producto terminado con residuos harinosos para evitar que se formen cúmulos de material.

Se debe anotar que los tres últimos puestos de trabajo conforman tres líneas bien definidas no interconectadas entre sí, inmersas en el proceso productivo general, puesto que cada peletizadora alimenta siempre la misma quebrantadora y esta a su vez la misma zaranda. Se asume entonces que cada una de las tres líneas, línea-pelet 1, línea-pelet 2 y línea-pelet 3, conforman una sola unidad de trabajo con una característica especial en la segunda línea, por poseer una máquina aceitadora al final de la misma. Si un producto solo necesita ser procesado en una máquina de alguna de las tres líneas, este debe y puede pasar por las siguientes máquinas (si las hay), sin sufrir ninguna transformación.

10. *Empaque*: Comprende cuatro máquinas empacadoras, dosificadoras y selladoras; dos automáticas para las líneas de harinas, peletizados y quebrantados y una manual para las línea de los extrusados; donde el producto terminado es empacado en sacos de lona.

a. *Tolvas de almacenamiento*: sobre casi toda el área productiva de esta empresa están ubicadas una serie de tolvas de almacenamiento, interconectadas entre sí por transportadores y elevadores; las cuales reciben, almacenan y suministran los materiales que intervienen en los diferentes puestos de trabajo mencionados. ...Ver Anexo E...

b. *Líquidos*: aquí se encuentran cinco tanques de almacenamiento, tres tanques de consumo y una tolva-báscula de líquidos, dispuesta para proveer el proceso productivo. ...Ver Anexo F...

c. *Caldera*: por medio de la cual se suministra vapor al extruder y peletizadoras.

- *Descripción de los materiales*. Los materiales involucrados en el proceso productivo de la organización se clasifican en materiales antes y después de la mezcla.

Los *materiales antes de la mezcla* no son más que todas las materias primas que pueden ser utilizadas en este proceso, las cuales se clasifican en: pesadas macro o mayores, pesadas micro, pesadas menores y líquidos.

1. *Pesadas macro*. Corresponde a los principales componentes de la mezcla, siendo las de mayor utilización y por ende mayor volumen, las podemos encontrar tanto en silos (materiales granulados) como en bodega (materiales harinosos) para ser encauzadas hacia las 14 tolvas que el dosificado dispone para proveer la tolva-báscula.

Por el objeto comercial de la empresa, existe un gran número de estas materias primas, sustitutivas entre si. Su compra depende de la disponibilidad en el mercado según las condiciones climáticas, tiempos de cosecha y de sus precios en el mercado a nivel nacional e internacional. Las más utilizadas son: maíz, arroz, torta de girasol, salvado de trigo, gluten de maíz, soya integral, yuca molida, sorgo; entre otras.

2. *Pesadas menores*. Por lo general son materiales harinosos que en comparación con las pesadas mayores requieren menos utilización en lo que a volumen, tamaño y cantidad respecta. Su peso en comparación con el total del lote, representa un factor máximo de 0,06832.

3. *Pesadas micro*. Son los llamados Px, premezclas o aditivos. Se distinguen drogas evolutivas, drogas gravadas y droga 16%. Son materiales de laboratorio y su adición a la mezcla la hace un operario de manera manual, debido a que su peso en comparación con el tamaño del lote es casi despreciable a un factor máximo de $6,45 * 10^{-3}$.

4. *Líquidos*. Son pesados en la tolva-báscula de líquidos y se agregan durante la mezcla del producto. Los más usados son: aceite de palma, cebo, melaza, aceite de pollo, cloruro de colina, entre otros. La adición de los líquidos no afecta el tiempo de procesamiento, siempre y cuando se encuentren disponibles en las cantidades requeridas.

Los *materiales después de la mezcla*, son los productos en proceso y los productos terminados,; que a su vez se clasifican en: harinas, peletizados, quebrantados y extrusados. A excepción de las harinas, por ser ya un producto terminado, todos requieren procesos adicionales.

1. *Productos en proceso*. Corresponde a los materiales que al salir del dosificado requieren procesos adicionales de cocción, formación, separación y extracción de calor o vapor, es decir, aquellos que van a ser peletizados después de mezclados, quebrantados después de peletizados, molidos después de mezclados, extrusados después de molidos y finalmente secados y enfriados.

2. *Productos terminados*. En la actualidad, la organización causa alrededor de 76 productos correspondientes a la avicultura, porcicultura, acuicultura, ganadería, equinos y mascotas; agrupados de acuerdo a especificaciones de material y procesos similares. Por ejemplo, en la avicultura se encuentran las llamadas

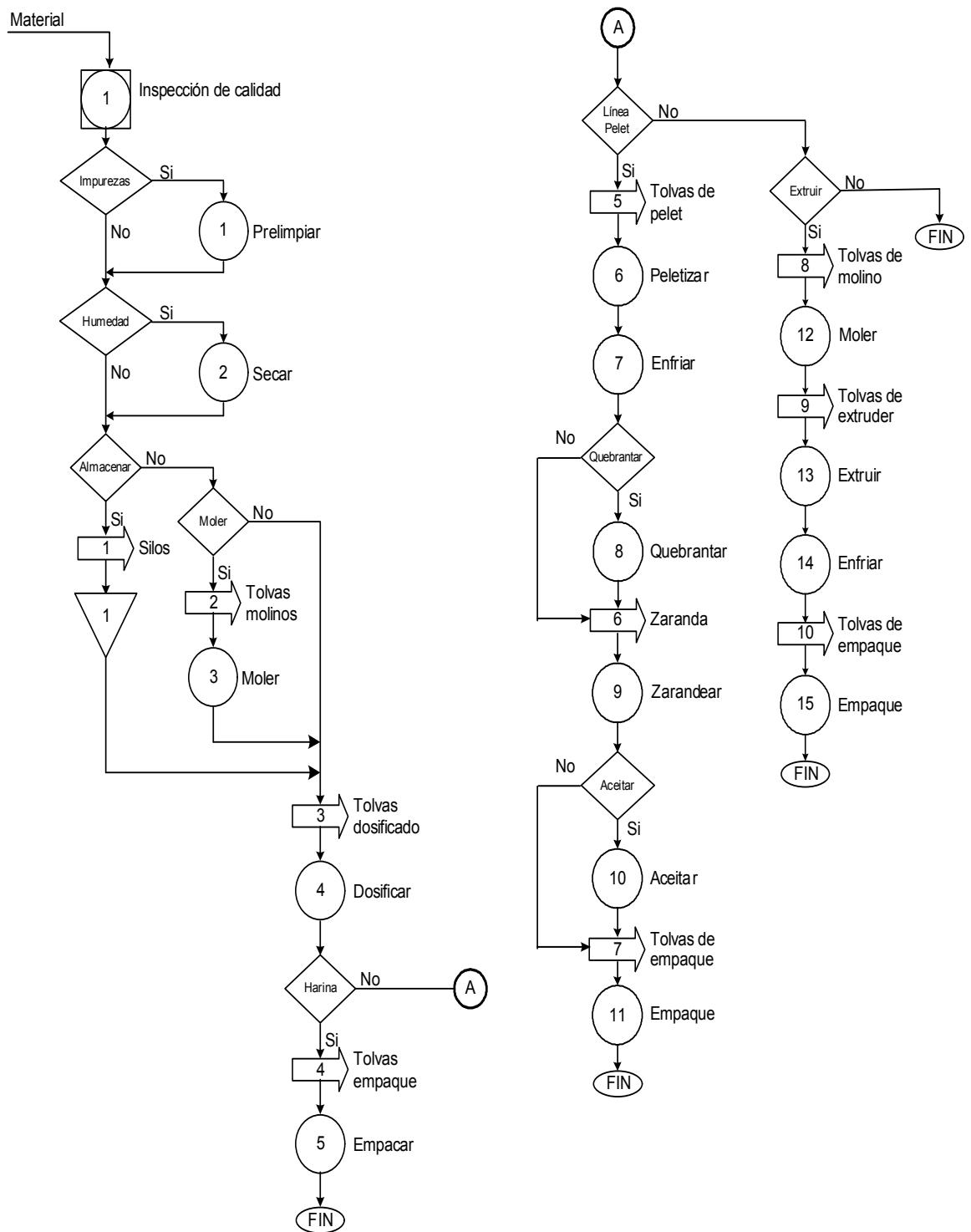
reproductoras, ponedoras, engorde, entre otras.

- **Descripción del sistema.** Las materias primas son descargadas según su contextura en las áreas dispuestas para ello. Inminente al descargue, control de calidad realiza una prueba para medir el porcentaje de humedad e impurezas que ellas presentan y si es necesario, se hace prelimpiar y/o secar para cumplir con estas especificaciones. Dichas materias se trasladan a sus respectivos sitios de almacenamiento, en donde se conservan con niveles de humedad y temperatura predefinidos, o son llevadas directamente hacia el área productiva, si esta así lo amerita. La transferencia de materia prima (granos y harinas), producto en proceso y terminado, se realizan por medio de elevadores y transportadores mecánicos por aire y tierra, que comunican a las diferentes áreas y puestos de trabajo con las 38 tolvas de almacenamiento, dispuestas sobre el área productiva. ...Ver Anexo E...

Las materias primas que se dirigen hacia el área productiva puede tomar dos rumbos bien definidos: el primero, conduce hacia molinos y atañe aquellas materias que por efectos de producción requieren la reducción de tamaño de grano, luego se encaminan hacia dosificado (segundo rumbo), donde en 14 tolvas les almacenan, disponen para su formulación y mezclan.

Terminado el tiempo de mezcla, el producto en proceso puede ser extrusado o peletizado. Antes de ir hacia el extruder, el material pasa por un proceso de molienda y en seguida el proceso de extrusión le da forma (hojuelas) y consistencia requerida. Por último, el material transita por la secadora donde se le extrae vapor y luego se dispone para su empaque y comercialización. Los productos peletizados se encaminan hacia cualquiera de las tres líneas-pelet conocidas; en la peletizadora, con la adición de vapor se transforma la mezcla en gránulos cilíndricos, luego se dejan caer a una enfriadora donde se disminuye su temperatura, se transporta a una quebrantadora, que para los productos que sea preciso, el tamaño del grano es alterado (si no es necesario el producto pasa por la quebrantadora sin sufrir ningún cambio) y por último se traslada a una zaranda la cual realiza una separación de granos y polvos finos que evita la formación de cúmulos. Los productos que necesiten la adición de aceites en su fase final, siguen únicamente la línea-pelet 2, puesto que dicha adición se hace en la máquina aceitadora siguiente a la zaranda. Preparado el producto terminado, se dirige a cualquiera de las 12 tolvas dispuestas para las dos máquinas dosificadoras, empacadoras y selladoras con motivo de su almacenaje y comercialización.

Figura 19. Esquema general del proceso de producción de ITALCOL S.C.A. Bucaramanga



Los autores

7.1.2 Caracterización del sistema productivo. A medida que se hace esta caracterización, consideraciones especiales se tienen en cuenta, que de manera lógica y concisa, excluyen y delimitan elementos del proceso y hacen que la construcción del modelo en función del objetivo planteado, no se sobrestime. Las mismas, llamadas asunciones, aparecen en el transcurso del presente apartado y garantizan que el modelamiento tenga en cuenta los cursos de acción mínimos, determinados por las componentes y variables controlables necesarias, en función de una muy buena representación de lo real. En primera instancia se delimita el proceso general productivo, tomando dos puntos de referencia como inicio y final del marco de acción del modelo.

El inicio, sitúa el momento en que las materias primas se encuentran disponibles para su pesaje y posterior mezcla, háblese de pesadas macro, micro, menores y líquidos, justo sobre el área de dosificado. Debido a políticas de la empresa y disposición de recursos, las existencias en almacenamientos siempre son muy altas, garantizando así disponibilidad de material primo ilimitada. Definimos nuestro punto de partida en el mismo instante en que las pesadas macro caen a la tolva-báscula, se suman los aditivos y la mezcla es inminente. Este primer tiempo de transferencia, marca el inicio del horizonte de programación.

El final lo define el momento en el que el producto terminado realiza su último recorrido, esto es cuando la totalidad del lote se encuentra alojada y disponible en cualquiera de las 14 tolvas de empaque dispuestas para ello. ...Ver cuadro 7 en página 88 y Anexo E...

- **Materiales.** Un estudio más detallado de los materiales antes de la mezcla, arroja que las pesadas micro, menores y líquidos, en su valor máximo alcanzan el 6,932% del lote, su adición se realiza iniciada la mezcla y no influye en el tiempo de procesamiento de la misma; por lo tanto se consideran despreciables para la construcción del modelo. En adelante las materias primas se refieren solo a las pesadas macro 93,068% del lote.

Se crean familias de materias primas que agrupan, cada una, las materias sustitutas entre si utilizadas en la formulación de los diferentes productos. Para la elaboración de un producto es necesaria la utilización de varias familias y dentro de cada familia se encuentran las materias primas para su formulación. Las proporciones de esta formulación son un parámetro de entrada que el encargado debe ingresar al modelo, siendo esta información confidencial por pertenecer al **Know How** de la empresa. El siguiente cuadro presenta la denominación de familias y sus materias primas.

Cuadro 6. Familias de materias primas.

FAMILIAS	MATERIAS PRIMAS
FMP 1	Sorgo - Maíz - Trigo - Arroz
FMP 2	Torta de ajonjolí - Torta de algodón - Torta de soya
FMP 3	Gluten de maíz - Harina de arroz - Mogolla de trigo
FMP 4	Sorgo - Maíz
FMP 5	Harina de arroz - Salvado de trigo - Mogolla de trigo - Germen de malta
FMP 6	Torta de soya - Torta de algodón
FMP 7	Harina de arroz - mogolla de trigo, (torta de maíz, yuca)
FMP 8	Sorgo - maíz - arroz
FMP 9	Torta de soya, soya integral, torta de algodón, torta de girasol, (torta de germen de maíz)
FMP 10	Salvado de maíz, salvado de trigo, harina de arroz (torta de palmiste)
FMP 11	Germen de malta - mogolla de trigo - harina de arroz

Los autores

La codificación de los materiales primos se hace en base al cuadro anterior, FMP#; para los productos en proceso y terminados se efectúa teniendo en cuenta las siglas que la organización maneja para sus productos, anteponiendo la inicial del proceso al cual se encuentra expuesto el material y definiendo, si es necesario para diferenciar, con una letra minúscula al final, la línea de producción a la que pertenece. Ejemplo:

M/SPE P p “Mezclar/Super Pollo Engorde Pintado peletizado”

M/SPE P q “Mezclar/Super Pollo Engorde Pintado quebrantado”

La consecución de los productos terminados depende del plan de producción a seguir, ya sea por los pedidos que realicen los clientes, con anticipación o en el transcurso de la jornada, generando una función de demanda que por lo general cambia varias veces en un mismo día. Los productos a modelar se discriminan al realizar un estudio de las ventas de los últimos meses, detallado en el apartado correspondiente a la demanda.

- **Almacenamientos e Inventarios.** Es claro que los inventarios de materia prima y producto terminado no son de interés para el desarrollo del modelo. Sin embargo, si lo son, los almacenamientos e inventarios intermedios que corresponden a algunas tolvas y sus capacidades, son supervisadas continuamente por un operario, encargado de suministrar información sobre las existencias en cada una de ellas y sobre su disponibilidad. Dichas capacidades, restringen la cantidad de lote en transferencia de producto en proceso y se citan en la tabla a continuación.

Cuadro 7. Tolvas de almacenamiento de producto en proceso y sus capacidades

PUESTO DE TRABAJO	TOLVAS	CAPACIDAD [bultos (c/u)]*
DOSIFICADO	1-4	250
	5-14	600
PELET 1	16, 18	400
PELET 2	23	700
	29	400
PELET 3	15, 17	400
MOLINOS	25-28	800
EXTRUDER	31, 32	300
EMPAQUE PELETS	19-22	600
	24, 30	700
	33-35	500
	36	600
	37	300
	38	250
EMPAQUE EXTRUDER	A y B	150

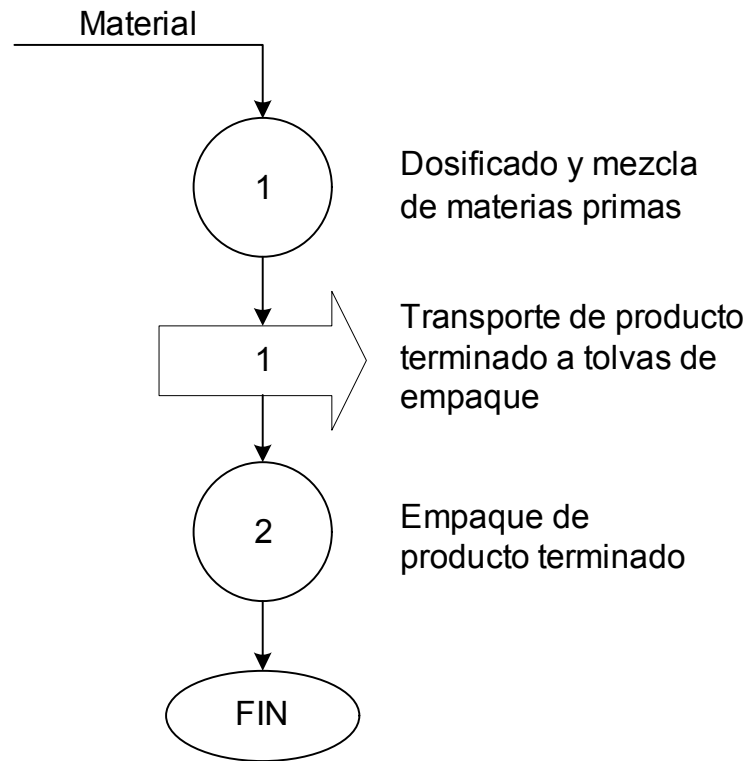
* 1 ton equivale a 25 bultos

Los autores

- **Procesos.** Los siguientes diagramas de proceso muestran la secuencia y operaciones de las líneas de producción definidas como: harinas, peletizados, quebrantados y extrusados.

1. Harinas

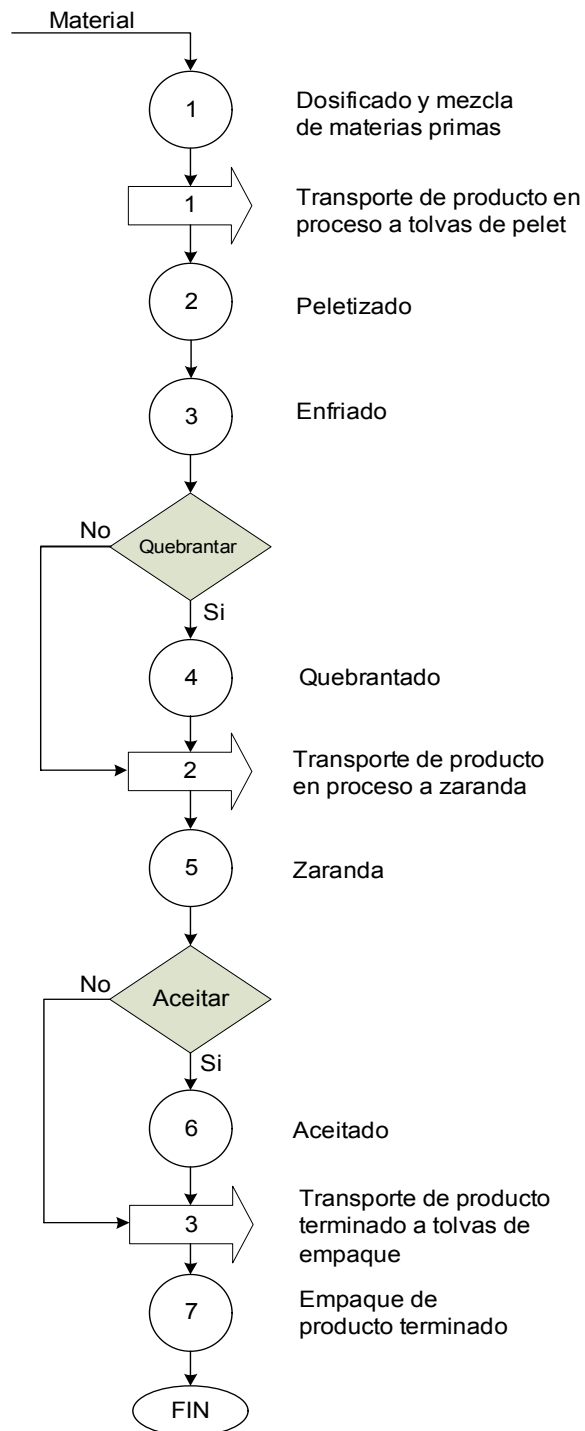
Figura 20. Diagrama de operaciones y secuencia de producción de la línea de harinas



Los autores

2. Peletizados y quebrantados

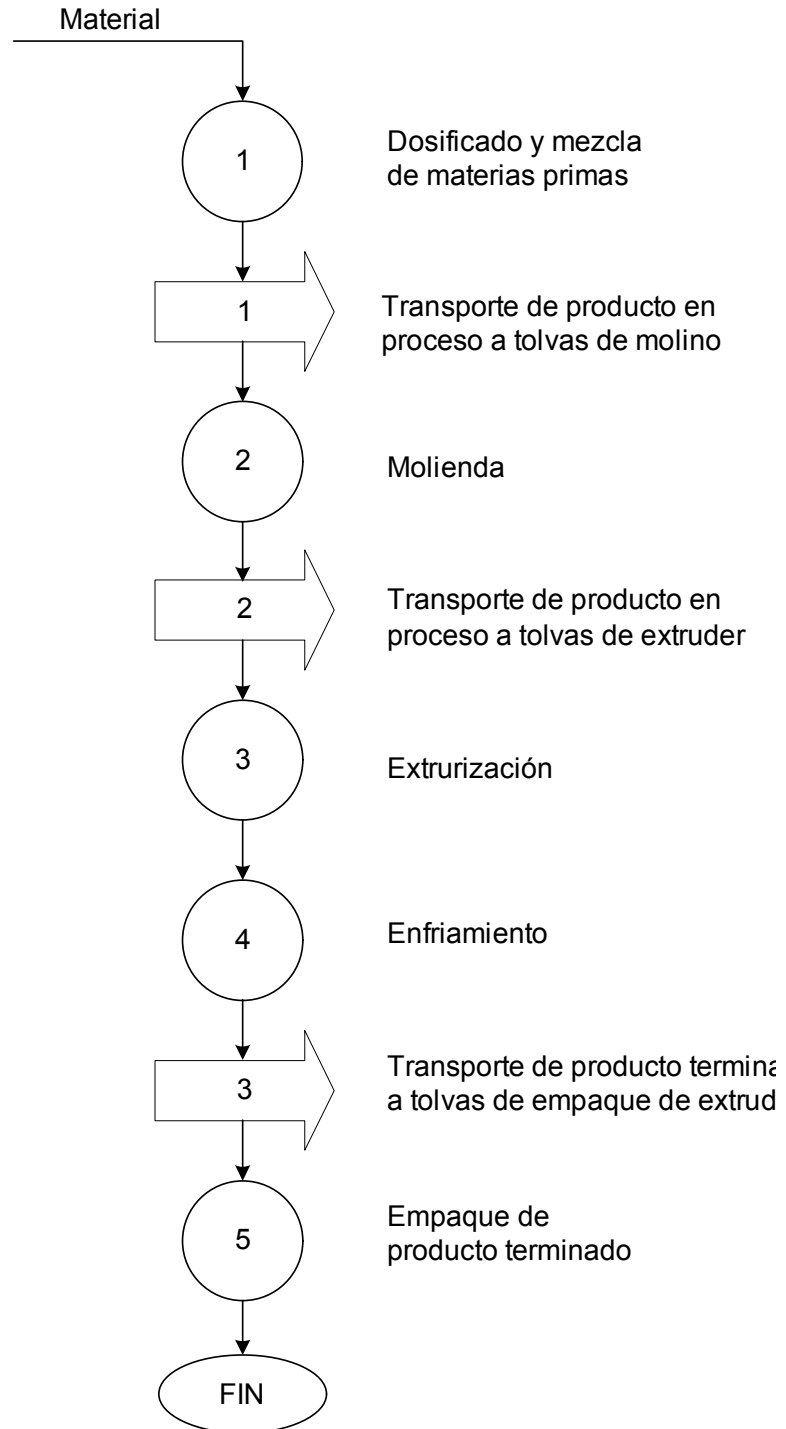
Figura 21. Diagrama de operaciones y secuencia de producción de la línea de peletizados y quebrantados



Los autores

3. Extrusados

Figura 22. Diagrama de operaciones y secuencia de producción de la línea de extrusados



Los autores

- *Tareas.*

1. *Mezclar.* Esta operación la realiza el dosificador en la consola de producción donde controla de manera automatizada todo este proceso. Después que corrobora la disponibilidad de materia prima en las tolvas de dosificado, empieza con la revisión de la fórmula y la programación del número de lotes a realizar con el software específico. Luego, por medio del control de mando activa la tolva-báscula que de manera automática y por medio de sensores pesa cada materia prima, necesaria para el lote de producción. Cuando el lote está listo, cae a la mezcladora, se activa e inicia el tiempo de mezcla, mientras las pesadas micro van siendo agregadas para que los materiales se fusionen en seco durante 30 segundos. Después si el producto lo amerita, se agregan los líquidos de manera automática mientras todos los materiales se van mezclando durante un tiempo estandarizado, que depende de la línea de producto a elaborar, la cantidad de lote a procesar, se rige también por un estándar de producción que se puede ver en el cuadro 8.

Cuadro 8. Tiempo de mezcla

LÍNEA DE PRODUCTO A ELABORAR	TIEMPO DE MEZCLA	LOTE MÁXIMO
Avicultura	5 min.	4 ton
Acuicultura	4 min.	3 ton

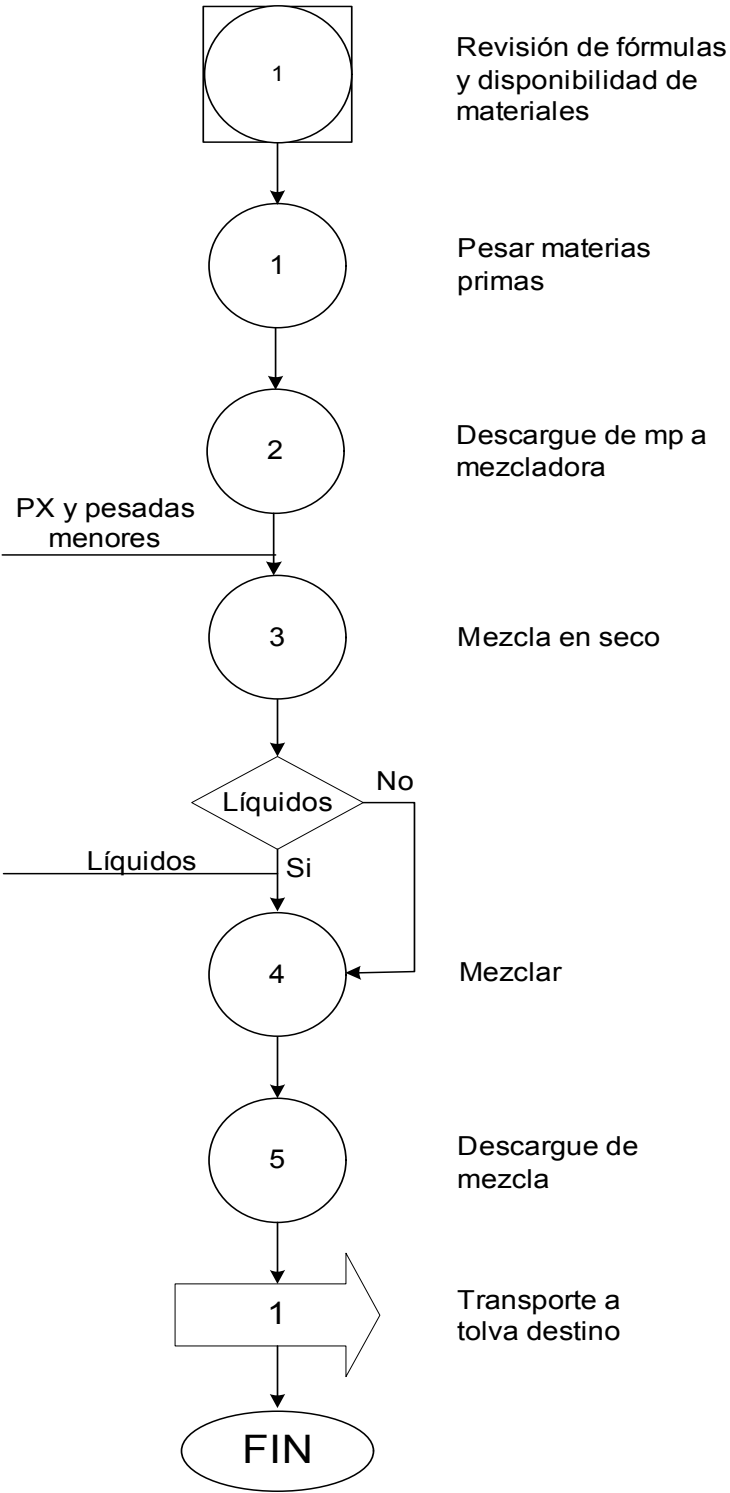
Los autores

Terminado el tiempo de mezcla, automáticamente se abre la compuerta de descargue durante 13 segundos, para que el producto en proceso o producto terminado (en el caso de las harinas) caiga a una tolva de alivio y luego a través de arrastres y elevadores, sea transportado hacia la tolva de destino, bien sea de empaque (harinas), línea-pelet (peletizado o quebrantado) o línea-extruder (extrusados).

Esta es una importante tarea requerida por todos los productos a fabricar y en nuestro caso, a modelar.

El siguiente es el diagrama de operaciones para esta tarea.

Figura 23. Diagrama de operaciones para la tarea mezclar

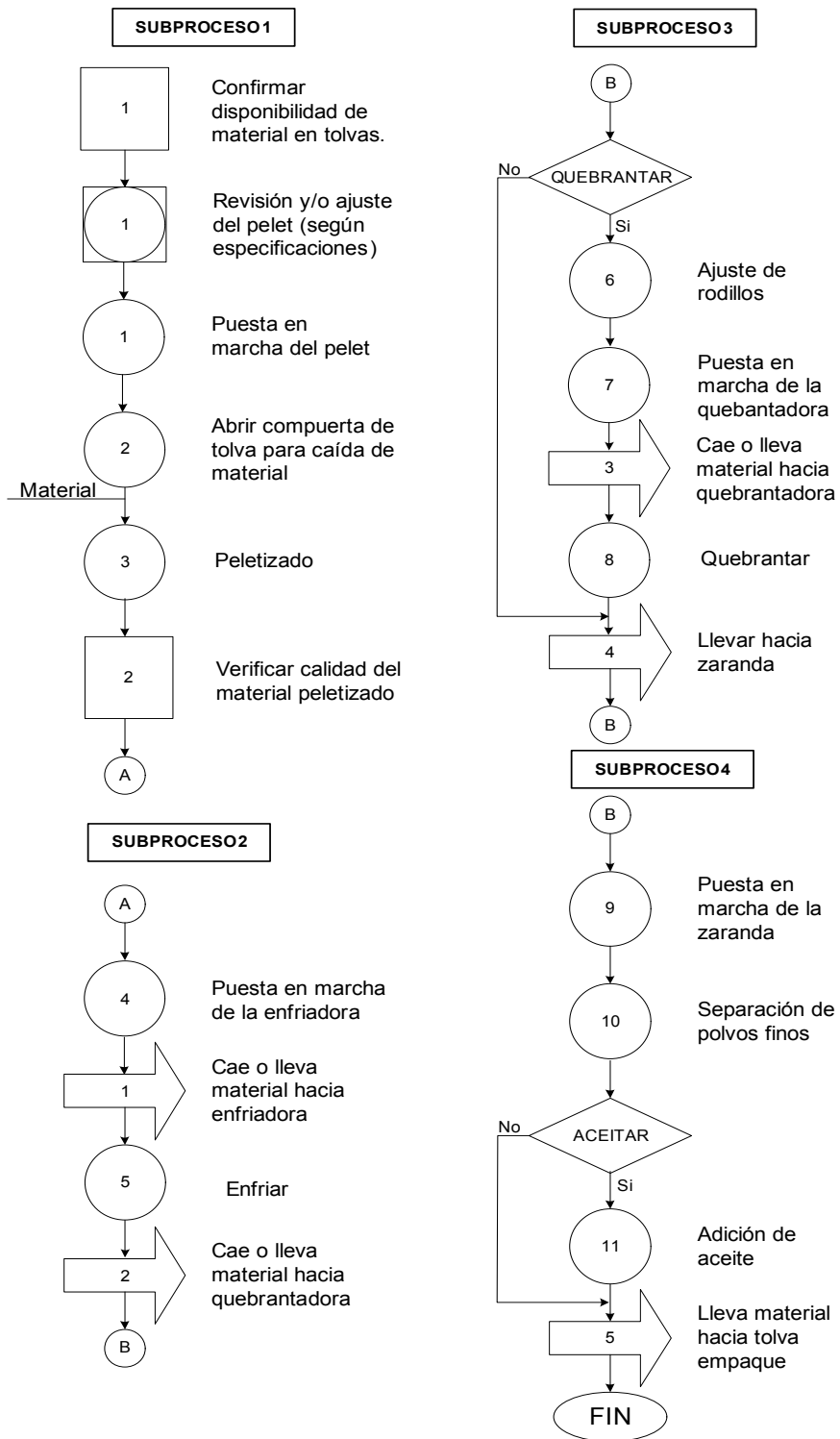


Los autores

2. *Peletizar y quebrantar*. Almacenada la mezcla de producto en proceso en las tolvas correspondientes a la línea-pelet programada, se abre la compuerta respectiva y el material cae a la tolva de alivio dispuesta sobre cada una. El material inicia un recorrido continuo por las diferentes máquinas correspondientes a la línea, asumidas como una sola unidad de trabajo ...Segunda viñeta en 7.1.1... El primer subproceso toma lugar en la peletizadora, que preparada con el dado apropiado da forma y tamaño al material. Luego para el segundo subproceso, el material cae a un enfriador que disminuye su temperatura; así se encamina hacia la quebrantadora, tercer subproceso, donde al material peletizado le es disminuido el tamaño de grano, esto para la tarea *quebrantar*. Para la tarea *peletizar*, el paso por la quebrantadora es obligatorio aunque el producto no sea quebrantado, sencillamente no sufre ninguna transformación y la quebrantadora se comporta como parte del transportador que conducen a la zaranda, cuarto subproceso, donde se efectúa la separación de polvos y el producto en su fase final es dirigido a la tolva de destino final para su empaque.

Es importante informar que el aceitado para los productos que lo requieran, se hace después de la zaranda solo por la Línea-Pelet 2; hace un recorrido por una tolva de recibo, un plato esparsor (donde es aceitado) y un tornillo sin fin para transportarlo a la tolva destino de empaque. La ilustración de este proceso, puede verse claramente en el siguiente diagrama de operaciones.

Figura 24. Diagrama de operaciones para las tareas peletizar y quebrantar



Los autores

3. *Moler*. En esta tarea el material (primo o en proceso) cae al molino desde la tolva respectiva y dependiendo del diámetro de criba, según especificación del material, es tratado y encauzado hacia dosificado para las materias primas o hacia el extruder para productos en proceso. Teniendo en cuenta la declaración y justificación del momento de inicio del modelo, en nuestro caso, esta tarea solo es utilizada para los productos a extrudir. Su diagrama de operaciones puede verse en la figura 25.

4. *Extrudir*. El material pasa por un tornillo sin fin, se especifica el valor de carga de vapor para su cocción, el cual varía por producto. En seguida, un tubo transportador lo conduce a un extractor que alimenta la secadora y de manera automática regula su temperatura y humedad. La figura 26 ilustra el diagrama de operaciones.

- *Unidades*. Dentro del contexto del modelo de producción, la estructura de las unidades involucradas en este proceso, exceptuando el dosificado, se define como la unión entre la máquina y la (s) tolva (s) de alimentación correspondiente, definiendo este último elemento la capacidad de la unidad.

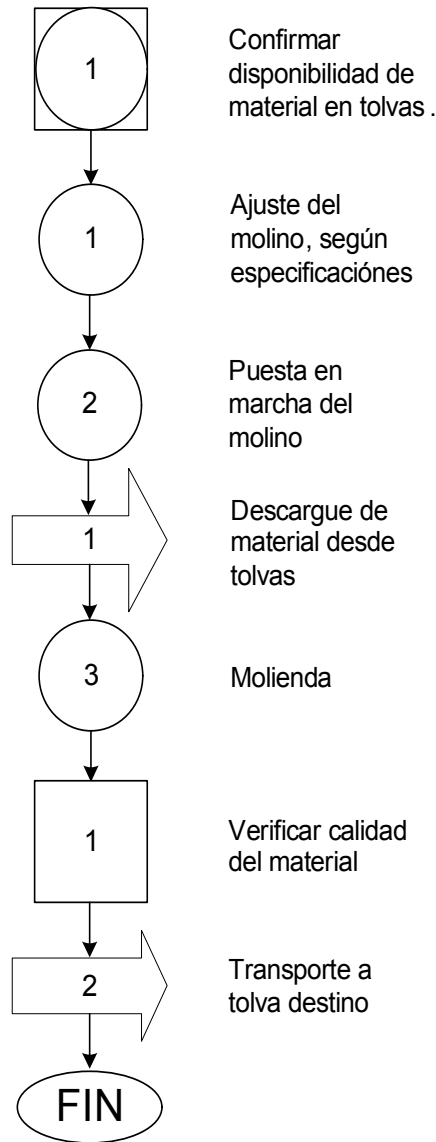
Cuadro 9. Límites capacidad de unidades

UNIDAD	ABREVIACIÓN	CAPACIDAD max	
		Bultos	Toneladas*
Dosificado	DOS	100 bultos	4 ton
Línea-Pelet 1	PEL1	800 bultos	32 ton
Línea-Pelet 2	PEL2	1100 bultos	44 ton
Línea-Pelet 3	PEL3	800 bultos	32 ton
Molino	MOL	1600 bultos	64 ton
Extruder	EXT	300 bultos	12 ton

Los autores

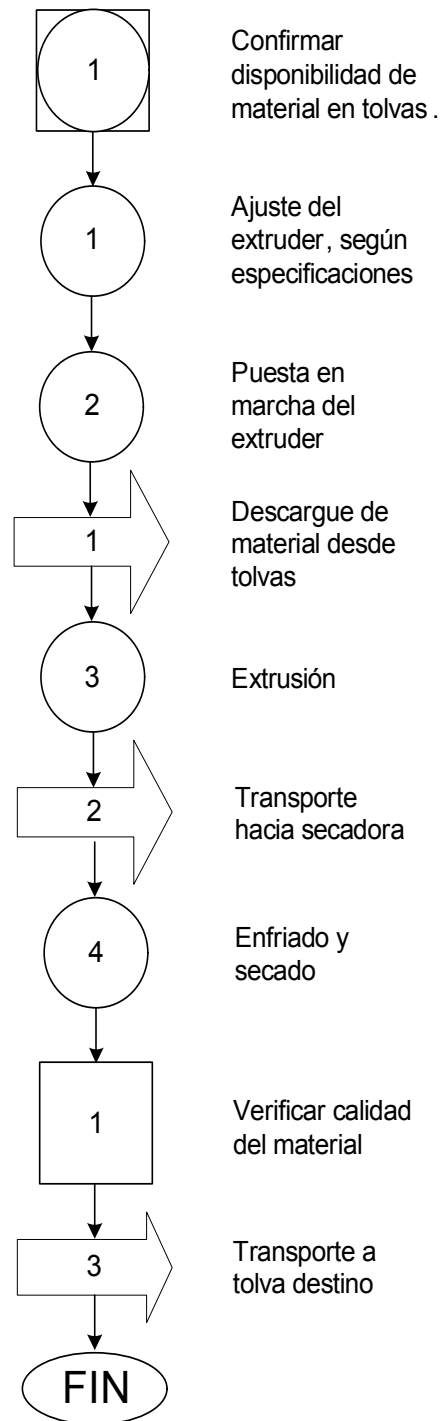
* 1 Bulto corresponde a 40 Kg.

Figura 25. Diagrama de operaciones para la tarea moler



Los autores

Figura 26. Diagrama de operaciones para la tarea extrudir



Los autores

- **Demanda.** Es de los parámetros más importantes para programar la producción de esta organización. Su valor varía con alta frecuencia (a veces en el transcurso de un mismo día) por la gran cantidad de productos que se manejan y las condiciones cambiantes del mercado; esta situación obliga a generar de manera continua un nuevo plan de producción, que mediante el empleo del modelamiento matemático, se ve mejorada al poder responder con mayor rapidez y facilidad, tan solo con un cambio sencillo en los parámetros de entrada.

De acuerdo a un análisis del historial de demanda de enero-agosto de 2006 ...Ver Anexo A/Demanda.xls... los productos finales de mayor rotación que involucran los procesos más representativos de la organización, se listan en el siguiente cuadro indicando su familia, nombre, código y porcentaje de producción.

Cuadro 10. Productos tipo para el modelo

DEMANDA ITALCOL enero-agosto de 2006 en kilogramos			
FAMILIA	PRODUCTO	CODIGO	%
POLLO ENGORDE	S.POLLO ENG PINTADO p	SPE_Pp	19,1862%
POSTURA	S.H.PREPICO HNA	SHPH	10,1217%
POLLO ENGORDE	S.PTO INICIACIÓN 2S	SPI_2S	9,9966%
POSTURA	S.HUEVO PREPICO 100 HNA	SHP_100h	6,2931%
POLLO ENGORDE	S.POLLO ENG PINTADO q	SPE_Pq	6,0663%
POSTURA	S.H.PREPICO CROMB	SHPq	5,9073%
POSTURA	S.HUEVO PREPICO 100 CROMB	SHP_100q	5,3113%
POLLO ENGORDE	S.PTO INICIACIÓN 3S	SPI_3S	4,2843%
POSTURA	PREPICO REPRODUCTORA	PRq	3,8741%
POLLO ENGORDE	PTO. PREINICIADOR	PPq	2,4935%
POLLO ENGORDE	POLLO CRIOLLO	PCp	2,3518%
POSTURA	POLLA LEVANTE CROMB	PLq	1,3236%
ACUACULTURA	SUPER MOJARRA 24 EXTRUDIZADA	SM 24 e	1,0394%
POLLO ENGORDE	POLLO CRIOLLO	PCq	1,0079%
ACUACULTURA	SUPER MOJARRA 34 EXTRUDIZADA	SM 34 e	0,3174%
		TOTAL	79,5746%

Los autores

Es claro que los productos que lideran la producción, pertenecen a la familia avícola y en un sentido más específico a la de pollo engorde y postura; sin embargo, se tienen en cuenta los productos que pertenecen a la familia acuicultura pese a su baja producción, ya que estos son los que se causan por la línea de producción de los extrusados y el no contemplarlos exime una parte importante del proceso, lo que hace que el modelo se aleje de la realidad.

Se observa que de la demanda total, se tienen en cuenta los productos que suman alrededor del 80% de la misma, lo que favorece el interés de los autores, de modelar en tanto sea posible la situación más real.

- *Horizonte de programación.* De acuerdo a las características de producción, a la configuración del sistema y al objetivo general planteado, esta es la variable que prima para medir la solución del modelo, ya que establecida la demanda, el objetivo se centra en minimizar este tiempo de producción.

- *Tiempo de procesamiento.* Conocidas y definidas las operaciones y secuencias que los productos deben seguir, se observan y obtienen los tiempos de procesamiento en cada unidad para cada tarea, de formatos de producción identificados para cada puesto de trabajo ...Ver Anexo A/Formatos... Esta información se analiza, tabula y resume en el cuadro 11. Este tiempo se compone de una parte constante, que con excepción del dosificado, corresponde al tiempo de alistamiento de las máquinas y otra parte variable que incluye el tiempo de producción y el tiempo de transferencia de material hacia el siguiente almacenamiento. El tiempo de producción varía según el tamaño de lote, el tipo y requerimientos del producto a procesar, es por esto que para cada producto se identifica la tasa de producción medida en bultos por hora (bts/hr) y su relación inversa para calcular el tiempo medido en horas que tarda en realizar un bulto, correspondiente a una unidad de masa.

Para el dosificado solo se considera la parte variable del tiempo, ya que el alistamiento del siguiente lote se realiza mientras el actual esta siendo procesado, es decir, dentro del tiempo de mezcla. Este tiempo obedece al tipo de producto que se realice ...Ver pág. 92... y contempla el tiempo de transferencia hacia la línea de producción específica.

El estudio estadístico de este cuadro se puede analizar en el anexo A/Tablas Tiempos y Parámetros.xls

Cuadro 11. Tiempos de procesamiento en cada unidad para cada tarea

TIEMPOS DE LAS UNIDADES												
MÁQUINA	DOSIFICADO					LINEA PELET 1 y 3					LINEA PELET 2	
	100 bul a Pe1	100 bul a Pe2	100 bul a Pe3	Rata Prod (bts/h)	α (const)	β (variable) Hr/bts	Rata Prod (bts/h)	α (const)	β (variable) Hr/bts	Rata Prod (bts/h)	α (const)	β (variable) Hr/bts
SPIG 2S	0,219222222	0,214259259	0,219958333	275,829	0,166666667	0,00362544	275,829	0,166666667	0,00362544	275,829	0,166666667	0,00362544
SPIG 3S	0,219222222	0,214259259	0,219958333	257,369	0,166666667	0,00388546	257,369	0,166666667	0,00388546	257,369	0,166666667	0,00388546
SPE P p		0,214259259					182,265	0,166666667	0,00548653			
SPE P q		0,214259259					176,902	0,166666667	0,00565285			
SHP h	0,214537037											
SHP q	0,219222222	0,214259259	0,219958333	159,923	0,166666667	0,00625	159,923	0,166666667	0,00625	159,923	0,166666667	0,00625
SHP 100 h	0,214537037											
SHP 100 q	0,219222222	0,214259259	0,219958333	183,137	0,166666667	0,00546	183,137	0,166666667	0,00546	183,137	0,166666667	0,00546
PC p	0,219222222	0,214259259	0,219958333	146,826	0,166666667	0,00681	146,826	0,166666667	0,00681	146,826	0,166666667	0,00681
PC q	0,219222222	0,214259259	0,219958333	119,34	0,166666667	0,00838	119,34	0,166666667	0,00838	119,34	0,166666667	0,00838
P R	0,219222222	0,214259259	0,219958333	207,37	0,166666667	0,00482	207,37	0,166666667	0,00482	207,37	0,166666667	0,00482
P L	0,219222222	0,214259259	0,219958333	194,93	0,166666667	0,00513	194,93	0,166666667	0,00513	194,93	0,166666667	0,00513
P P	0,219222222	0,214259259	0,219958333	257,863	0,166666667	0,00388	257,863	0,166666667	0,00388	257,863	0,166666667	0,00388
MÁQUINA	DOSIFICADO					MOLINO					EXTRUDER	
PRODUCTO	75 bul α (hrs)					Rata Prod (bul/h)	α (const)	β (variable) Hr/bto	Rata Prod (bul/h)	α (const)	β (variable) Hr/bto	
M 24	0,201111111					40	0,166666667	0,02500	46,7309	0,166666667	0,02140	
M 34	0,201111111					30	0,166666667	0,03333	30,8898	0,166666667	0,03237	

7.2 ETAPA 2: CONSTRUCCIÓN DEL MODELO MATEMÁTICO

En este momento se tiene claridad sobre el proceso, las variables y ecuaciones que lo describen y los parámetros al que esta sujeto. Se procede a escribir el código en GAMS/CPLEX, a compilar y corregir errores y a estudiar la solución que se obtenga.

Para dicha escritura, se aprovecha el haber modelado de manera algebraica los ejemplos provistos en la formulación estudiada, de tal forma que la construcción del modelo final, requiera solo la inyección de los parámetros establecidos por el proceso real y algunas modificaciones muy sencillas en ciertas ecuaciones. ...Ver Anexos B y C y página 53, sección Evaluación...

Anterior a la manipulación del problema, es *fundamental* el diseño y arquitectura del diagrama correspondiente a la técnica de modelamiento elegida, en este caso, el diagrama receta RD. ...Ver figura 28 o 29... La construcción del modelo depende de manera directa de este y gracias a su representación gráfica, hace más fácil y práctica la comprensión del mismo. En el RD se encuentra recopilada la información referente a los flujos de material, tareas y unidades involucradas, los materiales que se causan en cada tarea y a su vez en cada unidad, porcentajes de transferencia y codificación de cada material, tarea y unidad; que acompañado de los parámetros del proceso, marcan el inicio de la escritura del código fuente para el modelo ITALCOL 1.1 en el lenguaje de modelado algebraico GAMS.

A la fecha de inicio de construcción de este primer prototipo no se ha procesado un análisis exhaustivo de demanda, por eso, la selección de los productos y sus proporciones se basan en la experiencia y observaciones del encargado. Su RD se encuentra en la figura 28, el código fuente en el Anexo G y el archivo como tal en el Anexo A/ITALCOL 1.1.gms. Este prototipo se construye como plataforma del modelo final, con el objetivo de corregir todos los errores de programación y que este último soporte ajustar los elementos referentes a parámetros de entrada, mientras se recolectan y alimentan los datos e información necesaria del proceso productivo.

Posterior al estudio de los resultados arrojados por el análisis de demanda y en general toda la caracterización del sistema productivo...Ver página 98 y sección 7.1.2..., se realiza el RD final, figura 29 y dos cuadros 12 y 13, que resumen los parámetros a tener en cuenta para la construcción del prototipo final. Esta construcción se enfoca en alimentar el modelo con toda esta información. Se inicia con la definición de los conjuntos, subconjuntos y sus elementos, se precisan los parámetros de demanda, inventarios, capacidades y tiempos de procesamiento; todo lo referente a definición de variables y ecuaciones no representa gran

atención ya que esta parte del código se desarrolló con anterioridad de forma general, es decir que puede ser usada para diferentes modelos al estar escrita completamente de manera algebraica, siendo la asignación de letras para los subíndices el único componente que se debe respetar y mantener.

Se finaliza entonces con la apreciación del modelo dividido en dos grandes partes. La primera refiere todo lo relacionado a definición de conjuntos y parámetros y la segunda representa la estructura matemática de las variables y ecuaciones relacionadas entre si para modelar el sistema productivo. A continuación se expone la segunda parte, el código fuente de este modelo llamado ITALCOL 2.1 se puede ver en el Anexo H y su archivo se encuentra en el Anexo A/ITALCOL 2.1.gms. Este prototipo final comparado con su antecesor, presenta algunas diferencias en los productos y proporciones demandas, los ahora tenidos en cuenta corresponden al historial real de producción demandada de los últimos meses. ...Ver Anexo A/Demanda.xls...

Definición de Variables

Variable Binarias

$Y(i,j,k)$ variable que indica si la unidad j comienza la tarea i en el momento $T(k)$

Variables Positivas

$Z(j,k)$ variable que indica si la unidad j *comienza una tarea* en el momento $T(k)$

$ym(i,j,k)$ si la unidad j esta continuando con la tarea i en el momento $T(k)$

$YE(i,j,k)$ si la unidad j finalizo la tarea i y descarga su lote en el momento $T(k)$

$SL(k)$ Longitud del **slot** k

$B(i,j,k)$ Tamaño del lote de la tarea i en la unidad j comenzando en $T(k)$

$bm(i,j,k)$ Cantidad de material que reside en la unidad j justo antes de $T(k)$

$BE(i,j,k)$ Cantidad que la tarea i descarga a su realización en $T(k)$

$t(j,k)$ Denota el tiempo faltante a $T(k)$ para completar la tarea que estuvo en progreso durante el **slot** k en la unidad j

$N(m,k)$ Balance de material m en $T(k)$

Variable Libre

MS Tiempo total de producción

Ecuaciones

Este primer grupo de ecuaciones, limita algunas variables en su valor máximo y mínimo en casos específicos como en el inicio y/o final del horizonte de programación ($k=0$ y $k=K$ respectivamente),

$Z.lo(j,k)=0;$

$Z.up(j,k)=1;$

$ym.lo(i,j,k)=0;$

$ym.up(i,j,k)=1;$

$YE.lo(i,j,k)=0;$
 $YE.up(i,j,k)=1;$
 $SL.up("k0")=0;$
 $Z.up(j,k)\$(ord(k)=card(k))=0;$
 $Y.up(i,j,k)\$(ord(k)=card(k))=0;$
 $B.up(i,j,k)\$(ord(k)=card(k))=0;$
 $ym.up(i,j,"k0")=0;$
 $ym.up(i,j,k)\$(ord(k)=card(k))=0;$
 $YE.up(i,j,"k0")=0;$
 $t.up(j,"k0")=0;$
 $t.up(j,k)\$(ord(k)=card(k))=0;$
 $bm.up(i,j,"k0")=0;$
 $bm.up(i,j,k)\$(ord(k)=card(k))=0;$
 $BE.up(i,j,"k0")=0;$

MakeSpan.. $\text{Sum}(k\$(ord(k)>1), SL(k)) = e = MS;$ Garantiza que todas las operaciones se realizan dentro del horizonte de programación, es nuestra función objetivo.

$\text{Inicio}(j,k)\$(ord(k)<card(k))..$
 $Z(j,k) = e = \text{Sum}(i\$(JI(j,i), Y(i,j,k));$ Asigna e indica el comienzo de una nueva tarea en todas las unidades

$\text{Tamano1}(i,j,k)\$(JI(j,i) \text{and } ord(i)>1 \text{ and } ord(k)<card(k))..$
 $B(i,j,k) = l = Bu(i,j)*Y(i,j,k);$ Limita la capacidad máxima de las unidades

$\text{Tamano2}(i,j,k)\$(ord(i)>1 \text{ and } ord(i)<12 \text{ and } JI(j,i) \text{ and } ord(k)<card(k))..$
 $B(i,j,k) = g = Bl(i,j)*Y(i,j,k);$ Limita la capacidad mínima de las unidades

$\text{Estado1}(i,j,k)\$(ord(k)>1 \text{ and } JI(j,i) \text{ and } ord(k)<card(k))..$
 $ym(i,j,k) = e = ym(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k);$ Define el estado de la unidad j en cualquier momento.

$\text{Estado2}(j,k)\$(ord(k)>1 \text{ and } ord(k)<card(k))..$
 $Z(j,k) = e = \text{sum}(i\$(JI(j,i), YE(i,j,k));$ Garantiza que un lote sea descargado solo una vez de la unidad j en el momento t(k).

$\text{Tiempo}(j,k)\$(ord(k)<card(k))..$
 $t(j,k+1) = g = t(j,k) + \text{sum}(i\$(JI(j,i) \text{ and } ord(i)>1), c(i,j)*Y(i,j,k) + v(i,j)*B(i,j,k)) - SL(k+1);$
 Garantiza que si la tarea no se completa en el slot k, lo hará en el siguiente.

$\text{Lote1}(i,j,k)\$(ord(k)>1 \text{ and } JI(j,i) \text{ and } ord(i)>1)..$
 $bm(i,j,k) = e = bm(i,j,k-1) + B(i,j,k-1) - BE(i,j,k);$ Balance del lote de material de la tarea i en la unidad j en el momento t(k)

Lote21(i,j,k) $\$(ord(i)>1 \text{ and } ord(i)<14 \text{ and } ord(k)>1 \text{ and } ord(k)<card(k) \text{ and } JI(j,i))..$
 $bm(i,j,k) = g = Bl(i,j)*ym(i,j,k);$ Garantiza que el lote no exceda la capacidad permitida por la unidad j procesando la tarea i

Lote22(i,j,k) $\$(ord(k)>1 \text{ and } ord(k)<card(k) \text{ and } JI(j,i) \text{ and } ord(i)>1)..$
 $bm(i,j,k) = l = Bu(i,j)*ym(i,j,k);$ Garantiza que el lote respete la capacidad mínima necesaria por la unidad j procesando la tarea i

Lote31(i,j,k) $\$(ord(i)>1 \text{ and } ord(i)<14 \text{ and } ord(k)>1 \text{ and } ord(k)<card(k) \text{ and } JI(j,i))..$
 $BE(i,j,k) = g = Bl(i,j)*YE(i,j,k);$ El lote descargado cumple con la cantidad mínima necesaria por la unidad j para procesar la tarea i

Lote32(i,j,k) $\$(ord(k)>1 \text{ and } ord(k)<card(k) \text{ and } JI(j,i) \text{ and } ord(i)>1)..$
 $BE(i,j,k) = l = Bu(i,j)*YE(i,j,k);$ El lote descargado no excede la capacidad máxima de la unidad j para procesar la tarea i

Lote4(j,k) $\$(ord(k)>1 \text{ and } ord(k)<card(k))..$
 $t(j,k) = l = \text{sum}(i\$(JI(j,i) \text{ and } ord(i)>1), c(i,j)*ym(i,j,k) + v(i,j)*bm(i,j,k));$
 Balance del Tiempo de procesamiento

Inventario(m,k)..
 $N(m,k) = e = li(m)\$(ord(k)=1) + N(m,k-1) + \text{sum}(i\$(PM(i,m), \text{sum}(j\$(JI(j,i), BE(i,j,k))) - \text{sum}(i\$(CM(i,m), \text{sum}(j\$(JI(j,i), PCM(i,m)*B(i,j,k))));$
 Balance de la cantidad de material en cada estado, en cualquier momento

Y las siguientes ecuaciones que limitan algunas variables evitando redundancia dentro del modelo.

Limite1(k).. $SL(k) = l = \text{smax}(j, \text{smax}(i\$(JI(j,i) \text{ and } ord(i)>1), c(i,j) + v(i,j)*Bu(i,j)));$

Limite2(j,k).. $t(j,k) = l = \text{smax}(i\$(JI(j,i) \text{ and } ord(i)>1), c(i,j) + v(i,j)*Bu(i,j));$

Limite3(m,i,k) $\$mc(m).. N(m,k) = l = Ca(m);$

Limite4(i,j,k) $\$(JI(j,i) \text{ and } ord(i)>1 \text{ and } ord(k)>1 \text{ and } ord(k)<card(k))..$
 $B(i,j,k) = l = Bu(i,j);$

Limite5(i,j,k) $\$(JI(j,i) \text{ and } ord(i)>1 \text{ and } ord(k)>1 \text{ and } ord(k)<card(k))..$
 $bm(i,j,k) = l = Bu(i,j);$

Limite6(i,j,k) $\$(JI(j,i) \text{ and } ord(i)>1 \text{ and } ord(k)>1 \text{ and } ord(k)<card(k))..$
 $BE(i,j,k) = l = Bu(i,j);$

Demanda(m,k) $\$(md(m) \text{ and } ord(k)=card(k)).. N(m,k) = g = D(m);$ Cumplimiento de la demanda de cada producto.

Model itacol /all/;
Solve itacol using mip minimizing MS;

El componente algebraico en la definición de las ecuaciones es evidente. De esta manera el modelo ofrece la posibilidad de reprogramar con rapidez la producción, mediante cambios muy sencillos en la entrada de datos y así encontrar una respuesta rápida y confiable.

Como última medida y en pro de nuestro principal objetivo, se diseña una interfase gráfica en Microsoft Office Excel basada en manejo de objetos por medio de Visual Basic, que permite al usuario modificar los parámetros de manera sencilla y evite manipular el código fuente, realice cambios en cualquier momento, corra el modelo y tome decisiones. Esta interfase contempla: selección de productos, de puestos de trabajo y cantidades a ser producidas. Su diseño se puede observar en la figura a continuación y su archivo como tal en el AnexoA/INTERFAZ2.2.xls.

Para la lectura de los datos definidos en la interfase, se hace uso de la herramienta *wtools/xls2gms* y se modifica el código fuente del modelo ITALCOL 2.1, dando como resultado la estructura del modelo final ITALCOL 2.2 para maximizar beneficio e ITALCOL 2.2_MS para minimizar **makespan**. La estructura del código fuente para el caso de maximización, puede detallarse en el Anexo I y sus archivos electrónicos en el AnexoA/ITALCOL 2.2.gms y AnexoA/ITALCOL 2.2_MS.gms, respectivamente.

Figura 27. Interfase gráfica para el modelo ITALCOL 2.2

PRODUCTOS		UNIDAD	
<input type="checkbox"/>	680 SHP h	<input checked="" type="checkbox"/>	DOSIFICADO
<input type="checkbox"/>	SHP q	<input checked="" type="checkbox"/>	LINEA PELET 1
<input type="checkbox"/>	SHP 100 h	<input checked="" type="checkbox"/>	LINEA PELET 2
<input type="checkbox"/>	1200 SHP 100 q	<input checked="" type="checkbox"/>	LINEA PELET 3
<input type="checkbox"/>	265 SPE P p	<input checked="" type="checkbox"/>	MOLINO
<input type="checkbox"/>	SPE P q	<input checked="" type="checkbox"/>	EXTRUDER
<input type="checkbox"/>	1000 SPI 2S		
<input type="checkbox"/>	SPI 3S		
<input type="checkbox"/>	830 PCp		SLOTS (K)
<input type="checkbox"/>	PCq		<input type="text" value="k0*k38"/>
<input type="checkbox"/>	PR		HORIZONTE (H)
<input type="checkbox"/>	PL		<input type="text" value="10"/>
<input type="checkbox"/>	PP		
<input type="checkbox"/>	597 MOJ 24		
<input type="checkbox"/>	MOJ 34		

Los autores

Figura 28. Diagrama RD ITALCOL 1.1

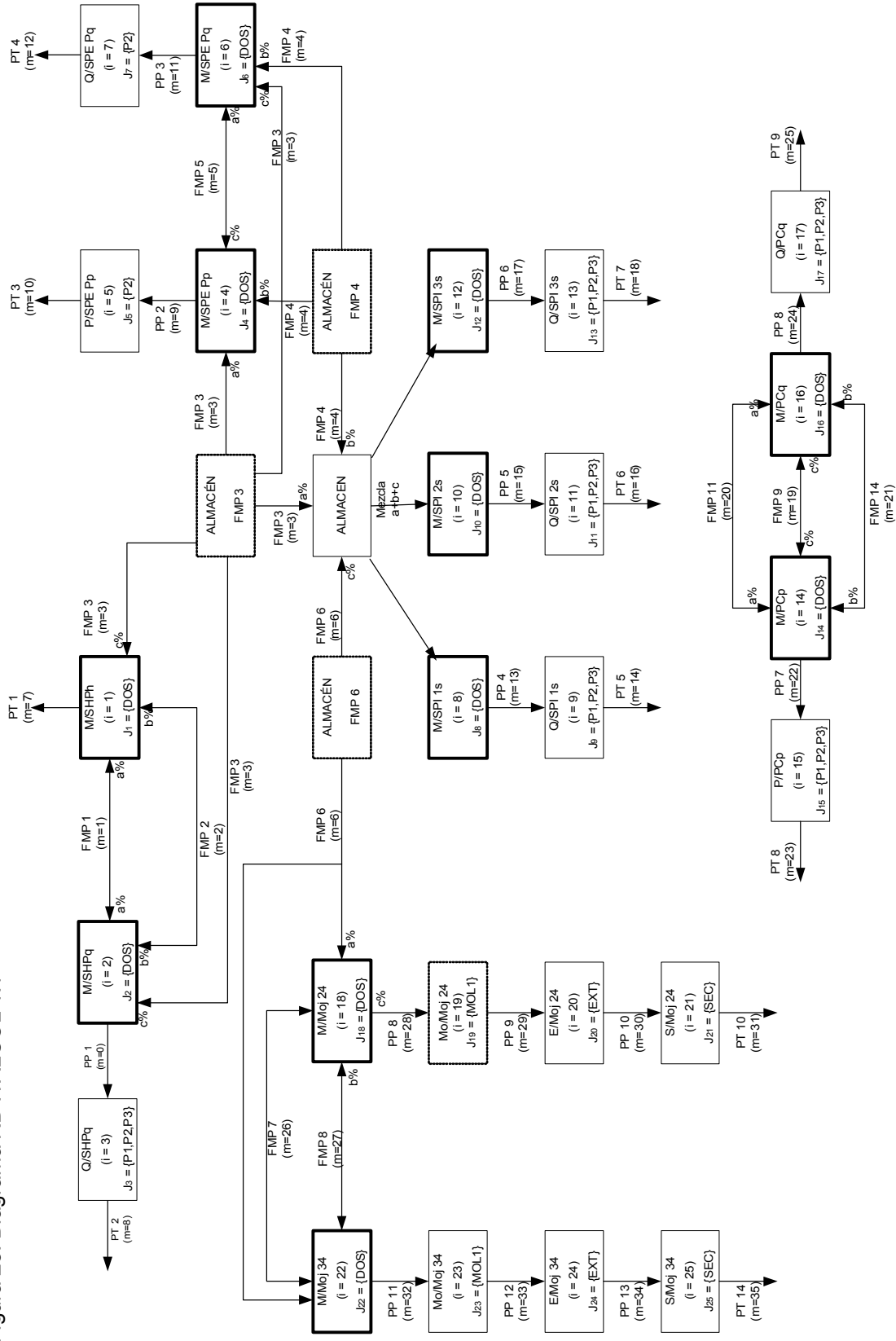
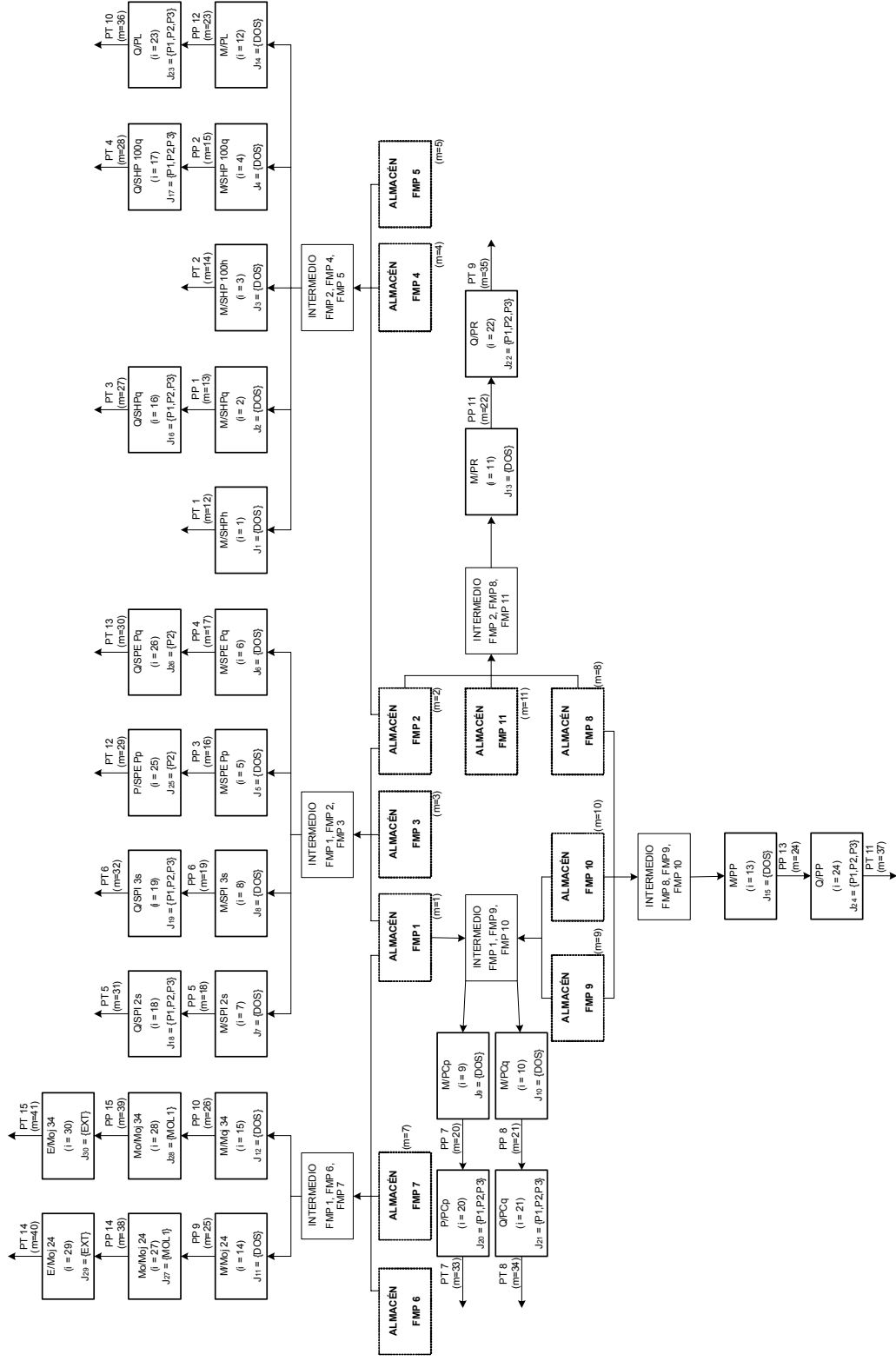


Figura 29. Diagrama RD ITALCOL 2.1



Los autores

Cuadro 12. Parámetros de tareas y unidades

TAREA i	UNIDAD	α_{ij}	β_{ij}	B_{ij}^L		B_{ij}^U	
				TON	BTS	TON	BTS
M-SHP	DOS	0,2145	0	4	100	4	100
M-SHPq	DOS	0,217813272	0	4	100	4	100
M-SHP_100h	DOS	0,214537037	0	4	100	4	100
M-SHP_100q	DOS	0,217813272	0	4	100	4	100
M-SPE_Pp	DOS	0,214259259	0	4	100	4	100
M-SPE_Pq	DOS	0,214259259	0	4	100	4	100
M-SPI_2S	DOS	0,217813272	0	4	100	4	100
M-SPI_3S	DOS	0,217813272	0	4	100	4	100
M-PCp	DOS	0,217813272	0	4	100	4	100
M-PCq	DOS	0,217813272	0	4	100	4	100
M-PR	DOS	0,217813272	0	4	100	4	100
M-PL	DOS	0,217813272	0	4	100	4	100
M-PP	DOS	0,217813272	0	4	100	4	100
M-MOJ_24	DOS	0,201111111	0	3	75	3	75
M-MOJ_34	DOS	0,201111111	0	3	75	3	75
Q-SHPq	PEL1	0,166666667	0,00625302	—	—	32	800
	PEL2	0,166666667	0,00625302	—	—	44	1100
	PEL3	0,166666667	0,00625302	—	—	32	800
Q-SHP_100q	PEL1	0,166666667	0,005460407	—	—	32	800
	PEL2	0,166666667	0,005460407	—	—	44	1100
	PEL3	0,166666667	0,005460407	—	—	32	800
Q-SPE_Pq	PEL2	0,166666667	0,005652849	—	—	44	1100
	PEL1	0,166666667	0,003625437	—	—	32	800
	PEL2	0,166666667	0,003625437	—	—	44	1100
Q-SPI_2S	PEL3	0,166666667	0,003625437	—	—	32	800
	PEL1	0,166666667	0,003625437	—	—	44	1100
	PEL2	0,166666667	0,003625437	—	—	32	800

TAREA i	UNIDAD	α_{ij}	β_{ij}	B_{ij}^L		B_{ij}^U	
				TON	BTS	TON	BTS
Q-SPI_3S	PEL1	0,1666666667	0,003885465	--	--	32	800
	PEL2	0,1666666667	0,003885465	--	--	44	1100
	PEL3	0,1666666667	0,003885465	--	--	32	800
Q-PCq	PEL1	0,1666666667	0,008379407	--	--	32	800
	PEL2	0,1666666667	0,008379407	--	--	44	1100
	PEL3	0,1666666667	0,008379407	--	--	32	800
Q-PR	PEL1	0,1666666667	0,0048223	--	--	32	800
	PEL2	0,1666666667	0,0048223	--	--	44	1100
	PEL3	0,1666666667	0,0048223	--	--	32	800
Q-PL	PEL1	0,1666666667	0,005130046	--	--	32	800
	PEL2	0,1666666667	0,005130046	--	--	44	1100
	PEL3	0,1666666667	0,005130046	--	--	32	800
Q-PP	PEL1	0,1666666667	0,003878022	--	--	32	800
	PEL2	0,1666666667	0,003878022	--	--	44	1100
	PEL3	0,1666666667	0,003878022	--	--	32	800
P-SPE_Pp	PEL2	0,1666666667	0,005486527	--	--	44	1100
	PEL1	0,1666666667	0,006810779	--	--	32	800
	PEL2	0,1666666667	0,006810779	--	--	44	1100
P-PCp	PEL2	0,1666666667	0,006810779	--	--	32	800
	PEL3	0,1666666667	0,006810779	--	--	32	800
	PEL1	0,1666666667	0,006810779	--	--	44	1100
MO-MOJ_24	MOL	0,1666666667	0,025	--	--	64	1600
MO-MOJ_34	MOL	0,1666666667	0,0333333333	--	--	64	1600
E-MOJ_24	EXT	0,1666666667	0,021399105	--	--	12	300
E-MOJ_34	EXT	0,1666666667	0,032373143	--	--	12	300

Los autores

Cuadro 13. Parámetros de los materiales

MATERIAL	CAPACIDAD DE ALMACENAMIENTO		INVENTARIO
	<i>m</i>	[Ton]	[Bultos]
1	IL	IL	DP
2	IL	IL	DP
3	IL	IL	DP
4	IL	IL	DP
5	IL	IL	DP
6	IL	IL	DP
7	IL	IL	DP
8	IL	IL	DP
9	IL	IL	DP
10	IL	IL	DP
11	IL	IL	DP
12	IL	IL	0
13	44	1100	0
14	IL	IL	0
15	44	1100	0
16	44	1100	0
17	44	1100	0
18	44	1100	0
19	44	1100	0
20	44	1100	0
21	44	1100	0
22	64	1600	0
23	64	1600	0
24	44	1100	0
25	44	1100	0
26	64	1600	0
27	IL	IL	0
28	IL	IL	0
29	IL	IL	0
30	IL	IL	0
31	IL	IL	0
32	IL	IL	0
33	IL	IL	0
34	IL	IL	0
35	IL	IL	0
36	IL	IL	0
37	IL	IL	0
38	12	300	0
39	12	300	0
40	IL	IL	0
41	IL	IL	0

Los autores

7.3 ETAPA 3: SOLUCIÓN Y VALIDACIÓN DEL MODELO

Para la consecución de esta etapa se tomaron tres casos reales de producción, indicando los productos demandados, las cantidades y el tiempo total de producción. Cada caso se resuelve para ambos objetivos de programación, se realizan las iteraciones necesarias para hallar el número de **slots** adecuado e identifican las soluciones factibles. La mejor solución corresponde a aquella que además de satisfacer la demanda, utilice el menor número de **slots**, evitando redundancia de los mismos.

Los casos en que no existe solución entera factible o el problema sea no factible, se resuelven a través de la *programación lineal entera mixta relajada* (PLEMR) para comparar esta solución con el caso real y concluir acerca de la validación del modelo.

A continuación se muestran los tres casos reales comparados con la solución que arroja el modelo, se concluye en cada uno y se presenta el análisis de la solución de uno de estos.

7.3.1 Validación.

- *Caso 1:*

Cuadro 14. Proceso real para el caso 1

PROCESO REAL			
Material	PRODUCTO	CANTIDAD EMPACADA	
(m)		[Bts]	[Ton]
m35	PR	314	12,56
m12	SHPH	303	12,12
m30	SPE_Pq	1684	67,36
m37	PP	1227	49,08
m29	SPE_Pp	314	12,56
m27	SHPq	1085	43,4
PRODUCCIÓN TOTAL		4927	197,08
TIEMPO HORIZONTE (H)			10 horas

Los autores

Cuadro 15. Solución del modelo para maximización de beneficio, caso 1

MAXIMIZACIÓN DE BENEFICIO			
K	OBJETIVO MIP [Bts]	T. DE SLN	OBSERVACIONES
[hr]		[seg]	
[20, 70]	–	–	No existe solución entera factible
Solución RMIP			
50	–	–	Problema no factible
40	Dual 4100	–	No cumple demanda m29=0
60	Dual 4623	32,5	Satisface la demanda
59	Dual 4625	20,015	Satisface la demanda
58	Dual 4623,2914	24,281	Satisface la demanda
57	–	–	Problema no factible

Los autores

Cuadro 16. Solución del modelo para minimización de **makespan**, caso 1

MINIMIZACIÓN DE MAKESPAN			
K	OBJETIVO MIP	T. DE SLN	OBSERVACIONES
[hr]	[hr]	[seg]	
[20, 70]	–	–	No existe solución entera factible
Solución RMIP			
50	11,33679	8,046	Satisface la demanda
40	–	–	Problema no factible
45	Dual 12,288196	23,796	Satisface la demanda
[46, 49]	–	–	Problema no factible

Los autores

Para ambos objetivos de programación, en este caso no existe una solución entera factible, por lo cual se relaja el problema y obtienen soluciones que dictaminan el grado de valides del modelo. El cuadro 16 muestra dos soluciones que difieren en 1.34 y 2.3 horas respectivamente con el tiempo total de producción del caso real. El modelo se acerca considerablemente a la realidad.

- **Caso 2:**

Cuadro 17. Proceso real para el caso 2

PROCESO REAL			
Material	PRODUCTO	CANTIDAD EMPACADA	
		[Bts]	[Ton]
(m)			
m14	SHP_100h	221	8,84
m29	SPE_Pp	620	24,8
m37	PP	776	31,04
m27	SHPq	1050	42
m12	SHPH	613	24,52
m31	SPI_2S	662	26,48
m32	SPI_3S	815	32,6
PRODUCCIÓN TOTAL		4757	190,28
TIEMPO HORIZONTE (H)			12 horas

Los autores

Cuadro 18. Solución del modelo para maximización de beneficio del caso 2

MAXIMIZACIÓN DE BENEFICIO			
K	OBJETIVO MIP	T. DE SLN	OBSERVACIONES
[hr]	[Bts]	[seg]	
[20, 70]	–	–	No existe solución entera factible
Solución RMIP			
70	5544,821884	17,015	Excedente en m29 de 782 bultos.
60	5544,8219	10,968	No mostró nivel de variable N(m,k)
50	5000	4,201	Excedente en m29 de 94 y en m14 148 bultos
40	–	8,609	Se detuvo por exceder el número de iteraciones, aunque satisface la demanda
39	–	5,828	Se detuvo por exceder el número de iteraciones, aunque satisface la demanda
38	–	–	Problema no factible
37	–	–	Problema no factible

Los autores

Cuadro 19. Solución del modelo para minimización de **makespan** del caso 2

MINIMIZACIÓN DE MAKESPAN			
K	OBJETIVO	T. DE SLN	OBSERVACIONES
[hr]	MIP [hr]	[seg]	
[20, 70]	–	–	No existe solución entera factible
Solución RMIP			
70	10,312	11,718	Satisface la demanda
60	10,312619	10,781	Satisface la demanda
50	11	10,625	Satisface la demanda
40	–	–	Problema no factible
43	10,312	13,828	Satisface la demanda
42	–	–	Problema no factible
44	–	–	Problema no factible
45	–	–	Problema no factible
48	10,312019	9,5	Satisface la demanda
47	–	–	Problema no factible

Los autores

Para este caso no existe una solución entera factible al ser evaluado para los dos objetivos de programación, entonces el problema se relaja y obtienen soluciones que dictaminan el grado de validez del modelo. El cuadro 18 presenta soluciones en las cuales se excede la producción demandada con respecto a la real en el mismo tiempo horizonte. El cuadro 19 permite aseverar la veracidad y la mejora que presenta el modelo en la solución con K=43, difiriendo en el tiempo total de producción de manera positiva en 1.7 horas mejorando la situación real.

- **Caso 3:**

Cuadro 20. Proceso real para el caso 3

PROCESO REAL			
Material	PRODUCTO	CANTIDAD EMPACADA	
(m)		[Bts]	[Ton]
m12	SHP _h	615	24,6
m31	SPI_2S	700	28
m35	PR	102	4,08
m28	SHP_100q	500	20
m14	SHP_100h	500	20
m37	PP	112	4,48
PRODUCCIÓN TOTAL		2529	101,16
TIEMPO HORIZONTE (H)			8 horas

Los autores

Cuadro 21. Solución del modelo para maximización de beneficio del caso 3

MAXIMIZACIÓN DE BENEFICIO			
K	OBJETIVO MIP	T. DE SLN	OBSERVACIONES
[hr]	[Bts]	[seg]	
40	3400	689,75	m12 excede 85 bts, m14 excede 600, m35 excede 98, m37 excede 88 y el resto satisface la demanda
30	2937	114,21	m37 excede 88 bts, m35 excede 36, m14 excede 200 y m12 excede 75 y el resto satisface la demanda
25	–	–	No existe solución entera factible
27	–	–	No existe solución entera factible
28	2800	13,75	m12 excede 85 bts, m35 excede 98, m37 excede 88 y el resto satisface la demanda
29	2717,864002	278,734	m12 excede 85 bts, m14 excede 100, y m37 excede 4 y el resto satisface la demanda

Los autores

Cuadro 22. Solución del modelo para minimización de **makespan** del caso 3

MINIMIZACIÓN DE MAKESPAN			
K	OBJETIVO MIP	T. DE SLN	OBSERVACIONES
[hr]	[hr]	[seg]	
40	7,09337	1000,625	m12 excede 85 bts y el resto satisface la demanda
30	–	–	No existe solución entera factible
27	–	–	No existe solución entera factible
28	–	–	Límite de recursos excedido y no se encontró solución entera factible
29	–	–	Límite de recursos excedido y no se encontró solución entera factible
35	7,0134	1000,468	m12 excede 85 bts y el resto satisface la demanda
34	6,37723	1000,593	m12 excede 85 bts y el resto satisface la demanda
33	8,77276	1000,359	m12 excede 85 bts, m37 excede 88 y el resto satisface la demanda
32	7,402271	1000,281	m12 excede 85 bts y el resto satisface la demanda
31	–	–	No existe solución entera factible

Los autores

El cuadro 21 permite ratificar que el modelo no solo cumple con los requerimientos de este caso, sino también que excede la producción real para algunos productos en el mismo tiempo horizonte de producción. El cuadro 22 confirma la cercanía que presenta la solución del modelo con el caso real para las diferentes soluciones, ya que las mismas se aproximan al tiempo de producción real y la mejor solución para k=32, mejora este tiempo en 0.6 horas.

Es evidente que el modelo matemático ITALCOL 2.2, se acerca de manera propia a la situación productiva real estudiada, manifestando mejoras en la solución de ambos objetivos de programación, ya que algunos casos se produjo mas de lo mínimo establecido en un mismo tiempo horizonte de producción y en otros este tiempo se mejora cumpliendo con la demanda requerida.

A continuación se presenta la solución para el caso 3, indicando la duración de cada **slot**, la asignación y cantidad de material procesado y descargado en cada tarea en cada unidad en determinado **slot**.

7.3.2 Solución.

Figura 32. Cantidad de material (tamaño de lote) involucrado y descargado en cada tarea i en el slot k

italcol 2.2.gms		italcol 2.2.ms.gms		italcol 2.2.lst		
872 VARIABLE BE.L Cantidad que la tarea i descarga a su realizacion en T (k)						
	+	K11	K12	K13	K14	K15
M-PR .DOS					100.000	
M-PP .DOS						100.000
Q-SHP_lq.PEL2						128.436
Q-SPI_2S.PEL1						14.108
872 VARIABLE BE.L Cantidad que la tarea i descarga a su realizacion en T (k)						
	+	K16	K17	K18	K19	K20
M-SHP .DOS				100.000		
M-SHP_1h.DOS		100.000	100.000		100.000	
M-SPI_2S.DOS						100.000
Q-SHP_lq.PEL1					13.204	
Q-SPI_2S.PEL2						14.108
Q-SPI_2S.PEL3		13.204		9.927		
Q-PR .PEL2						
872 VARIABLE BE.L Cantidad que la tarea i descarga a su realizacion en T (k)						
	+	K21	K22	K23	K24	K25
M-SHP .DOS				100.000		
M-SHP_lq.DOS		100.000			100.000	
M-SPI_2S.DOS			100.000			100.000
Q-SPI_2S.PEL2		281.005				
Q-PR .PEL1		166.094				
Q-PP .PEL3		100.000				

Figura 33. Cantidad de material (tamaño de lote) involucrado y descargado en cada tarea i en el **slot** k

The screenshot shows a software window with a toolbar at the top containing icons for file operations, a search icon, a printer icon, and a refresh icon. Below the toolbar, there are three tabs: 'italcol 2.2.gms', 'italcol 2.2_ms.gms', and 'italcol 2.2.lst'. The active tab is 'italcol 2.2.lst', which displays the following table:

872 VARIABLE BE.L		Cantidad que la tarea i descarga a su realizacion en		
		T (k)		
	+	K26	K27	K28
M-SHP	.DOS			
M-SHP	1h.DOS	100.000		100.000
Q-SHP	1q.PEL1	8.767	129.640	
Q-SHP	1q.PEL3		223.791	
Q-SPI	2S.PEL2		337.059	
Q-PR	.PEL1			9.927

Los autores

Se puede observar en las figuras expuestas, la programación de las operaciones para este caso indicando, la duración de cada **slot**, la cantidad que se procesa en cada tarea en la unidad correspondiente y el momento en que es descargada, al final del **slot** k.

La siguiente figura muestra los productos demandados, indicando el material y la cantidad en bultos que corresponde.

Figura 34. Cantidad de los productos demandados en último slot K

872 VARIABLE N.L Balance de material m en T(k)

	K24	K25	K26	K27	K28
+					
m1	1790.000	1790.000	1790.000	1790.000	1790.000
m2	1310.000	1280.000	1250.000	1220.000	1220.000
m3	1720.000	1720.000	1720.000	1720.000	1720.000
m4	1580.000	1550.000	1520.000	1490.000	1490.000
m5	1440.000	1400.000	1360.000	1320.000	1320.000
m8	1880.000	1880.000	1880.000	1880.000	1880.000
m9	1940.000	1940.000	1940.000	1940.000	1940.000
m10	1920.000	1920.000	1920.000	1920.000	1920.000
m11	1920.000	1920.000	1920.000	1920.000	1920.000
m12	500.000	500.000	500.000	600.000	700.000
m14	400.000	400.000	500.000	500.000	500.000
m15	262.198	353.431			
m18	337.059	337.059			
m22	9.927	9.927	9.927		
m28	137.802	137.802	146.569	500.000	500.000
m31	362.941	362.941	362.941	700.000	700.000
m35	190.073	190.073	190.073	190.073	200.000
m37	200.000	200.000	200.000	200.000	200.000

Los autores

8. CONCLUSIONES

- El modelo matemático para programar la producción en ITALCOL S.C.A. Bucaramanga, obtuvo soluciones factibles, válidas y es una buena representación del sistema productivo como herramienta cuantitativa sólida, rápida y confiable, que soporta el proceso de decisión para programar las operaciones, aislando el componente subjetivo de los encargados.
- El éxito del desarrollo de la metodología se logró por el grado y calidad de competencias adquiridas en cuanto a la vinculación y conocimiento de la dinámica del sistema de producción, generando un diagnóstico que permitió analizar todos sus componentes, procesos, variables y relaciones que definieron el marco de acción del modelo matemático.
- La formulación matemática desarrollada por Arul Sundaramoorthy e I.A. Karimi, se estudió, evaluó y desarrolló de manera satisfactoria, con excelentes resultados como herramienta eficiente para programar producción a corto plazo en organizaciones con configuración continua y lotes multipropósito.
- La construcción de un lenguaje propio de modelado, generó un bloque de variables y ecuaciones generalizado en lenguaje de programación GAMS para la consecución y aplicación de la formulación estudiada a diferentes procesos productivos con las mismas características
- El diseño y construcción de una interfase gráfica, permite al usuario alimentar y modificar los parámetros necesarios dentro del modelo, para realizar una corrida de producción en cualquier momento, obtener solución y tomar decisiones al respecto.

9. RECOMENDACIONES

- En busca de ventajas competitivas, las empresas productivas deben apoyarse en herramientas matemáticas e informáticas, que les permitan establecer un plan de producción de manera más rápida, exacta y confiable y así, puedan dedicar más tiempo y esfuerzo a otras actividades que lo requieran.
- La profundización en la academia acerca de lenguajes de modelado y herramientas informáticas que motiven y faciliten el trabajo del investigador.
- Es aconsejable que la Escuela de Estudios Industriales y Empresariales de continuidad e incentive de manera más activa la investigación en la programación de operaciones, para lograr un trabajo completo que integre alcances diferentes y a su vez que sirva de apoyo a la competitividad de la industria en la región.

BIBLIOGRAFÍA

ADAM E., Everett, Jr y EBERT J., Ronald. Administración de la producción y las operaciones. 4 ed. México: PRENTICE HALL, 1981. p. 144-148.

ARBELAEZ DE MONCALEANO, Ruby. Metodología de la Investigación. 1 ed. Bucaramanga: CEDEUIS. 1992.

BAKER, Kenneth R. Introduction to sequencing and scheduling. USA: John Wiley & Sons, 1974.

BUFFA, Elwood S. and SARIN, Rakesh K. Administración de la producción y de las operaciones. México: Limusa, 1996.

CASTRO, P; BARBOSA-PÓVOA, A. P. F. D. and MATOS, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 40, No. 9 (mar., 2001); p. 2059-2068. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

GAVETT, William J. Production and operations management. USA: Harcourt, Brace & World, 1968. p. 8-13.

GIANNELOS, Nikolaos F. and GEORGIADIS, Michael C. A simple new continuous-time formulation for short-term scheduling of multipurpose batch processes. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 41, No. 9 (abr., 2001); p. 2178-2184. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

HERNANDEZ SAMPIERI, Roberto; FERNANDEZ COLLADO, Carlos y BAPTISTA LUCIO, Pilar. Metodología de la Investigación. 3 ed. México: MCGRAW-HILL, 2003.

IBARRA MIRÓN, Santiago. Configuraciones productivas: conceptos y tipologías fundamentales. [en línea] Disponible en <<http://www.monografias.com/trabajos16/configuraciones-productivas/configuraciones-productivas.shtml>> [citado 26 de julio de 2006]

IERAPETRITOU, M. G. and FLOUDAS C. A. Effective continuous-time formulation for short-term scheduling: multipurpose batch plant. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 37, No. 11 (oct., 1998); p. 4341-4359. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

ITALCOL S.A. Historia. [en línea] Disponible en <<http://www.italcol.com/corporativa.htm>> [citado 27 de diciembre de 2006].

LINARES, Pedro et al. Modelos matemáticos de optimización [en línea]. (octubre 2001). Disponible en <www.gams.com/docs/contributed/modelado_en_gams.pdf> [citado 29 de julio de 2006]

LUCENA, Abilio y BEASLEY John E. Branch and cut algorithms. [en línea] Disponible en <people.brunel.ac.uk/~mastjjb/jeb/book/chapter5.pdf> [citado 27 agosto de 2006]

MALDONADO, Carolina. Planificación, programación y control de las operaciones. [en línea] Disponible en <<http://www.monografias.com/trabajos18/direccion-operaciones/direccion-operaciones.shtml>> [citado 26 de julio de 2006]

MITCHELL, John E. Branch and cut algorithms for combinatorial optimization problems. [en línea] Disponible en <www.rpi.edu/~mitchj/papers/bc_hao.ps> [citado 27 agosto de 2006]

MARAVELIAS, Christos T and GROSSMANN, Ignacio E. A new general continuous-time state task network formulation for short-term scheduling of multipurpose batch plants. En: Industrial and Engineering Chemistry Research [base de datos en línea]. Vol. 42 (mar., 2003); p. 3056-3074. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

MOSKOWITZ, Herbert y WRIGHT P., Gordon. Investigación de Operaciones. 3 Ed. México: PRENTICE HALL, 1996. Págs. 13, 21-22, 259-266, 463-472.

NIÑO LÓPEZ, Myriam Leonor. Modelamiento a través de la programación lineal entera mixta del problema de programación de operaciones en el taller de trabajo. En: Revista Ingenierías. N° 2 (may., 2003); pags. 9-17.

Programación lineal entera-mixta [en línea]. Disponible en <departamentos.unican.es/macc/personal/profesores/castillo/Libro/Chap2.pdf> [citado 29 de julio de 2006]

SUNDARAMOORTHY, Arul and KARIMI, I.A. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. En: Chemical Engineering Science [base de datos en línea]. Vol. 60 (may., 2005); p. 2679-2702. [citado 12 de marzo de 2006]. Disponible en SCOPUS Research Databases.

ANEXO A
HIPERVÍNCULOS DE ACCESO ELECTRÓNICO A ARCHIVOS
SOPORTE

X: Unidad de disco establecida

X:\Anexos electrónicos\Arul y Karimi_eje1a.gms

X:\Anexos electrónicos\Arul y Karimi_eje3b.gms

X:\Anexos electrónicos\Arul y Karimi_eje1a_MS.gms

X:\Anexos electrónicos\Demanda.xls

X:\Formatos

X:\Anexos electrónicos\Tablas Tiempos y Parametros.xls

X:\Anexos electrónicos\ITALCOL 1.1.gms

X:\Anexos electrónicos\ITALCOL 2.1.gms

X:\Anexos electrónicos\INTERFAZ 2.2.xls

X:\Anexos electrónicos\ITALCOL 2.2.gms

X:\Anexos electrónicos\ITALCOL 2.2_MS.gms


```

arul y karimi_eje3a.gms
H2 100
R1 100 150
R2 100 150
R3 100 150
S 300
M 200 200

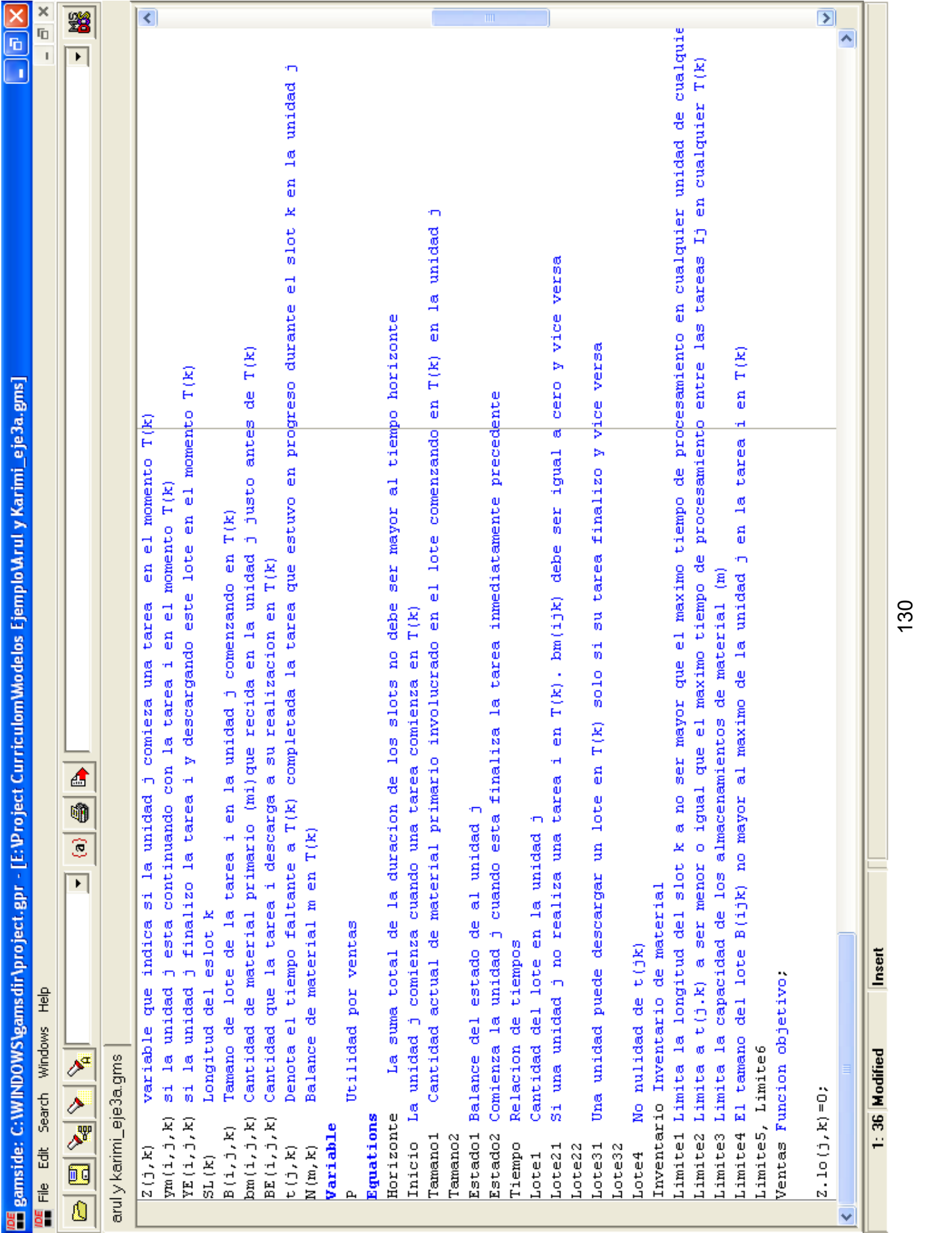
Table B1(i,j) Cantidad minima de la unidad j con la tarea i
MR1 MR2
M 20 20

Table PCM(i,m) Porcentaje de transferencia hacia la tarea
m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13
H1 1
H2 0.25 0.75
R1 1
R2 0.5 0.5
R3
S 1 0.4
M 0.40 0.2

Table PPM(i,m) Porcentaje de transferencia desde la tarea
m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13
H1 1
H2
R1 1
R2 1
R3 0.1 0.4 0.5
S 0.6
M 1 0.4

Binary variables
Y(i,j,k) variable que indica si la unidad j comienza la tarea i en el momento T(k)
Positive variables

```



aruly_karimi_eje3a.gms

Z(j,k) variable que indica si la unidad j comienza una tarea en el momento T(k)
 ym(i,j,k) si la unidad j esta continuando con la tarea i en el momento T(k)
 YE(i,j,k) si la unidad j finalizo la tarea i y descargando este lote en el momento T(k)
 SL(k) Longitud del eslot k
 B(i,j,k) Tamano de lote de la tarea i en la unidad j comenzando en T(k)
 bm(i,j,k) Cantidad de material primario (mi) que recida en la unidad j justo antes de T(k)
 BE(i,j,k) Cantidad que la tarea i descarga a su realizacion en T(k)
 t(j,k) Denota el tiempo faltante a T(k) completada la tarea que estuvo en progreso durante el slot k en la unidad j
 N(m,k) Balance de material m en T(k)

Variable

p Utilidad por ventas

Equations

Horizonte La suma total de la duracion de los slots no debe ser mayor al tiempo horizonte

Inicio La unidad j comienza cuando una tarea comienza en T(k)

Tamano1 Cantidad actual de material primario involucrado en el lote comenzando en T(k) en la unidad j

Tamano2

Estado1 Balance del estado de al unidad j

Estado2 Comienza la unidad j cuando esta finaliza la tarea inmediatamente precedente

Tiempo Relacion de tiempos

Lote1 Cantidad del lote en la unidad j

Lote21 Si una unidad j no realiza una tarea i en T(k). bm(ijk) debe ser igual a cero y vice versa

Lote22

Lote31 Una unidad puede descargar un lote en T(k) solo si su tarea finalizo y vice versa

Lote32

Lote4 No nulidad de t(jk)

Inventario Inventario de material

Limite1 Limita la longitud del slot k a no ser mayor que el maximo tiempo de procesamiento en cualquier unidad de cualquier

Limite2 Limita a t(j.k) a ser menor o igual que el maximo tiempo de procesamiento entre las tareas Ij en cualquier T(k)

Limite3 Limita la capacidad de los almacenamientos de material (m)

Limite4 El tamano del lote B(ijk) no mayor al maximo de la unidad j en la tarea i en T(k)

Limite5, Limite6

Ventas Funcion objetivo;

Z.lo(j,k)=0;

MS Word icon, search icon, save icon, print icon, refresh icon, undo icon, redo icon, zoom icon, status bar (a)

arul y karimi_eje3a.gms

```

Z.up(j,k)=1;
ym.lo(i,j,k)=0;
ym.up(i,j,k)=1;
YE.lo(i,j,k)=0;
YE.up(i,j,k)=1;
SL.up("k0")=0;
Z.up(j,k){ord(k)=card(k)}=0;
Y.up(i,j,k){ord(k)=card(k)}=0;
B.up(i,j,k){ord(k)=card(k)}=0;
ym.up(i,j,"k0")=0;
ym.up(i,j,k){ord(k)=card(k)}=0;
YE.up(i,j,"k0")=0;
t.up(j,"k0")=0;
t.up(j,k){ord(k)=card(k)}=0;
bm.up(i,j,"k0")=0;
bm.up(i,j,k){ord(k)=card(k)}=0;
BE.up(i,j,"k0")=0;

Horizonte.. Sum(k$(ord(k)>1), SL(k)) =1= H;

Inicio(j,k){ord(k)<card(k)}..
Z(j,k) =e= Sum(i$JI(j,i), Y(i,j,k));

Tamano1(i,j,k){JI(j,i)and ord(i)>1and ord(k)<card(k)}..
B(i,j,k) =1= Bu(i,j)*Y(i,j,k);

Tamano2(i,j,k){ord(i)=8 and JI(j,i)and ord(k)<card(k)}..
B(i,j,k) =g= Bl(i,j)*Y(i,j,k);

Estado1(i,j,k){ord(k)>1 and JI(j,i)and ord(k)<card(k)}..
ym(i,j,k) =e= ym(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k);

Estado2(j,k){ord(k)>1 and ord(k)<card(k)}..
Z(j,k) =e= sum(i$JI(j,i), YE(i,j,k));
    
```

```

aruly_karimi_eje3a.gms
Tiempo(j,k)$ (ord(k)<card(k))..
t(j,k+1) =g= t(j,k) + sum(i$(JI(j,i)and ord(i)>1), c(i,j)*v(i,j,k) + v(i,j)*B(i,j,k) - SL(k+1));

Lote1(i,j,k)$ (ord(k)>1 and JI(j,i)and ord(i)>1)..
bm(i,j,k) =e= bm(i,j,k-1) + B(i,j,k-1) - BE(i,j,k);

Lote21(i,j,k)$ (ord(i)=8 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
bm(i,j,k) =g= B1(i,j)*ym(i,j,k);

Lote22(i,j,k)$ (ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
bm(i,j,k) =l= Bu(i,j)*ym(i,j,k);

Lote31(i,j,k)$ (ord(i)=8 and ord(k)>1 and ord(k)<card(k)and JI(j,i))..
BE(i,j,k) =g= B1(i,j)*YE(i,j,k);

Lote32(i,j,k)$ (ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
BE(i,j,k) =l= Bu(i,j)*YE(i,j,k);

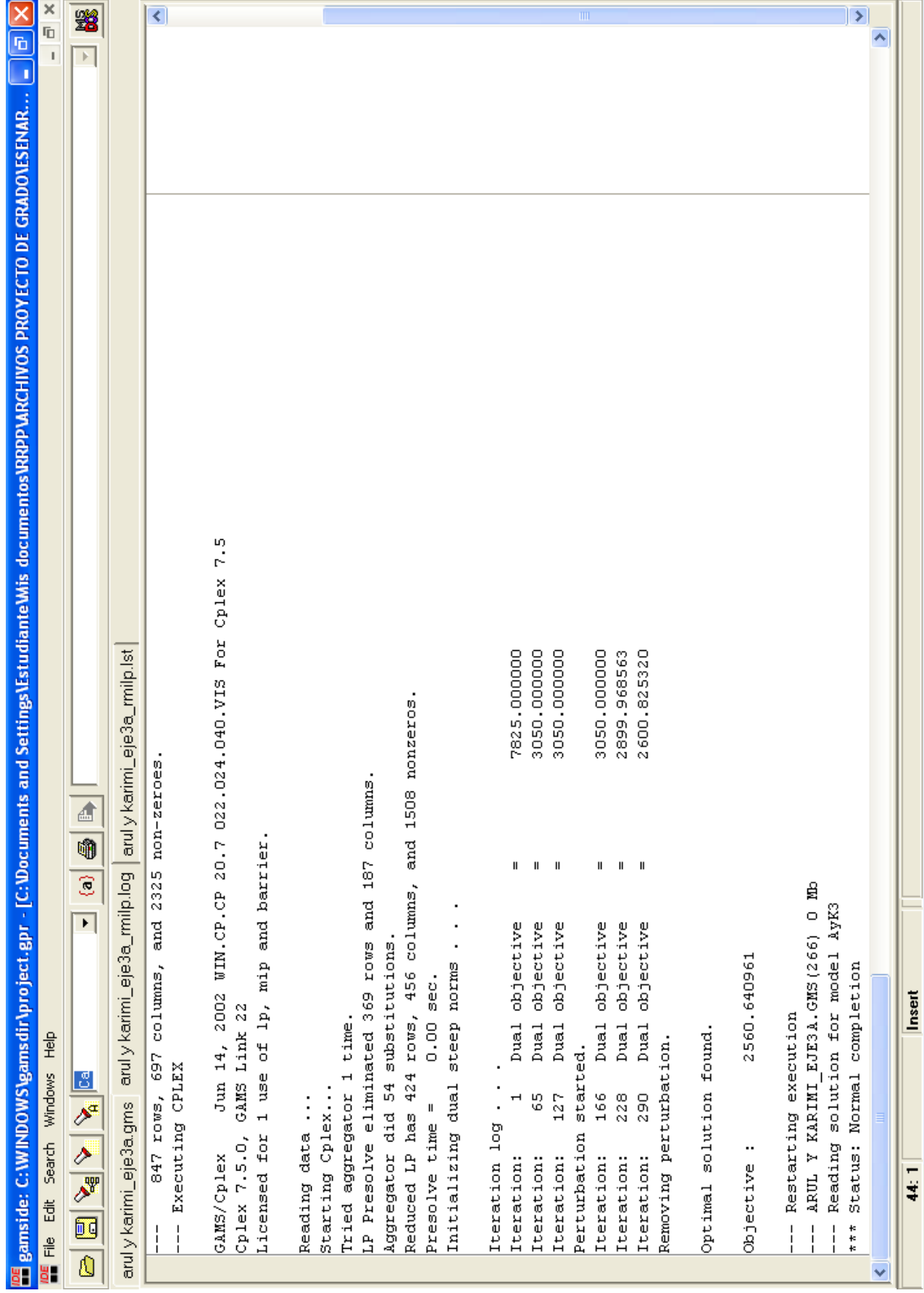
Lote4(j,k)$ (ord(k)>1 and ord(k)<card(k))..
t(j,k) =l= sum(i$(JI(j,i)and ord(i)>1), c(i,j)*ym(i,j,k) + v(i,j)*bm(i,j,k));

Inventario(m,k)..
N(m,k) =e= Ii(m)$ (ord(k)=1) + N(m,k-1) + sum(i$(PM(i,m), sum(j$(JI(j,i), PPM(i,m)*BE(i,j,k))
- sum(i$(CM(i,m), sum(j$(JI(j,i), PCM(i,m)*B(i,j,k))));

Limite1(k).. SL(k) =l= smax(j, smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j)));
Limite2(j,k).. t(j,k) =l= smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j));
Limite3(m,k)$ (ord(m)>2and ord(m)<11and ord(m)<> 8).. N(m,k) =l= Ca(m);
Limite4(i,j,k)$ (JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. B(i,j,k) =l= Bu(i,j);
Limite5(i,j,k)$ (JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. bm(i,j,k) =l= Bu(i,j);
Limite6(i,j,k)$ (JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. BE(i,j,k) =l= Bu(i,j);
Ventas(K)$ (ord(k)=card(k)).. P =e= Sum(m$(ord(m)>11), R(m)*N(m,k));
Model AyK3 /all/;
Solve AyK3 using rmp maximazing P;

```

EJEMPLO 3a Segunda Parte



```
--- 847 rows, 697 columns, and 2325 non-zeros.
--- Executing CPLEX
GAMS/Cplex Jun 14, 2002 WIN.CP.CP 20.7 022.024.040.VIS For Cplex 7.5
Cplex 7.5.0, GAMS Link 22
Licensed for 1 use of lp, mip and barrier.
Reading data ...
Starting Cplex...
Tried aggregator 1 time.
LP Presolve eliminated 369 rows and 187 columns.
Aggregator did 54 substitutions.
Reduced LP has 424 rows, 456 columns, and 1508 nonzeros.
Presolve time = 0.00 sec.
Initializing dual steep norms . . .
Iteration log . . .
Iteration: 1 Dual objective = 7825.000000
Iteration: 65 Dual objective = 3050.000000
Iteration: 127 Dual objective = 3050.000000
Perturbation started.
Iteration: 166 Dual objective = 3050.000000
Iteration: 228 Dual objective = 2899.968563
Iteration: 290 Dual objective = 2600.825320
Removing perturbation.
Optimal solution found.
Objective : 2560.640961
--- Restarting execution
--- ARUL Y KARIMI_EJE3A.GMS (266) 0 Mb
--- Reading solution for model AyK3
*** Status: Normal completion
```

EJEMPLO 3a Tercera Parte

The screenshot shows a GAMS software window titled "gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\Documents and Settings\Estudiante\Mis documentos\RRPP\ARCHIVOS PROYECTO DE GRADO\ESENAR...". The window contains the following text:

```
arulykarimi_eje3a.gms | arulykarimi_eje3a_rmilp.log | arulykarimi_eje3a_rmilp.lst

COMPILATION TIME = 0.015 SECONDS 0.9 Mb WIN207-133
DGAMS Rev 133 Windows NT/95/98 01/17/07 13:05:47 Page 2
General Algebraic Modeling System
Model Statistics SOLVE AYK3 Using RMIP From line 266

MODEL STATISTICS
BLOCKS OF EQUATIONS 21 SINGLE EQUATIONS 847
BLOCKS OF VARIABLES 11 SINGLE VARIABLES 697
NON ZERO ELEMENTS 2325 DISCRETE VARIABLES 102

GENERATION TIME = 0.016 SECONDS 1.6 Mb WIN207-133

EXECUTION TIME = 0.016 SECONDS 1.6 Mb WIN207-133
DGAMS Rev 133 Windows NT/95/98 01/17/07 13:05:47 Page 3
General Algebraic Modeling System
Solution Report SOLVE AYK3 Using RMIP From line 266

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

MODEL AYK3 OBJECTIVE P
TYPE RMIP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 266
```

The window also shows a taskbar with icons for File, Edit, Search, Windows, and Help, and a status bar at the bottom with "5: 1" and "Insert" buttons.

gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\Documents and Settings\Estudiante\Mis documentos\RRPP\ARCHIVOS PROYECTO DE GRADO\ESENAR...

File Edit Search Windows Help

aruly karimi_eje3a.gms | aruly karimi_eje3a_rmlp.log | aruly karimi_eje3a_rmlp.lst

```

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE    2560.6410

RESOURCE USAGE, LIMIT    0.062    1000.000
ITERATION COUNT, LIMIT  351      10000

GAMS/Cplex   Jun 14, 2002 WIN.CP.CP 20.7 022.024.040.VIS For Cplex 7.5
Cplex 7.5.0, GAMS Link 22
Licensed for 1 use of lp, mip and barrier.

Optimal solution found.

Objective :      2560.640961

**** REPORT SUMMARY :   0  NONOPT
                       0  INFEASIBLE
                       0  UNBOUNDED

EXECUTION TIME   =   0.016 SECONDS   0.8 Mb   WIN207-133

USER: Danilo Abril Hernandez           G021129:1442AB-WIN
      Universidad Industrial de Santander, Escuela de Estudios IndC4015

**** FILE SUMMARY

INPUT  C:\DOCUMENTS AND SETTINGS\ESTUDIANTE\MIS DOCUMENTOS\RRPP\ARCHIVOS PRO
       YECTO DE GRADO\ARUL Y KARIMI_EJE3A.GMS
OUTPUT C:\WINDOWS\GAMSDIR\ARUL Y KARIMI_EJE3A.LST

```

71:1 | Insert


```

arul y karimi_eje3b_ms.gms
R3      0.01333 0.00889      0.00667
S
M      0.00667 0.00667

Table Bu(i,j) Capacidad de la unidad j con la tarea i
      HR  RR1  RR2  SR  MR1  MR2
H1    100
H2    100
R1    100  150
R2    100  150
R3    100  150
S      300
M      200  200

Table Bl(i,j) Cantidad minima de la unidad j con la tarea i
      MR1  MR2
M      20   20

Table PCM(i,m) Porcentaje de transferencia hacia la tarea
      m1  m2  m3  m4  m5  m6  m7  m8  m9  m10  m11
H1     1
H2
R1     1
R2     0.5  0.5
R3
S
M      0.4  0.2

Table PPM(i,m) Porcentaje de transferencia desde la tarea
      m1  m2  m3  m4  m5  m6  m7  m8  m9  m10  m11  m12  m13
H1
H2
R1     1
R2     1
  
```

```

R3
S
M
0.1 0.4 0.5 0.4 0.6 0.4
1

Binary variables
Y(i,j,k) variable que indica si la unidad j comienza la tarea i en el momento T(k)
Positive variables
Z(j,k) variable que indica si la unidad j comienza una tarea en el momento T(k)
ym(i,j,k) si la unidad j esta continuando con la tarea i en el momento T(k)
YE(i,j,k) si la unidad j finalizo la tarea i y descargando este lote en el momento T(k)
SL(k) Longitud del eslot k
B(i,j,k) Tamano de lote de la tarea i en la unidad j comenzando en T(k)
bm(i,j,k) Cantidad de material primario (mi) que recida en la unidad j justo antes de T(k)
BE(i,j,k) Cantidad que la tarea i descarga a su realizacion en T(k)
t(j,k) Denota el tiempo faltante a T(k) completada la tarea que estuvo en progreso durante el slot k en la unidad j
N(m,k) Balance de material m en T(k)
Variable
MS Time de produccion
Equations
MakeSpan La suma total de la duracion de los slots no debe ser mayor al tiempo de produccion
Inicio La unidad j comienza cuando una tarea comienza en T(k)
Tamano1 Cantidad actual de material primario involucrado en el lote comenzando en T(k) en la unidad j
Tamano2
Estado1 Balance del estado de al unidad j
Estado2 Comienza la unidad j cuando esta finaliza la tarea inmediatamente precedente
Tiempo Relacion de tiempos
Lote1 Cantidad del lote en la unidad j
Lote21 Si una unidad j no realiza una tarea i en T(k). bm(ijk) debe ser igual a cero y vice versa
Lote22
Lote31 Una unidad puede descargar un lote en T(k) solo si su tarea finalizo y vice versa
Lote32
Lote4 No nulidad de t(jk)
Inventario Inventario de material
Limite1 Limita la longitud del slot k a no ser mayor que el maximo tiempo de procesamiento en cualquier unidad de cualquier

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [E:\Project Curriculum\Modelos Ejemplo\Arul y Karimi_eje3b_MS.gms]

File Edit Search Windows Help

aruly karimi_eje3b_ms.gms

```

Limite2 Limita a t(j,k) a ser menor o igual que el maximo tiempo de procesamiento entre las tareas Ij en cualquier T(k)
Limite3 Limita la capacidad de los almacenamientos de material (m)
Limite4 El tamano del lote B(ijk) no mayor al maximo de la unidad j en la tarea i en T(k)
Limite5, Limite6
Demanda Cumpimiento de la demanda;

Z.lo(j,k)=0;
Z.up(j,k)=1;
ym.lo(i,j,k)=0;
ym.up(i,j,k)=1;
YE.lo(i,j,k)=0;
YE.up(i,j,k)=1;
SL.up("k0")=0;
Z.up(j,k)$(ord(k)=card(k))=0;
Y.up(i,j,k)$(ord(k)=card(k))=0;
B.up(i,j,k)$(ord(k)=card(k))=0;
ym.up(i,j,"k0")=0;
ym.up(i,j,k)$(ord(k)=card(k))=0;
YE.up(i,j,"k0")=0;
t.up(j,"k0")=0;
t.up(j,k)$(ord(k)=card(k))=0;
bm.up(i,j,"k0")=0;
bm.up(i,j,k)$(ord(k)=card(k))=0;
BE.up(i,j,"k0")=0;

MakeSpan.. Sum(k$(ord(k)>1), SL(k)) =e= MS;

Inicio(j,k)$(ord(k)<card(k))..
Z(j,k) =e= Sum(i$JI(j,i), Y(i,j,k));

Tamano1(i,j,k)$(JI(j,i)and ord(i)>1and ord(k)<card(k))..
B(i,j,k) =1= Bu(i,j)*Y(i,j,k);

Tamano2(i,j,k)$(ord(i)=8 and JI(j,i)and ord(k)<card(k))..

```

185: 1 Modified Insert

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [E:\Project Curriculum\Modelos Ejemplo\Arul y Karimi_eje3b_MS.gms]
IDE File Edit Search Windows Help
[Icons] [a] [Print] [Run]
aruly karimi_eje3b_ms.gms
B(i,j,k) =g= Bl(i,j)*Y(i,j,k);
Estado1(i,j,k){(ord(k)>1 and ord(k)<card(k))..
ym(i,j,k) =e= ym(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k);
Estado2(j,k){(ord(k)>1 and ord(k)<card(k))..
Z(j,k) =e= sum(i$(JI(j,i), YE(i,j,k)));
Tiempo(j,k){(ord(k)<card(k))..
t(j,k+1) =g= t(j,k) + sum(i$(JI(j,i)and ord(i)>1), c(i,j)*Y(i,j,k) + v(i,j)*B(i,j,k) - SL(k+1));
Lote1(i,j,k){(ord(k)>1 and JI(j,i)and ord(i)>1)..
bm(i,j,k) =e= bm(i,j,k-1) + B(i,j,k-1) - BE(i,j,k);
Lote21(i,j,k){(ord(i)=8 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
bm(i,j,k) =g= Bl(i,j)*ym(i,j,k);
Lote22(i,j,k){(ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
bm(i,j,k) =l= Bu(i,j)*ym(i,j,k);
Lote31(i,j,k){(ord(i)=8 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
BE(i,j,k) =g= Bl(i,j)*YE(i,j,k);
Lote32(i,j,k){(ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
BE(i,j,k) =l= Bu(i,j)*YE(i,j,k);
Lote4(j,k){(ord(k)>1 and ord(k)<card(k))..
t(j,k) =l= sum(i$(JI(j,i)and ord(i)>1), c(i,j)*ym(i,j,k) + v(i,j)*bm(i,j,k));
Inventario(m,k)..
N(m,k) =e= Ii(m){(ord(k)=1) + N(m,k-1) + sum(i$(PPW(i,m), sum(j$(JI(j,i), PPM(i,m)*BE(i,j,k))
- sum(i$(CM(i,m), sum(j$(JI(j,i), PCM(i,m)*B(i,j,k)));
Limite1(k).. SL(k) =l= smax(j, smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j)));

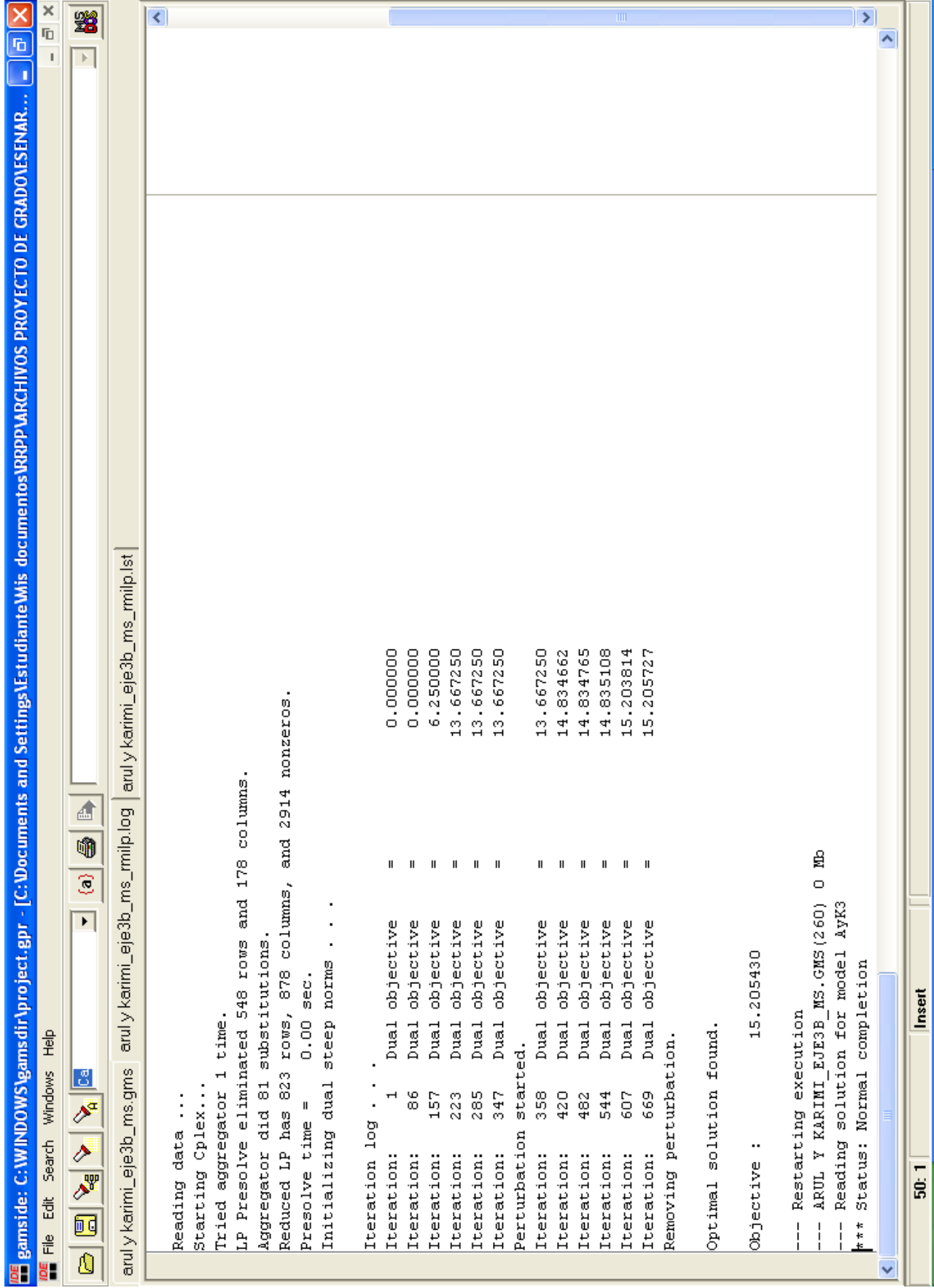
```

```

IDE gamside: C:\WINDOWS\gamsdir\project.gpr - [E:\Project Curriculum\Modelos Ejemplo\Arul y Karimi_eje3b_MS.gms]
File Edit Search Windows Help
aruly karimi_eje3b_ms.gms
Limite2(j,k).. t(j,k) =l= smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j));
Limite3(m,k){ord(m)>2and ord(m)<11and ord(m)<> 8}.. N(m,k) =l= Ca(m);
Limite4(i,j,k){(JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. B(i,j,k) =l= Bu(i,j);
Limite5(i,j,k){(JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. km(i,j,k) =l= Bu(i,j);
Limite6(i,j,k){(JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. BE(i,j,k) =l= Bu(i,j);
Demanda(m,k){ord(k)=card(k)and ord(m)>11}.. N(m,k) =g= D(m);
Model AyK3 /all/;
Solve AyK3 using Rmip minimizing MS;

```

EJEMPLO 3b Segunda Parte



EJEMPLO 3b Tercera Parte

The screenshot shows a GAMS window titled 'gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\Documents and Settings\Estudiante\Mis documentos\RRPP\ARCHIVOS PROYECTO DE GRADO\ESENAR...'. The window contains the following text:

```
arulykarimi_eje3b_ms.gms | arulykarimi_eje3b_ms_rmipb.log | arulykarimi_eje3b_ms_rmipb.lst  
  
COMPIATION TIME = 0.015 SECONDS 0.9 Mb WIN207-133  
DGAMS Rev 133 Windows NT/95/98 01/17/07 13:26:12 Page 2  
General Algebraic Modeling System  
Model Statistics SOLVE AyK3 Using RMIP From line 260  
  
MODEL STATISTICS  
BLOCKS OF EQUATIONS 21 SINGLE EQUATIONS 1452  
BLOCKS OF VARIABLES 11 SINGLE VARIABLES 1137  
NON ZERO ELEMENTS 3969 DISCRETE VARIABLES 170  
  
GENERATION TIME = 0.016 SECONDS 1.6 Mb WIN207-133  
  
EXECUTION TIME = 0.016 SECONDS 1.6 Mb WIN207-133  
DGAMS Rev 133 Windows NT/95/98 01/17/07 13:26:12 Page 3  
General Algebraic Modeling System  
Solution Report SOLVE AyK3 Using RMIP From line 260  
  
S O L V E S U M M A R Y  
  
MODEL AyK3 OBJECTIVE MS  
TYPE RMIP DIRECTION MINIMIZE  
SOLVER CPLEX FROM LINE 260  
  
**** SOLVER STATUS 1 NORMAL COMPLETION
```

The status bar at the bottom of the window shows '5: 1' and 'Insert'.

gamside: C:\WINDOWS\gamsdir\project.gpr - [C:\Documents and Settings\Estudiante\Mis documentos\RRPP\ARCHIVOS PROYECTO DE GRADO\ESENAR...

File Edit Search Windows Help

aruly karimi_eje3b_ms.gms | aruly karimi_eje3b_ms_rmilp.log | aruly karimi_eje3b_ms_rmilp.lst

```

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE    15.2054

RESOURCE USAGE, LIMIT   0.109    1000.000
ITERATION COUNT, LIMIT  675      10000

GAMS/Cplex   Jun 14, 2002 WIN.CP.CP 20.7 022.024.040.VIS For Cplex 7.5
Cplex 7.5.0, GAMS Link 22
Licensed for 1 use of lp, mip and barrier.

Optimal solution found.

Objective :      15.205430

**** REPORT SUMMARY :   0  NONOPT
                       0  INFEASIBLE
                       0  UNBOUNDED

EXECUTION TIME   =   0.015 SECONDS   0.8 Mb   WIN207-133

USER: Danilo Abril Hernandez           G021129:1442AB-WIN
      Universidad Industrial de Santander, Escuela de Estudios IndC4015

**** FILE SUMMARY

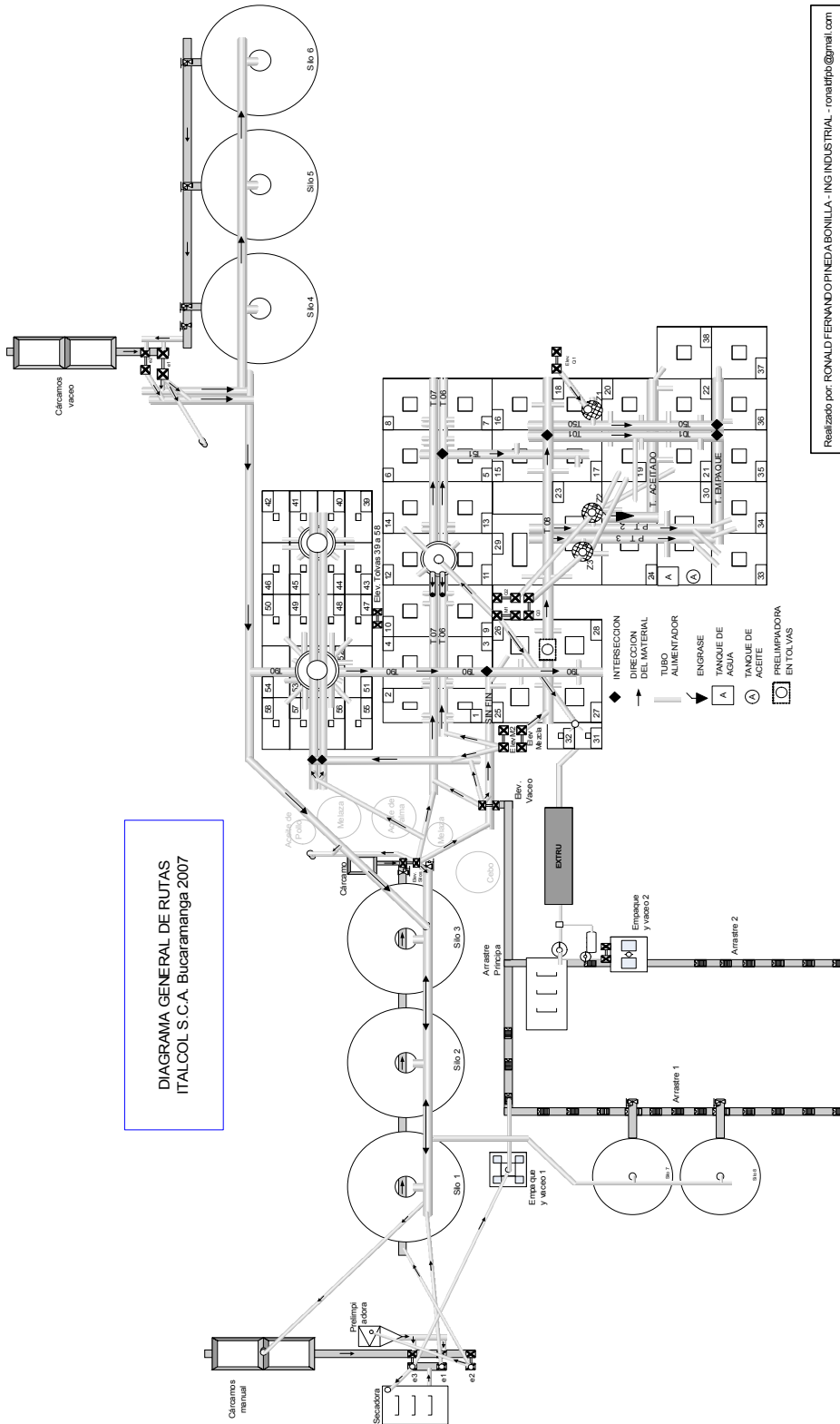
INPUT   C:\DOCUMENTS AND SETTINGS\ESTUDIANTE\MIS DOCUMENTOS\RRPP\ARCHIVOS PRO
        YECTO DE GRADO\ARUL Y KARIMI_EJE3B_MS.GMS
OUTPUT  C:\WINDOWS\GAMSDIR\ARUL Y KARIMI_EJE3B_MS.LST

```

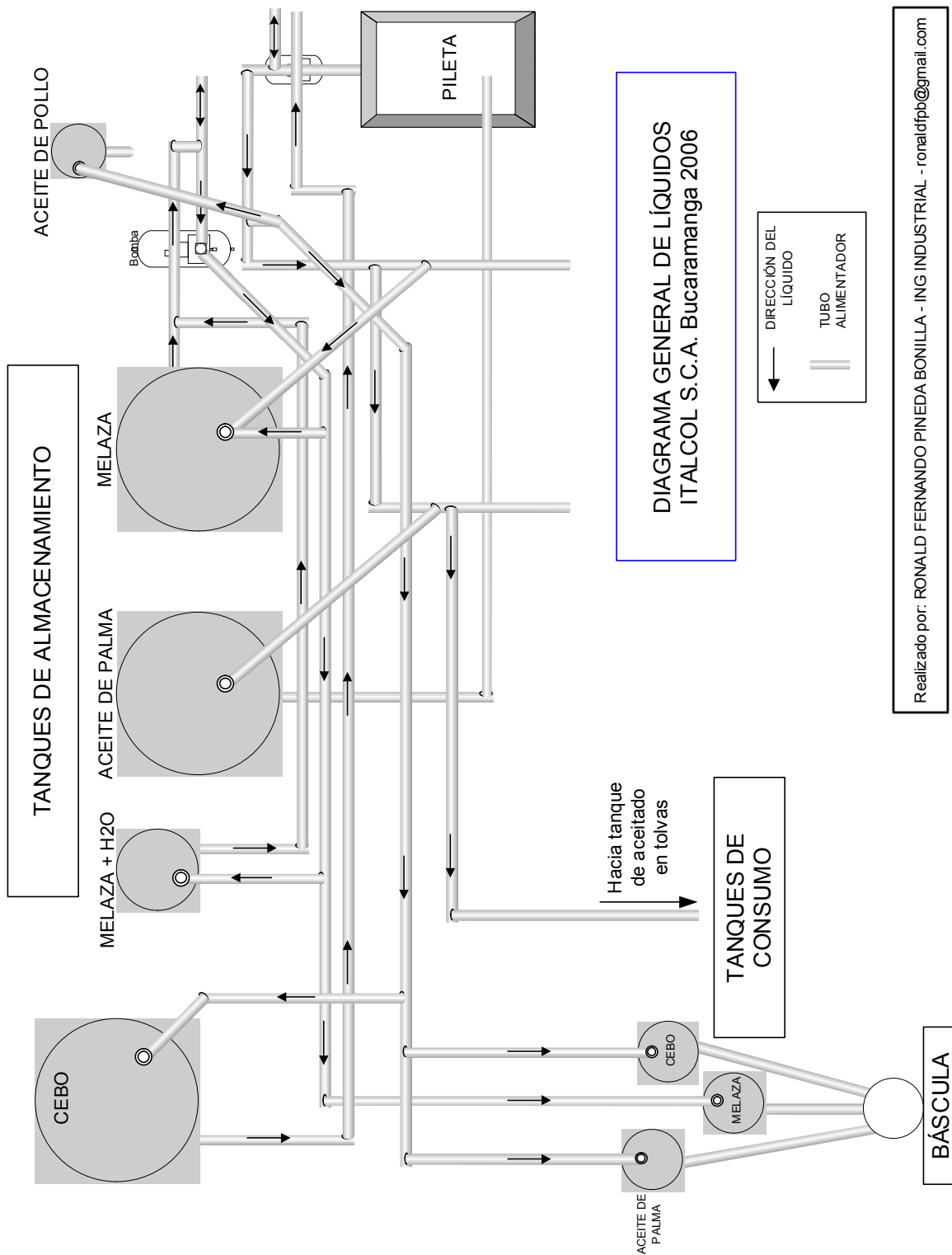
71: 1 | Insert

ANEXO E

DIAGRAMA GENERAL DE RUTAS ITALCOL S.C.A. Bucaramanga



ANEXO F
 DIAGRAMA GENERAL DE LÍQUIDOS ITALCOL S.C.A. Bucaramanga




```

italcol 1.1.gms

Q-SPI_3S
M-PCp      100
M-PCq      100
P-PCp
Q-PCq
M-MOJ_24   100
M-MOJ_34   100
MO-MOJ_24
MO-MOJ_34
E-MOJ_24
E-MOJ_34
S-MOJ_24
S-MOJ_34

Table PCM(i,m) Porcentaje de transferencia hacia la tarea material consumido por la tera
      m1  m2  m3  m4  m5  m6  m19  m20  m21  m26  m27
M-SHPH  0.3  0.3  0.4
M-SHPq   0.3  0.3  0.4
M-SPE_Pp  0.3  0.3  0.3  0.4
M-SPE_Pq  0.3  0.3  0.3  0.4
M-SPI_1s  0.3  0.3  0.4
M-SPI_2s  0.3  0.3  0.4
M-SPI_3s  0.3  0.3  0.4
M-PCp    0.3  0.3  0.3  0.4
M-PCq    0.3  0.3  0.3  0.4
M-MOJ_24 0.3
M-MOJ_34 0.3  0.4

Binary variables
Y(i,j,k) variable que indica si la unidad j comienza la tarea i en el momento T(k)
Positive variables
Z(j,k) variable que indica si la unidad j comienza una tarea en el momento T(k)
YM(i,j,k) si la unidad j esta continuando con la tarea i en el momento T(k)
YE(i,j,k) si la unidad j finalizo la tarea i y descargando este lote en el momento T(k)

```

```

italcol 1.1.gms
SL(k)
B(i,j,k)
bm(i,j,k)
BE(i,j,k)
t(j,k)
N(m,k)
Variable
MS
Equations
MakeSpan
Inicio
Tamano1
Tamano2
Estado1
Estado2
Tiempo
Lote1
Lote21
Lote22
Lote31
Lote32
Lote4
Inventario
Limite1
Limite2
Limite3
Limite4
Limite5, Limite6
Demanda
Cumpimiento de la demanda ;

Z.lo(j,k)=0;
Z.up(j,k)=1;
ym.lo(i,j,k)=0;
ym.up(i,j,k)=1;

```

Longitud del eslot k
 Tamano de lote de la tarea i en la unidad j comenzando en T(k)
 Cantidad de material primario (mi) que recida en la unidad j justo antes de T(k)
 Cantidad que la tarea i descarga a su realizacion en T(k)
 Denota el tiempo faltante a T(k) completada la tarea que estuvo en progreso durante el slot k en la unidad j
 Balance de material m en T(k)

Time de produccion
 La suma total de la duracion de los slots no debe ser mayor al tiempo de produccion
 La unidad j comienza cuando una tarea comienza en T(k)
 Cantidad actual de material primario involucrado en el lote comenzando en T(k) en la unidad j

Balance del estado de al unidad j
 Comienza la unidad j cuando esta finaliza la tarea inmediatamente precedente
 Relacion de tiempos
 Cantidad del lote en la unidad j
 Si una unidad j no realiza una tarea i en T(k). bm(ijk) debe ser igual a cero y vice versa

Una unidad puede descargar un lote en T(k) solo si su tarea finalizo y vice versa
 No nulidad de t(jk)
 Inventario de material
 Limita la longitud del slot k a no ser mayor que el maximo tiempo de procesamiento en cualquier unidad de cualquier
 Limita a t(j.k) a ser menor o igual que el maximo tiempo de procesamiento entre las tareas Ij en cualquier T(k)

Limita la capacidad de los almacenamientos de material (m)
 El tamano del lote B(ijk) no mayor al maximo de la unidad j en la tarea i en T(k)
 Limite5, Limite6
 Demanda Cumpimiento de la demanda ;



italcol 1.1.gms

```

YE.lo(i,j,k)=0;
YE.up(i,j,k)=1;
SL.up("k0")=0;
Z.up(j,k){ord(k)=card(k)}=0;
Y.up(i,j,k){ord(k)=card(k)}=0;
B.up(i,j,k){ord(k)=card(k)}=0;
ym.up(i,j,"k0")=0;
ym.up(i,j,k){ord(k)=card(k)}=0;
YE.up(i,j,"k0")=0;
t.up(j,"k0")=0;
t.up(j,k){ord(k)=card(k)}=0;
bm.up(i,j,"k0")=0;
bm.up(i,j,k){ord(k)=card(k)}=0;
BE.up(i,j,"k0")=0;

MakeSpan.. Sum(k$(ord(k)>1), SL(k)) =e= MS;

Inicio(j,k){ord(k)<card(k)}..
Z(j,k) =e= Sum(i$JI(j,i), Y(i,j,k));

Tamano1(i,j,k){JI(j,i)and ord(i)>1and ord(k)<card(k)}..
B(i,j,k) =1= Bu(i,j)*Y(i,j,k);

Tamano2(i,j,k){ord(i)>1 and ord(i)<12 and JI(j,i)and ord(k)<card(k)}..
B(i,j,k) =g= Bl(i,j)*Y(i,j,k);

Estado1(i,j,k){ord(k)>1 and JI(j,i)and ord(k)<card(k)}..
ym(i,j,k) =e= ym(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k);

Estado2(j,k){ord(k)>1 and ord(k)<card(k)}..
Z(j,k) =e= sum(i$JI(j,i), YE(i,j,k));

Tiempo(j,k){ord(k)<card(k)}..
t(j,k+1) =g= t(j,k) + sum(i$(JI(j,i)and ord(i)>1), c(i,j)*Y(i,j,k) + v(i,j)*B(i,j,k)) - SL(k+1));
    
```



italcol 1.1.gms

```

Lote1(i,j,k)$(ord(k)>1 and JI(j,i)and ord(i)>1)..
bm(i,j,k) =e= bm(i,j,k-1) + B(i,j,k-1) - BE(i,j,k);

Lote21(i,j,k)$(ord(i)>1 and ord(i)<12 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
bm(i,j,k) =g= B1(i,j)*ym(i,j,k);

Lote22(i,j,k)$(ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
bm(i,j,k) =l= Bu(i,j)*ym(i,j,k);

Lote31(i,j,k)$(ord(i)>1 and ord(i)<12 and ord(k)>1 and ord(k)<card(k)and JI(j,i))..
BE(i,j,k) =g= B1(i,j)*YE(i,j,k);

Lote32(i,j,k)$(ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
BE(i,j,k) =l= Bu(i,j)*YE(i,j,k);

Lote4(j,k)$(ord(k)>1 and ord(k)<card(k))..
t(j,k) =l= sum(i$(JI(j,i)and ord(i)>1), c(i,j)*ym(i,j,k) + v(i,j)*bm(i,j,k));

Inventario(m,k)..
N(m,k) =e= Ii(m)$(ord(k)=1) + N(m,k-1) + sum(i$(PM(i,m), sum(j$JI(j,i), BE(i,j,k)))
- sum(i$CM(i,m), sum(j$JI(j,i), PCM(i,m)*B(i,j,k)));

Limite1(k).. SL(k) =l= smax(j, smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j)));
Limite2(j,k).. t(j,k) =l= smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j));
Limite3(m,i,k)$mc(m).. N(m,k) =l= Ca(m);
Limite4(i,j,k)$(JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. B(i,j,k) =l= Bu(i,j);
Limite5(i,j,k)$(JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. bm(i,j,k) =l= Bu(i,j);
Limite6(i,j,k)$(JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. BE(i,j,k) =l= Bu(i,j);

Demanda(m,k)$(md(m)and ord(k)=card(k)).. N(m,k) =g= D(m);

Model italcol /all/;
Solve italcol using mip minimizing MS;
    
```

ANEXO H CÓDIGO FUENTE ITALCOL 2.1

```

gamside: C:\WINDOWS\games\dir\project_gpr - [E:\Project Curriculom\ITALCOL 2.1.gms]
File Edit Search Windows Help
italcol 2.1.gms
$*title [((((((((((((((((Modelo prototipo para programar produccion en ITALCOL S.C.A))))))))))))))))]]]]]]]]]]
* Autores: Jenny Angelica Angulo Soledad Y Ronald Fernando Fineda Bonilla
* Version 2.1

$OFFListing
option solprint=OFF;
option limcol = 0, limrow=0;

$set xls "C:\Documents and Settings\Estudiante\Mis documentos\RRPP\ARCHIVOS PROYECTO DE GRADO\ENTRADAS Y PARAMETROS.xls"
Sets
i tareas /TO, M-SHPH, M-SHPq, M-SHP 100h, M-SHP 100q, M-SPE Pp, M-SPE Pq, M-SPE Pg, M-SPE P, M-SPE 1S, M-SPI 2S, M-SPI 3S,
M-PCp, M-PCq, M-MOJ 24, M-MOJ 34, M-PR, M-PL, M-PP, Q-SHPq, Q-SHP 100q, Q-SHP 100q, Q-SPI 1S, Q-SPI 2S, Q-SPI 3S,
P-PCp, Q-PCq, PoQ-PR, PoQ-PL, PoQ-PP, P-SPE Pp, Q-SPE Pq, PoQ-SPEG, MO-MOJ 24, MO-MOJ 34, E-MOJ 24, E-MOJ 34/
j unidades /DOS, PEL1, PEL2, PEL3, MOL1, EXT/
JI(j,i)/(DOS, PEL1, PEL2, PEL3, MOL1, EXT).TO
DOS.(M-SHPH, M-SHPq, M-SHP 100h, M-SHP 100q, M-SPE Pp, M-SPE Pq, M-SPE Pg, M-SPE P, M-SPE 1S, M-SPI 2S, M-SPI 3S, M-PCp,
M-PCq, M-MOJ 24, M-MOJ 34, M-PR, M-PL, M-PP).(Q-SHPq, Q-SHP 100q, Q-SHP 100q, Q-SPI 1S, Q-SPI 2S, Q-SPI 3S,
P-PCp, Q-PCq, PoQ-PR, PoQ-PL, Q-PP), PEL2.(Q-SHPq, Q-SHP 100q, Q-SPI 1S, Q-SPI 2S, Q-SPI 3S, P-PCp, Q-PCq,
PoQ-PR, PoQ-PL, Q-PP, P-SPE Pp, Q-SPE Pq, PoQ-SPEG), MOL1.(MO-MOJ 24, MO-MOJ 34), EXT.(E-MOJ 24, E-MOJ 34)/
m materiales /m1*m45/
PM(i,m)/(M-SHPH,m13, M-SHPq,m14, M-SHP 100h,m14, M-SHP 100q,m15, M-SPE Pp,m16, M-SPE Pq,m17, M-SPE Pg,m18,
M-SPI 1S,m19, M-SPI 2S,m20, M-SPI 3S,m21, M-PCp,m22, M-PCq,m23, M-MOJ 24,m24, M-MOJ 34,m25, M-PR,m26, M-PL,m27,
M-PP,m28, Q-SHPq,m29, Q-SHP 100q,m30, Q-SPI 1S,m31, Q-SPI 2S,m32, Q-SPI 3S,m33, P-PCp,m34, Q-PCq,m35, PoQ-PR,m36,
PoQ-PL,m37, PoQ-PP,m38, P-SPE Pp,m39, Q-SPE Pq,m40, PoQ-SPEG,m41, MO-MOJ 24,m42, MO-MOJ 34,m43, E-MOJ 24,m44,
E-MOJ 34,m45/
CM(i,m)/(M-SHPH, M-SHPq, M-SHP 100h, M-SHP 100q, M-PL).(m2,m4,m5), (M-SPE Pp, M-SPE Pq, M-SPE Pg, M-SPI 1S, M-SPI 2S,
M-SPI 3S).(m1,m2,m3), (M-PCp, M-PCq).(m1,m9,m10), (M-MOJ 24, M-MOJ 34).(m1,m6,m7), M-PR.(m2,m8,m11),
M-PP.(m8,m9,m10), Q-SHPq,m13, Q-SHP 100q,m15, Q-SPI 1S,m19, Q-SPI 2S,m20, Q-SPI 3S,m21, P-PCp,m22, Q-PCq,m23,
PoQ-PR,m26, PoQ-PL,m27, Q-PP,m28, P-SPE Pp,m16, Q-SPE Pq,m17, PoQ-SPEG,m18, MO-MOJ 24,m24, MO-MOJ 34,m25,
E-MOJ 24,m42, E-MOJ 34,m43/
mc (m) conjunto de m restringidos por capacidad /m12, m13, m14, m15, m16, m17, m18, m19, m20, m21, m22, m23, m24, m25, m26,
m27, m28/
md (m) conjunto de m demandados /m12, m14, m29, m30, m31, m32, m33, m34, m35, m36, m37, m38, m39, m40, m41, m44, m45/
k slots /k0*k20/
1: 36 Modified Insert

```


gamside: C:\WINDOWS\gamsdir\project.epr - [E:\Project Curriculum\ITALCOL 2.1.gms]

File Edit Search Windows Help

italcol 2.1.gms

```

S-MOJ_34
Table v(i,j) termino variable del tiempo de procesamiento
DOS
M-SHPq
M-SHPq
Q-SHPq
M-SPE_Pp
M-SPE_Pq
P-SPE_Pp
Q-SPE_Pq
M-SPI_1s
M-SPI_2s
M-SPI_3s
Q-SPI_1S
Q-SPI_2S
Q-SPI_3S
M-PCp
M-PCq
P-PCp
Q-PCq
M-MOJ_24
M-MOJ_34
MO-MOJ_24
MO-MOJ_34
E-MOJ_24
E-MOJ_34
S-MOJ_24
S-MOJ_34
Table Bu(i,j) Capacidad de la unidad j con la tarea i
DOS
M-SHPq
M-SHPq
  
```

	PEL1	PEL2	PEL3	MOL1	EXT
DOS					
M-SHPq	0.00571	0.00571	0.00571		
M-SHPq					
Q-SHPq		0.00444			
M-SPE_Pp		0.005			
M-SPE_Pq					
P-SPE_Pp					
Q-SPE_Pq					
M-SPI_1s	0.005	0.005	0.005		
M-SPI_2s	0.005	0.005	0.005		
M-SPI_3s	0.005	0.005	0.005		
Q-SPI_1S	0.008	0.008	0.008		
Q-SPI_2S	0.008	0.008	0.008		
Q-SPI_3S					
M-PCp					
M-PCq					
P-PCp					
Q-PCq					
M-MOJ_24				0.02326	
M-MOJ_34				0.03333	
MO-MOJ_24					0.0608
MO-MOJ_34					0.0608
E-MOJ_24					
E-MOJ_34					
S-MOJ_24					
S-MOJ_34					
DOS					
M-SHPq					
M-SHPq					
PEL1					
PEL2					
PEL3					
MOL1					
EXT					

1: 36 Modified Insert


```

italcol 2.1.gms
M-SPI_3s      100
Q-SPI_1s
Q-SPI_2s
Q-SPI_3s
M-PCp        100
M-PCq        100
P-PCp
Q-PCq
M-MOJ_24     100
M-MOJ_34     100
MO-MOJ_24
MO-MOJ_34
E-MOJ_24
E-MOJ_34
S-MOJ_24
S-MOJ_34

Table PCM(i,m) Porcentaje de transferencia hacia la tarea material consumido por la tarea
m1 m2 m3 m4 m5 m6 m19 m20 m21 m26 m27
M-SHPH      0.3 0.3 0.4
M-SHPq      0.3 0.3 0.4
M-SPE_Pp    0.3 0.3 0.4
M-SPE_Pq    0.3 0.3 0.4
M-SPI_1s    0.3 0.3 0.4
M-SPI_2s    0.3 0.3 0.4
M-SPI_3s    0.3 0.3 0.4
M-PCp       0.3 0.3 0.4
M-PCq       0.3 0.3 0.4
M-MOJ_24    0.3
M-MOJ_34    0.3 0.4

Binary variables
Y(i,j,k) variable que indica si la unidad j comienza la tarea i en el momento T(k)
Positive variables

```



italcol2.1.gms

```

Z(j,k) variable que indica si la unidad j comieza una tarea en el momento T(k)
ym(i,j,k) si la unidad j esta continuando con la tarea i en el momento T(k)
YE(i,j,k) si la unidad j finalizo la tarea i y descargando este lote en el momento T(k)
SL(k) Longitud del eslot k
B(i,j,k) Tamano de lote de la tarea i en la unidad j comenzando en T(k)
bm(i,j,k) Cantidad de material primario (mi) que recida en la unidad j justo antes de T(k)
BE(i,j,k) Cantidad que la tarea i descarga a su realizacion en T(k)
t(j,k) Denota el tiempo faltante a T(k) completada la tarea que estubo en progreso durante el slot k en la unidad j
N(m,k) Balance de material m en T(k)
Variable
MS Time de produccion
Equations
MakeSpan La suma total de la duracion de los slots no debe ser mayor al tiempo de produccion
Inicio La unidad j comienza cuando una tarea comienza en T(k)
Tamano1 Cantidad actual de material primario involucrado en el lote comenzando en T(k) en la unidad j
Tamano2
Estado1 Balance del estado de al unidad j
Estado2 Comienza la unidad j cuando esta finaliza la tarea inmediatamente precedente
Tiempo Relacion de tiempos
Lote1 Cantidad del lote en la unidad j
Lote21 Si una unidad j no realiza una tarea i en T(k). bm(ijk) debe ser igual a cero y vice versa
Lote22
Lote31 Una unidad puede descargar un lote en T(k) solo si su tarea finalizo y vice versa
Lote32
Lote4 No nulidad de t(jk)
Inventario Inventario de material
Limite1 Limita la longitud del slot k a no ser mayor que el maximo tiempo de procesamiento en cualquier unidad de cualquier
Limite2 Limita a t(j.k) a ser menor o igual que el maximo tiempo de procesamiento entre las tareas Ij en cualquier T(k)
Limite3 Limita la capacidad de los almacenamientos de material (m)
Limite4 El tamano del lote B(ijk) no mayor al maximo de la unidad j en la tarea i en T(k)
Limite5, Limite6
Demanda Cumpimiento de la demanda ;
Z.lo(j,k)=0;

```



italcol 2.1.gms

```

Z.up(j,k)=1;
ym.lo(i,j,k)=0;
ym.up(i,j,k)=1;
YE.lo(i,j,k)=0;
YE.up(i,j,k)=1;
SL.up("KO")=0;
Z.up(j,k){ord(k)=card(k)}=0;
Y.up(i,j,k){ord(k)=card(k)}=0;
B.up(i,j,k){ord(k)=card(k)}=0;
ym.up(i,j,"KO")=0;
ym.up(i,j,k){ord(k)=card(k)}=0;
YE.up(i,j,"KO")=0;
t.up(j,"KO")=0;
t.up(j,k){ord(k)=card(k)}=0;
bm.up(i,j,"KO")=0;
bm.up(i,j,k){ord(k)=card(k)}=0;
BE.up(i,j,"KO")=0;

MakeSpan.. Sum(k$(ord(k)>1), SL(k)) =e= MS;

Inicio(j,k){ord(k)<card(k)}..
Z(j,k) =e= Sum(i$JI(j,i), Y(i,j,k));

Tamano1(i,j,k){JI(j,i)and ord(i)>1and ord(k)<card(k)}..
B(i,j,k) =1= Bu(i,j)*Y(i,j,k);

Tamano2(i,j,k){ord(i)>1 and ord(i)<12 and JI(j,i)and ord(k)<card(k)}..
B(i,j,k) =g= B1(i,j)*Y(i,j,k);

Estado1(i,j,k){ord(k)>1 and JI(j,i)and ord(k)<card(k)}..
ym(i,j,k) =e= ym(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k);

Estado2(j,k){ord(k)>1 and ord(k)<card(k)}..
Z(j,k) =e= sum(i$JI(j,i), YE(i,j,k));
    
```



italcol 2.1.gms

```
Tiempo(j,k)$ (ord(k)<card(k))..
t(j,k+1) =g= t(j,k) + sum(i$(JI(j,i)and ord(i)>1), c(i,j)*v(i,j,k) + v(i,j)*B(i,j,k) - SL(k+1));

Lote1(i,j,k)$ (ord(k)>1 and JI(j,i)and ord(i)>1)..
bm(i,j,k) =e= bm(i,j,k-1) + B(i,j,k-1) - BE(i,j,k);

Lote21(i,j,k)$ (ord(i)>1 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
bm(i,j,k) =g= B1(i,j)*ym(i,j,k);

Lote22(i,j,k)$ (ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
bm(i,j,k) =l= Bu(i,j)*ym(i,j,k);

Lote31(i,j,k)$ (ord(i)>1 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
BE(i,j,k) =g= B1(i,j)*YE(i,j,k);

Lote32(i,j,k)$ (ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
BE(i,j,k) =l= Bu(i,j)*YE(i,j,k);

Lote4(j,k)$ (ord(k)>1 and ord(k)<card(k))..
t(j,k) =l= sum(i$(JI(j,i)and ord(i)>1), c(i,j)*ym(i,j,k) + v(i,j)*bm(i,j,k));

Inventario(m,k)..
N(m,k) =e= Ii(m)$ (ord(k)=1) + N(m,k-1) + sum(i$(PM(i,m), sum(j$(JI(j,i), BE(i,j,k)))
- sum(i$(CM(i,m), sum(j$(JI(j,i), PCM(i,m)*B(i,j,k))));

Limite1(k).. SL(k) =l= smax(j, smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j)));
Limite2(j,k).. t(j,k) =l= smax(i$(JI(j,i)and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j));
Limite3(m,i,k)$mc(m).. N(m,k) =l= Ca(m);
Limite4(i,j,k)$ (JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. B(i,j,k) =l= Bu(i,j);
Limite5(i,j,k)$ (JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. bm(i,j,k) =l= Bu(i,j);
Limite6(i,j,k)$ (JI(j,i)and ord(i)>1and ord(k)>1and ord(k)<card(k)).. BE(i,j,k) =l= Bu(i,j);
Demanda(m,k)$ (md(m)and ord(k)=card(k)).. N(m,k) =g= D(m);

Model itacol /all/;
Solve itacol using mip minimizing MS;
```

ANEXO I CÓDIGO FUENTE ITALCOL 2.2

```
italcol2.2.gms

$title [[[[[[[[[[{{{({Modelo prototipo para programar produccion en ITALCOL S.C.A)}}}}}}}}]]]]]]]]]]]]
* Autores: Jenny Angelica Angulo Soledad Y Ronald Fernando Pineda Bonilla
* Version 2.2 Maximiza Beneficio

$Onlisting
$Onsymxref
$Onsymlist
$Onuexref
$Onuellist
$Onupper
option solprint=on;
option limcol = 300, limrow = 300;

$set xls "C:\Documents and Settings\Estudiante\Mis documentos\RRPP\ARCHIVOS PROYECTO DE GRADO\INTERFAZ 2.2.xls"
Sets
i tareas /
$calls =xls2gms r=E1:E32 i="%xls%" o=tareas.inc
$include tareas.inc
/
j unidades /
$calls =xls2gms r=a35:a40 i="%xls%" o=unidades.inc
$include unidades.inc
/
JI(j,i) /
$calls =xls2gms r=c3:c55 i="%xls%" o=unitarea.inc
$include unitarea.inc
/
m materiales /
$calls =xls2gms r=bw1:bw41 i="%xls%" o=emes.inc
$include emes.inc
/
PM(i,m) /
$calls =xls2gms r=b3:b32 i="%xls%" o=prodm.inc
$include prodm.inc
```



italcol 2.2.gms

```

/ CM(i,m) /
$call =xls2gms r=d3:d32 i="%xls%" o=consum.inc
$include consum.inc
/
inc(m) conjunto de m restringidos por capacidad /
$call =xls2gms r=bx13:bx39 i="%xls%" o=emesC.inc
$include emesC.inc
/
ind(m) conjunto de m demandados /
$call =xls2gms r=bz12:bz41 i="%xls%" o=emesP.inc
$include emesP.inc
/
k slots /
$call =xls2gms r=e51 i="%xls%" o=slot.inc
$include slot.inc
/
Scalar
H tiempo horizonte /
$call =xls2gms r=e52 i="%xls%" o=horizonte.inc
$include horizonte.inc
/
Parameters
Ii(m) Inventario inicial en bultos / (m1*m5) 2000 /
Ca(m) Capacidad de almacenamientos /
$call =xls2gms r=bx13:bx39 i="%xls%" o=cap.inc
$include cap.inc
/
Dmax(m) /
$call =xls2gms r=e35:f48 i="%xls%" o=demanda.inc
$include demanda.inc
/
Table c(i,j) termino constante del tiempo de procesamiento
$call =xls2gms r=e2:k32 i="%xls%" o=tabla1.inc

```

```

$include tabla1.inc
Table v(i,j) termino variable del tiempo de procesamiento
$call =xls2gms r=12:r32 i="%xls%" o=tabla2.inc
$include tabla2.inc
Table Bu(i,j) Capacidad de la unidad j con la tarea i
$call =xls2gms r=s2:y32 i="%xls%" o=tabla3.inc
$include tabla3.inc
Table B1(i,j) Cantidad minima de la unidad j con la tarea i
$call =xls2gms r=z2:aa17 i="%xls%" o=tabla4.inc
$include tabla4.inc
Table PCM(i,m) Porcentaje de transferencia hacia la tarea material consumido por la tarea
$call =xls2gms r=ag2:bi32 i="%xls%" o=tabla5.inc
$include tabla5.inc
Binary variables
Y(i,j,k) variable que indica si la unidad j comienza la tarea i en el momento T(k)
Positive variables
Z(j,k) variable que indica si la unidad j comienza una tarea en el momento T(k)
ym(i,j,k) si la unidad j esta continuando con la tarea i en el momento T(k)
YE(i,j,k) si la unidad j finaliza la tarea i descartando este lote en el momento T(k)
SL(k) Longitud del eslot k
B(i,j,k) Tamano de lote de la tarea i en la unidad j comenzando en T(k)
bm(i,j,k) Cantidad de material primario (mi) que recida en la unidad j justo antes de T(k)
BE(i,j,k) Cantidad que la tarea i descarga a su realizacion en T(k)
t(j,k) Denota el tiempo faltante a T(k) completada la tarea que estuvo en progreso durante el slot k en la unidad j
N(m,k) Balance de material m en T(k)
Variable
P Utilidad por ventas
Equations
Horizonte La suma total de la duracion de los slots no debe ser mayor al tiempo de produccion
Inicio La unidad j comienza cuando una tarea comienza en T(k)
Tamano1 Cantidad actual de material primario involucrado en el lote comenzando en T(k) en la unidad j
Tamano2
Estado1 Balance del estado de al unidad j
Estado2 Comienza la unidad j cuando esta finaliza la tarea inmediatamente precedente

```

gamside: C:\WINDOWS\gamsdir\project.gpr - [E:\Project Curriculum\ITALCOL 2.2.gms]

File Edit Search Windows Help

italcol 2.2.gms

Tiempo	Relacion de tiempos
Lote1	Cantidad del lote en la unidad j
Lote21	Si una unidad j no realiza una tarea i en T(k). $bm(ijk)$ debe ser igual a cero y vice versa
Lote22	
Lote31	Una unidad puede descargar un lote en T(k) solo si su tarea finalizo y vice versa
Lote32	
Lote4	No nulidad de t(jk)
Inventario	Inventario de material
Limite1	Limita la longitud del slot k a no ser mayor que el maximo tiempo de procesamiento en cualquier unidad de cualquier
Limite2	Limita a t(j.k) a ser menor o igual que el maximo tiempo de procesamiento entre las tareas Ij en cualquier T(k)
Limite3	Limita la capacidad de los almacenamientos de material (m)
Limite4	El tamano del lote B(ijk) no mayor al maximo de la unidad j en la tarea i en T(k)
Limite5, Limite6	
Ventas	Cumpimiento de la demanda
demandamax:	
Z.lo(j,k)=0;	
Z.up(j,k)=1;	
ym.lo(i,j,k)=0;	
ym.up(i,j,k)=1;	
YE.lo(i,j,k)=0;	
YE.up(i,j,k)=1;	
SL.up("k0")=0;	
Z.up(j,k)\$(ord(k)=card(k))=0;	
Y.up(i,j,k)\$(ord(k)=card(k))=0;	
B.up(i,j,k)\$(ord(k)=card(k))=0;	
ym.up(i,j,"k0")=0;	
ym.up(i,j,k)\$(ord(k)=card(k))=0;	
YE.up(i,j,"k0")=0;	
t.up(j,"k0")=0;	
t.up(j,k)\$(ord(k)=card(k))=0;	
bm.up(i,j,"k0")=0;	
bm.up(i,j,k)\$(ord(k)=card(k))=0;	
BE.up(i,j,"k0")=0;	

1:1 Modified Insert



italcol 2.2.gms

```

Horizonte.. Sum(k$(ord(k)>1), SL(k) =1= H;

Inicio(j,k)$(ord(k)<card(k))..
Z(j,k) =e= Sum(i$JI(j,i), Y(i,j,k));

Tamano1(i,j,k)$(JI(j,i)and ord(i)>1and ord(k)<card(k))..
B(i,j,k) =1= Bu(i,j)*Y(i,j,k);

Tamano2(i,j,k)$(ord(i)>1 and JI(j,i)and ord(k)<card(k))..
B(i,j,k) =g= B1(i,j)*Y(i,j,k);

Estado1(i,j,k)$(ord(k)>1 and JI(j,i)and ord(k)<card(k))..
ym(i,j,k) =e= ym(i,j,k-1) + Y(i,j,k-1) - YE(i,j,k);

Estado2(j,k)$(ord(k)>1 and ord(k)<card(k))..
Z(j,k) =e= sum(i$JI(j,i), YE(i,j,k));

Tiempo(j,k)$(ord(k)<card(k))..
t(j,k+1) =g= t(j,k) + sum(i$(JI(j,i)and ord(i)>1), c(i,j)*Y(i,j,k) + v(i,j)*B(i,j,k) - SL(k+1));

Lote1(i,j,k)$(ord(k)>1 and JI(j,i)and ord(i)>1)..
bm(i,j,k) =e= bm(i,j,k-1) + B(i,j,k-1) - BE(i,j,k);

Lote21(i,j,k)$(ord(i)>1 and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
bm(i,j,k) =g= B1(i,j)*ym(i,j,k);

Lote22(i,j,k)$(ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
bm(i,j,k) =1= Bu(i,j)*ym(i,j,k);

Lote31(i,j,k)$(ord(i)>1and ord(k)>1 and ord(k)<card(k) and JI(j,i))..
BE(i,j,k) =g= B1(i,j)*YE(i,j,k);

Lote32(i,j,k)$(ord(k)>1 and ord(k)<card(k) and JI(j,i)and ord(i)>1)..
BE(i,j,k) =1= Bu(i,j)*YE(i,j,k);
    
```

```

italcol 2.2.gms

Lote4(j,k)$(ord(k)>1 and ord(k)<card(k))..
t(j,k) =l= sum(i$(JI(j,i) and ord(i)>1), c(i,j)*ym(i,j,k) + v(i,j)*bm(i,j,k));

Inventario(m,k)..
N(m,k) =e= Ii(m)$(ord(k)=1) + N(m,k-1)$(ord(k)>1) + sum(i$(PM(i,m), sum(j$(JI(j,i), BE(i,j,k))) - sum(i$(CM(i,m), sum(j$(JI(j,i),
Limite1(k).. SL(k) =l= smax(j, smax(i$(JI(j,i) and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j)));
Limite2(j,k).. t(j,k) =l= smax(i$(JI(j,i) and ord(i)>1), c(i,j) + v(i,j)*Bu(i,j));
Limite3(m,i,k) $mc(m).. N(m,k) =l= Ca(m);
Limite4(i,j,k) $(JI(j,i) and ord(i)>1 and ord(k)>1 and ord(k)<card(k)).. B(i,j,k) =l= Bu(i,j);
Limite5(i,j,k) $(JI(j,i) and ord(i)>1 and ord(k)>1 and ord(k)<card(k)).. bm(i,j,k) =l= Bu(i,j);
Limite6(i,j,k) $(JI(j,i) and ord(i)>1 and ord(k)>1 and ord(k)<card(k)).. BE(i,j,k) =l= Bu(i,j);
Ventas(k)$(ord(k)=card(k)).. P =e= Sum(m$(md(m),N(m,k)));
demandaMax(m,k)$(ord(k)=card(k) and md(m)).. N(m,k) =g= Dmax(m);

Model italcol /all/;
Solve italcol using rmip maximizing P;
Display SL.1, SL.m, B.1, B.m, bm.1, bm.m, BE.1, BE.m, N.1, N.m;
italcol.reslim = 5000;

```