



TESIS DE GRADO EN
INGENIERÍA INDUSTRIAL

EVALUACIÓN DE ALTERNATIVAS PARA
PROGRAMACIÓN DE LA PRODUCCIÓN

Autor: Alejandro Nicolás Carabba
Legajo: 45.455

Director de Tesis:
Ing. Andrés Basilio Agres, MSc.

Año 2009

RESUMEN

El problema de programación de la producción puede sintetizarse de la siguiente manera: dada cierta cantidad de recursos (máquinas, mano de obra, insumos, tiempo) y cierta cantidad de órdenes de producción a realizarse durante un período de tiempo, secuenciar – es decir, ordenar – los trabajos de tal manera que se optimice un conjunto de indicadores definidos. Así enunciado, no parece demasiado complejo, pero algo más de un siglo de historia en intentos de solución demuestra que lo es. Actualmente está probado que por sus características este problema es uno de los más difíciles dentro de la gestión de la producción (y dentro de la teoría de la optimización en general), debido a que no puede resolverse con exactitud en un tiempo acotado.

En este proyecto final se plantea la siguiente solución innovadora al clásico problema descrito: generar diversas alternativas para la programación de la producción y evaluar sus resultados en un ambiente controlado para lograr definir una metodología, algoritmo o técnica que proporcione la mejor performance de acuerdo al conjunto de indicadores seleccionado. Dicho ambiente controlado se materializa en un modelo de simulación que representa la operación de una planta y sirve como plataforma de prueba para las diversas alternativas. Además de la construcción del propio modelo de simulación, en el proyecto también se discute el conjunto de metodologías a utilizar (desde el punto de vista teórico y práctico), la definición de indicadores, la validación del modelo, la comparación entre escenarios, y los resultados de este enfoque en el tratamiento de problemas de scheduling.

Durante el proyecto se desarrolla un caso de asesoramiento a una empresa real, cuyos datos se mantienen en reserva por razones de confidencialidad. Dicho caso se centra en la definición de una técnica para programar la producción en una planta de roscado, y permite evaluar el éxito de la solución adoptada y validar los criterios empleados.

Sobre el final de la tesis se resumen los resultados obtenidos, los cuales se verifican satisfactorios en todos los casos. Mediante el enfoque empleado, se logra demostrar la superioridad de las metodologías propuestas frente a la actual y se elabora un criterio de decisión para la posterior selección de la técnica. Además, como conclusión, se profundiza sobre los distintos aspectos de la complejidad de un problema de scheduling, se clasifican las variables intervinientes, y se elabora un criterio para la selección de posibles metodologías de resolución según el tipo de problema.

DESCRIPTOR BIBLIOGRÁFICO

El presente trabajo se centra en el problema de programación de la producción, particularmente en la definición de metodologías para su solución. Se analiza el estado de la cuestión pasando revista a las principales técnicas empleadas actualmente. Se desarrolla un caso real en el cual se genera un modelo de simulación para la comparación de distintas metodologías entre sí en un ambiente controlado, y se evalúa la performance de las mismas. Finalmente, además de validar el método empleado para la definición de una técnica, el caso se emplea como base para la elaboración de criterios más universales en cuanto a qué metodología aplicar según características propias de cada problema. Se logra solucionar satisfactoriamente un problema particular y extender las conclusiones obtenidas a un problema más general.

Palabras Clave: algoritmos genéticos, búsqueda tabú, heurísticas, programación de la producción, programación lineal, scheduling, secuenciamiento, simulación.

ABSTRACT

This paper is focused on the scheduling problem, particularly on the definition of a methodology for its solution. Firstly, we analyze the state of the issue by considering the main scheduling techniques nowadays. Then, a real case is explained in detail; in it, a simulation model is used to compare different methodologies by their performance. Finally, the case is used to validate the proposed method to define a technique, and also as a basis to develop more universal criteria related to the selection of methodologies for each problem according to its characteristics. A specific problem is successfully solved and conclusions are then generalized.

Keywords: genetic algorithms, heuristic methods, linear programming, production programming, scheduling, sequencing, simulation, tabu search.

AGRADECIMIENTOS

En estas breves líneas, quisiera expresar mi profundo agradecimiento a todas las personas que han colaborado, en mayor o menor medida, al desarrollo y concreción del presente trabajo. De dicho conjunto de personas, me gustaría reconocer especialmente a las siguientes:

- ▶ En primer lugar, al Ing. Andrés Basilio Agres, mi director de tesis. Sin su constante apoyo académico, profesional y hasta motivacional, jamás hubiera completado esta tarea.
- ▶ En segundo lugar, a mi familia, cuyo sustento emocional ha sido fundamental a la hora de escribir estos párrafos.
- ▶ Por último, a mis amigos, por inspirarme.

A todos ellos, mi infinita gratitud por su ayuda, en el sentido más amplio de la palabra.

TABLA DE CONTENIDOS

< I > Introducción

I.1 ♦ Una Introducción al Problema de la Programación de la Producción	1
I.2 ♦ Relevancia del Problema	3
I.3 ♦ Motivación para Abordar el Problema	4
I.4 ♦ Etapas del Proyecto	4
I.5 ♦ Criterios de Éxito del Proyecto	5

< II > Estado de la Tecnología

II.1 ♦ Evolución del Enfoque Conceptual Empleado para el Problema	7
II.2 ♦ Clasificación de Técnicas Modernas para la Resolución del Problema	12
II.3 ♦ La Técnica Más Popular: Programación Lineal	24
II.4 ♦ Casos Recientes de Aplicación de los Principales Métodos	28
II.5 ♦ Resumen del Estado de la Cuestión	30

< III > Definición del Problema

III.1 ♦ Conceptos Básicos y Terminología	31
III.2 ♦ El Problema	34
III.3 ♦ Objetivos, Restricciones y Recursos	34

< IV > Caso de Estudio

IV.1 ♦ Una Solución Integradora	37
IV.2 ♦ Desarrollo del Modelo de Simulación	38
IV.3 ♦ Desarrollo de Metodologías de Optimización	53
IV.4 ♦ Desarrollo de la Interfase	60
IV.5 ♦ Definición de Indicadores	61

< V > Resultados Obtenidos

V.1 ♦ Presentación de Resultados	64
V.2 ♦ Análisis y Discusión de Resultados	67

< VI > Conclusiones y Futuras Líneas de Investigación

VI.1 ♦ Recomendaciones en Base al Proyecto	73
VI.2 ♦ Conclusiones del Proyecto	75
VI.3 ♦ Evaluación del Éxito del Proyecto	77
VI.4 ♦ Futuras Líneas de Investigación	78

< VII > Bibliografía

VII.1 ♦ Bibliografía General	80
VII.2 ♦ Documentos Técnicos Específicos	80
VII.3 ♦ Otras Fuentes de Información	81

< VIII > Anexos

VIII.1 ♦ Algoritmo de Johnson	82
VIII.2 ♦ Mapa Completo de Metodologías de Resolución	85
VIII.3 ♦ Confidencialidad de la Información	86
VIII.4 ♦ Simulación: Distintos Paradigmas y Principales Conceptos	86
VIII.5 ♦ Resultados Completos de Algunas Corridas	87
VIII.6 ♦ Diagrama IDEF0 del Proceso de Roscado y Flujogramas Adicionales	88

< I > INTRODUCCIÓN

I.1 ♦ UNA INTRODUCCIÓN AL PROBLEMA DE LA PROGRAMACIÓN DE LA PRODUCCIÓN

El problema de la programación de la producción surge de manera natural al considerar la operación de cualquier empresa industrial mediante el clásico **ciclo de planeamiento** (véase *Figura 1*).

(Debe tenerse en cuenta que en este esquema simplificado se consideran únicamente los niveles de planeamiento que impactan directamente sobre la producción, dejando de lado el planeamiento comercial, el planeamiento financiero y cualquier otro perteneciente a demás áreas específicas de la empresa.)



Figura 1 - Ciclo de planeamiento

El **planeamiento estratégico** es usualmente llevado a cabo por los directores de la compañía, considerando horizontes temporales de más de 5 años y basándose en la misión de la firma. En este nivel se definen las grandes líneas de acción de la empresa en el largo plazo.

En un nivel jerárquico inferior, se toman las decisiones referentes al **planeamiento de la producción**. En este estrato se traducen los objetivos globales del planeamiento estratégico en metas concretas para las diversas categorías de producto.

Durante la ejecución del **planeamiento agregado**, los gerentes o jefes de producción se nutren del trabajo comentado en el párrafo anterior para planificar las líneas de producto a su cargo en el mediano plazo.

Finalmente, antes de llegar al propio acto de la producción, hace falta definir qué se debe producir en cada momento, en qué máquina, etc.. La generación de este tipo de planes detallados y de corto plazo es usualmente responsabilidad de **programación de la producción**. La tarea consiste en definir exactamente cómo se hará para producir lo prometido en el plan con los recursos disponibles y de la mejor manera posible.

Esta breve descripción muestra que el **ciclo de planeamiento** es la forma de traducir las ideas e iniciativas en producción concreta; es un proceso escalonado en donde se van incorporando progresivamente las restricciones propias de la operación para bajar de lo ideal a lo real. A su vez, al tratarse cada vez de problemas más concretos a medida que se desciende en el ciclo, se requiere un menor horizonte temporal, ya que las decisiones que pueden tomarse tienen un menor alcance.

El resultado de la interacción entre las dos variables mostradas en las flechas verticales de la *Figura 1* hace que la programación de la producción sea el problema con mayor número de restricciones y menor horizonte temporal dentro del ciclo de planeamiento; es decir, es el problema más concreto y menos subjetivo de los cuatro. Sería insólito que una empresa confiara su planeamiento estratégico a una computadora o lo tercerizara, ya que del mismo se desprende casi el total de la operación de la compañía; sin embargo, la tarea de traducir requerimientos y restricciones de producción en un plan concreto es mucho menos influyente a nivel estratégico y puede ser tratada como un problema aislado sin mayores complicaciones.

En definitiva, el problema central que se plantea en la presente tesis es: de qué manera se pueden armonizar los distintos inputs (algunos de ellos mostrados en la *Figura 2*) para lograr una mejor programación de la producción en base a criterios cuantificables. Para abordarlo, se pasará revista a las técnicas pasadas y actuales empleadas en su resolución, y se propondrá un enfoque innovador sobre el mismo. Finalmente, se presentarán los resultados alcanzados y se obtendrán algunas conclusiones.

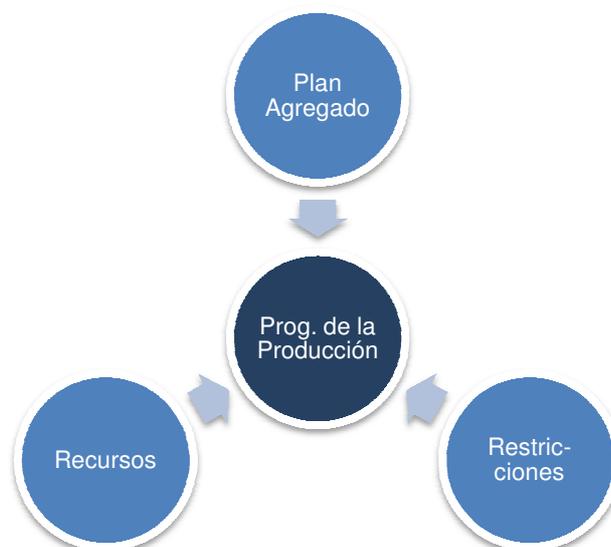


Figura 2 - Principales inputs del problema

1.2 ♦ RELEVANCIA DEL PROBLEMA

La relevancia del problema a tratar en el ámbito industrial es, sin duda, reconocida. Desde los desarrollos iniciales del estudio del trabajo – a principios del siglo XX – se ha tenido en consideración el problema de la programación de la producción, ensayándose diversas soluciones durante los últimos 100 años.

Los beneficios de un buen enfoque y una resolución satisfactoria al mismo son evidentes: maximización de la utilización de los equipos, posibilidad de cumplir mayor cantidad de órdenes, reconocimiento con antelación de órdenes que no pueden cumplirse para renegociar la fecha de entrega, minimización de los tiempos de entrega, correcta distribución de la mano de obra, planificación de horas extras, planificación de la capacidad a mediano plazo e identificación de cuellos de botella dinámicos, entre otros (véase Figura 3). Particularmente para empresas industriales, cobra gran relevancia económica la minimización del tiempo ocioso de la maquinaria, ya que el costo de oportunidad de un equipo ocioso encarece al producto de forma notoria e innecesaria.

Si bien el problema del **scheduling** (el equivalente anglosajón para “programación de la producción”) es importante para la operación de cualquier empresa, queda muchas veces relegado detrás del planeamiento de la producción, debido a la incumbencia estratégica de este último. Por tales razones, al existir menos aplicativos generales para programación de la producción, es posible abordar el problema de una forma más amplia y desestructurada.



Figura 3 - Principales beneficios

Para evidenciar la presencia del tema a nivel académico y empresarial, vale la pena destacar la existencia de millares de libros editados a nivel internacional sobre *scheduling*¹ y el hecho de que algunas de las compañías más grandes del mundo – de las llamadas **Fortune 500** – invierten sumas impensables para mejorar su situación en esta materia [Kumar, 2006].

¹ Basta con ingresar a Google Books (véase VII.3 [25]) y realizar una búsqueda de “scheduling” para comprobar la existencia de 20,000 libros referentes al tema.

Por otra parte, el problema de programación de la producción también posee relevancia desde el punto de vista económico y social. Debe entenderse que, en última instancia, lo que se busca es hacer más eficiente el uso de los recursos de los que dispone la sociedad y organizar la producción de manera óptima para aprovecharlos al máximo. Asimismo, los desarrollos en esta disciplina pueden volcarse con facilidad en otras industrias debido a la semejanza existente entre todos los problemas del área, con lo cual los beneficios obtenidos de este tipo de análisis pueden redundar en rápidas mejoras en otros ámbitos con implicaciones sociales mucho más evidentes.

I.3 ♦ MOTIVACIÓN PARA ABORDAR EL PROBLEMA

Existen básicamente tres motivos fundamentales para abordar el problema descrito en el presente trabajo.

- ▶ En primer lugar, el interés personal en cuanto a la temática
- ▶ En segundo lugar, el conocimiento de técnicas de simulación que permiten dar un enfoque distinto a un problema clásico
- ▶ Y por último, la realización de un trabajo de asesoría en servicios tecnológicos a una empresa industrial de roscado de tubos que permitió el contacto real con el problema de la programación de la producción y fue el disparador de una investigación referente al tema.

Las tres razones mencionadas, sumadas a la idea de que puede ser un trabajo fructífero a nivel académico y profesional, constituyeron la base para encarar este proyecto.

I.4 ♦ ETAPAS DEL PROYECTO

El trabajo de servicios tecnológicos presentado como caso de estudio para esta tesis puede esquematizarse como se aprecia en la *Figura 4 (página siguiente)*. El mismo consistió en 5 etapas secuenciales (que en la práctica se dieron con cierto solapamiento), aunque en el esquema se incluye también una sexta etapa de **Implementación** que se encuentra fuera del alcance de este documento.

Puede observarse claramente como durante todo el proyecto se da una interacción entre el mundo real y el mundo “virtual”, es decir, la representación del mundo real mediante un modelo. Esta interacción cobra gran relevancia en las 4 etapas recuadradas y será fundamental para el proyecto.

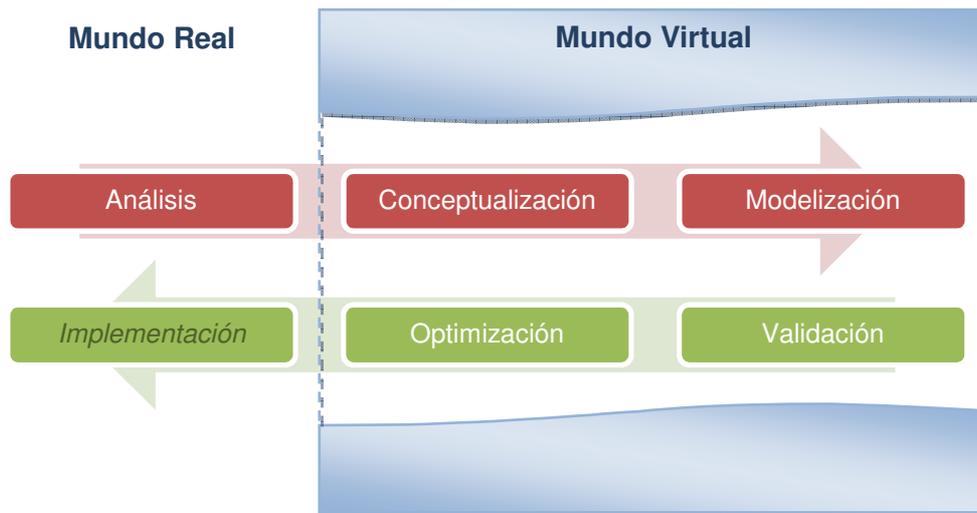


Figura 4 - Etapas del proyecto

En la primera etapa, **análisis**, lo que se hace es básicamente relevar el proceso a estudiar: la programación de la producción de tubos para una planta de roscado. A continuación, se **conceptualiza** el proceso en diagramas estándar y se define con mayor claridad cada parte del mismo. En esta etapa, se acuerdan con el cliente los límites del proyecto. Luego se encara la **modelización** del proceso en un software específico. En este período se busca representar de la mejor manera posible un problema real en un entorno virtual, materializado en un modelo de simulación. En la fase final de la construcción del modelo se procede a simular la situación real de la planta y **validar** los resultados obtenidos contrastándolos contra los reales de períodos pasados. Una vez aprobado el modelo, se trabaja en la **optimización** de la programación de la producción, ensayando diversas técnicas y comparando los resultados de cada una de ellas para definir la más adecuada. Finalmente, se **implementa** la técnica seleccionada mediante un sistema de programación de la producción que maneje el scheduling de la planta en cuestión.

1.5 ♦ CRITERIOS DE ÉXITO DEL PROYECTO

El éxito o fracaso del proyecto no podrá dictaminarse de una manera categórica, ya que la escala de medida de los criterios que aquí se mencionan es decididamente gradual. Se considerará que el proyecto ha sido exitoso si se han podido cumplir, en alguna medida, estos cuatro objetivos principales, siendo los primeros más personales e importantes para la valoración del éxito del trabajo y los últimos, algo más cercano a una expresión de deseo sobre los resultados ideales del proyecto.

1. En primer lugar, se priorizará que el proyecto genere un **aprendizaje** para el autor en esta temática y en el desarrollo de proyectos en general, ya que esta tesis cierra el ciclo de formación profesional.
2. Además, se buscará que la solución generada para el problema en cuestión sea **satisfactoria** desde el punto de vista del equipo y del cliente, de manera de haber trabajado en un proyecto que conforme a ambos.
3. También será importante considerar si el proyecto logró o no una **mejora en los indicadores** que monitorean el problema a tratar, de manera de corroborar objetivamente si la solución al problema fue exitosa.
4. Finalmente, se pretenderá que este proyecto colabore en la **creación de nuevos conocimientos** tanto para la universidad como para la comunidad académico-industrial argentina.

< II > ESTADO DE LA TECNOLOGÍA

II.1 ♦ EVOLUCIÓN DEL ENFOQUE CONCEPTUAL EMPLEADO PARA RESOLVER EL PROBLEMA DE SCHEDULING²

El problema de la programación de la producción surgió conjuntamente con el advenimiento del trabajo organizado en talleres y pequeñas fábricas, durante la revolución industrial. Sin embargo, no se tomó conciencia del mismo ni se estudió formalmente hasta principios del siglo XX. A partir de entonces, desde hace algo más que 100 años, se han ensayado diversos enfoques para su resolución, producto de los distintos paradigmas de la administración – en sentido amplio – por los que ha pasado la sociedad contemporánea.

En este apartado se pretende realizar una recorrida por esos enfoques y paradigmas, con el objeto de comprender el entorno en el cual se estudia el problema y evitar caer en lugares comunes de resolución.

II.1.a – El Camino Hacia La Programación Lineal

Si bien el estudio del problema en cuestión no se inició hasta principios del siglo XX, la necesidad de programar los trabajos siempre estuvo allí a la hora de producir. La falta de definición y reconocimiento del problema durante el siglo XIX no implicaban que el problema no existiera; y, lógicamente, todo problema requiere de alguna solución.

La solución más habitual en aquellos tiempos puede catalogarse como “a dedo” en el lenguaje coloquial actual. Lo que se hacía era simplemente seguir las órdenes del encargado al mando sin reparar en ningún tipo de optimización y, lógicamente, los resultados no eran para nada buenos. Dependían totalmente de la voluntad del jefe y de las rotaciones de personal, pero al no realizarse ningún análisis referido al tema, la decisión unilateral era totalmente arbitraria. Este enfoque podría denominarse, de manera simplista y algo exagerada, **fuerza bruta**. Fue la única solución al problema durante décadas y respondía al escaso avance en organización del trabajo de la época.

A principios del siglo XX, los primeros análisis de Fredrick Winslow Taylor referidos al estudio del trabajo tuvieron sus consecuencias en la planificación y programación de la producción. Conjuntamente, se desarrolló una herramienta de fundamental importancia para la administración de tareas en paralelo: el diagrama o gráfico de Gantt, introducido por Henry Gantt (*véase Figura 5 en página siguiente*).

² Adaptado principalmente de [8], [10] y [13].

En función de dichos estudios y herramientas, se desarrolló una metodología que consistía en 3 sencillos pasos:

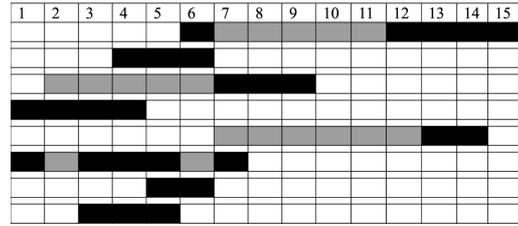


Figura 5 - Diagrama de Gantt (Ej.)

1. Aceptar la cantidad de órdenes permitida por la capacidad de las máquinas
2. Asignar los trabajos pendientes a las máquinas desocupadas
3. Monitorear el trabajo mediante el Diagrama de Gantt

Es decir, los trabajos pendientes se iban asignando a medida que las máquinas se iban desocupando, de manera de **evitar las paradas de máquina**. Lógicamente, esta política busca de alguna manera minimizar el tiempo ocioso pero de manera muy rudimentaria. Entre sus principales problemas puede mencionarse que los trabajos no siempre podían realizarse de la forma prevista y que las mejoras por simple inspección del diagrama eran muy difíciles de lograr.

El desarrollo de la programación lineal por George Dantzig en el año 1947 generó un quiebre en la resolución de problemas de scheduling. Si bien al día de hoy es una de las técnicas favoritas para encarar este tipo de problemas (véase la sección II.3), en su momento lo que generó fue un cambio de enfoque radical.

Al permitir una óptima definición del mix de productos a fabricar mediante cálculos relativamente sencillos (recuérdese que debían hacerse a mano), la programación lineal dirigió las miradas industriales a resolver principalmente el problema del planeamiento de la producción, dejando la programación de la producción a cargo de los supervisores. Lo que se priorizaba era **optimizar el “qué”**, mientras el “cómo” y el “cuándo” quedaban a merced de la experiencia de los jefes de producción.

De alguna manera, este paso implicó un retroceso a la **fuerza bruta**, ya que se dejó de desarrollar una solución al problema del scheduling y se perdió el foco durante algunos años. Sin embargo, al lograr optimizar el mix y adecuarlo a las capacidades de máquina, los resultados presentaron mejorías con respecto a los obtenidos en la pre-guerra.

II.1.b – Métodos Alternativos Promediando El Siglo XX

Las primeras incursiones en la resolución del problema del “**cómo**”/ “**cuándo**”, que había sido abandonado con el advenimiento de nuevas

técnicas, se dieron bajo el paradigma de la simplificación. Lo central era hacer uso de las herramientas de las que se disponía para resolver óptimamente el problema planteado. Para lograrlo, se recurría a la simplificación extrema del problema bajo la idea madre “**divide y vencerás**”, resultando a veces en una representación tan irreal del conflicto a resolver que su secuenciamiento se convertía en una mera discusión teórica sin ningún tipo de aplicación.

Por ejemplo, uno de los primeros algoritmos que resolvían el problema se atribuye a S. M. Johnson, quien en 1954 logró diseñar los pasos para arribar una solución óptima del problema reducido a dos máquinas solamente³. Sin embargo, para lograrlo debió hacer valer una gran lista de supuestos y simplificaciones explícitas e implícitas que alejaban el problema de la realidad.

Los avances basados en este paradigma fueron importantes, algunas restricciones lograron derribarse y con el correr de los años se logró formular problemas que aproximarán mejor la situación de un centro productivo; estos avances dieron lugar al siguiente paradigma dominante, que postulaba que los problemas de scheduling eran **demasiado complejos** como para intentar resolverlos. Se comprendió que para resolver satisfactoriamente un problema de programación de la producción requerían importantes recursos que podían ser destinados a otras actividades con mayores retornos.

De esta manera, hasta principios de los '70, los avances en procedimientos de resolución fueron escasos. Se desarrollaron pocos algoritmos bajo la filosofía de “**divide y vencerás**”, y se demostró que muchos de ellos no podían resolverse con la tecnología disponible; es decir, se podría no llegar a una solución dependiendo de las características del problema particular.

A partir de 1980, habiendo ya aceptado que era prácticamente imposible hallar soluciones generales óptimas al problema de scheduling, se comenzó a trabajar en diversas heurísticas que permitieran encontrar soluciones aproximadas con un esfuerzo razonable. Además, se hallaron contextos en los cuales la programación de la producción podía resolverse de forma óptima y se estudiaron dichos casos. De esta manera, se ingresó en una etapa de **aproximar la solución**, durante la cual se desarrollaron algunas de las soluciones heurísticas que se analizarán más adelante (*véase II.2*). La mejora en los tiempos de procesamiento y la posibilidad de hallar soluciones razonables aumentó la aceptación de estos desarrollos en el ambiente empresarial, dejando atrás aquella idea de que la complejidad inherente al problema hacía infructuoso su tratamiento.

³ Véase VIII.1 para una explicación sobre el funcionamiento del algoritmo de Johnson

II.1.c – La Era De La Información

Hacia fines de los '80, con el auge de los sistemas de **planeamiento de los requerimientos de materiales (MRP)** a nivel mundial, se inició una tendencia que ha permanecido vigente hasta nuestros días en el tratamiento de este tipo de problemas: el manejo de información mediante sistemas para facilitar la toma de decisiones. Desde entonces se han desarrollado diversas formas de almacenar y mostrar información para permitir que el decisor tenga un panorama lo más completo posible a la hora de accionar (un fruto más reciente de esta tendencia es el “tablero de comando”).

En este contexto, el primer enfoque aceptado para la resolución de los problemas de scheduling se basó en contar con la **información precisa**. A la hora de decidir, se propiciaba una interacción entre el gerente/supervisor y el sistema o software, para lograr una solución en conjunto. Para ello se requería que la persona contara con toda la información disponible referente al tema. De esta manera, se pensó por primera vez a los métodos cuantitativos como complementarios – y no como sustitutivos – de la experiencia y el buen juicio de los analistas.

Los resultados de este modelo de integración para hallar soluciones fueron buenos, pero lógicamente subóptimos. En la mayoría de los casos fue imposible saber cuánto mejor se podrían hacer las cosas, pero al menos existía una relativa certeza de que se estaban haciendo bastante bien.

En la década final del siglo XX se produjo nuevamente un cambio radical de enfoque, equivalente a la creencia anteriormente mencionada de que los problemas de scheduling eran **demasiado complejos**. En este caso, debido al avance de la filosofía japonesa del **justo a tiempo (JIT)** principalmente en el ámbito académico aunque también en el industrial. Se redujo el problema de programación de la producción a simples reglas de prioridad; por el esquema de flujo de trabajo que se planteaba, no se requería resolver enormes y complicados problemas, sino que con dichas reglas podía darse por terminado el conflicto sin siquiera definirlo. La idea madre que apoyaba esta posición era básicamente que **no era necesario** resolver estos problemas; la independencia de máquinas, versatilidad de personal y tecnología en el manejo coordinado de operaciones los volvían irrelevantes. Dado que sólo un bajo porcentaje de empresas adoptó esta filosofía a nivel mundial, debieron ensayarse otras técnicas para resolver los problemas de scheduling que siguieron existiendo.

Durante la última década, conjuntamente con el abismal desarrollo de sistemas informáticos para empresas, se comprendió que los problemas de programación de la producción no son más que un conjunto de restricciones

que determinan un espacio factible de soluciones. Ese espacio puede recorrerse de diversas maneras (algoritmos, heurísticas, programación lineal) y los ordenadores pueden hallar buenas soluciones – a veces óptimas – mediante un esfuerzo razonable. De esta manera, el enfoque imperante a partir del año 2000 consiste en dejar el problema **en manos de las computadoras**.

Si bien las soluciones halladas son significativamente mejores a las obtenidas con las metodologías anteriores, algunos autores entienden que la principal desventaja de este enfoque reside en la ausencia de **aprendizaje real** sobre cómo resolver un problema de scheduling. Es decir, aún cuando un software puede hallar soluciones convenientes, las personas no comprenden el porqué de dichas soluciones ni desarrollan capacidades para abordar problemas del mismo tipo que puedan presentarse en el futuro.

II.1.d – Síntesis y Cronología

Luego de algo más de un siglo de historia en la resolución de problemas de programación de la producción, se puede decir que se cuenta con buenos desarrollos concernientes al tema, muchos de ellos provenientes de diversas corrientes metodológicas.

Puede observarse que si bien el scheduling es un problema establecido y reconocido como tal, en todas las épocas han existido paradigmas que lo excluían de la agenda de las empresas por

considerarlo una parte menor de un problema más completo, una

disquisición inútil e intrincada o un problema sencillo gracias al avance de los métodos de producción, respectivamente (*véase Figura 6*).

Además de contextualizar el problema y la solución propuesta en el presente trabajo, este recorrido histórico también es rico desde el punto de vista de la definición de metodologías, ya que permite comprender la forma y secuencia en la que surgieron las distintas técnicas que se combinarán y analizarán a lo largo de este proyecto.

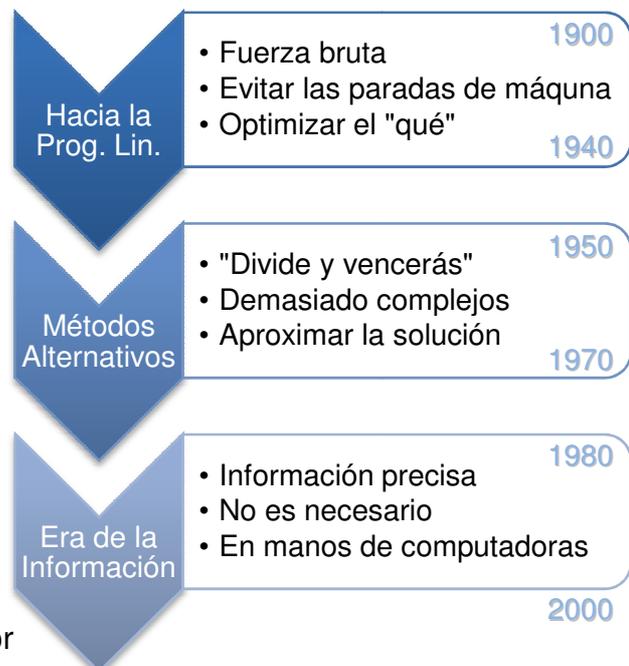


Figura 6 – Síntesis de la evolución de enfoques

II.2 ♦ CLASIFICACIÓN DE TÉCNICAS MODERNAS PARA LA RESOLUCIÓN DEL PROBLEMA

II.2.a – Un Posible Mapa De Metodologías

Para lograr dar una solución válida al problema planteado, se ha recorrido la evolución histórica de los distintos enfoques empleados por las empresas durante más de 100 años. Ahora, se propone un resumen transversal para comprender los resultados de dicha evolución: las técnicas que se manejan hoy en día para abordar el problema de scheduling.

No se pretende realizar una clasificación taxativa ni exhaustiva de todas las metodologías existentes al día de la fecha, sino agrupar aquellas que más relevancia tengan para el presente trabajo de manera conveniente para su estudio y análisis. Con dicho objetivo, se propone el siguiente mapa resumido de metodologías⁴ (véase Figura 7).

Se observan 3 grupos principales de metodologías bien diferenciados que se analizarán en los próximos apartados de esta sección. Vale remarcar entonces la diferencia conceptual entre estas ideas centrales.

Un **algoritmo** es, en resumidas cuentas, un procedimiento probado para resolver cierto problema mediante una serie de pasos establecidos.



Figura 7 - Mapa de metodologías

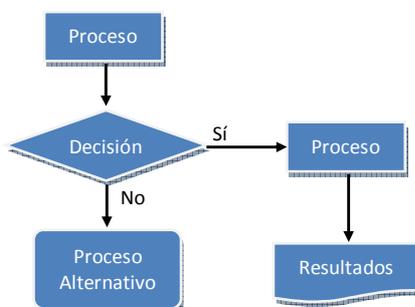


Figura 8 - Diag. de flujo

Usualmente, la secuencia de operaciones y las decisiones involucradas en la resolución del problema pueden esquematizarse en un diagrama de bloques o un diagrama de flujo (véase Figura 8). De esta manera, se puede comprender el mecanismo a aplicar para obtener los resultados esperados de forma sistemática.

⁴ La versión completa del mismo puede encontrarse en el *Anexo VIII.2.*

En los casos que el problema a resolver se vuelve demasiado complejo, puede ocurrir que se hallen “reglas” o “pasos” que lleven a una buena solución, pero sin las formalidades matemáticas requeridas como para llamarlos algoritmos. De esta manera, las **heurísticas** (y metaheurísticas) son usualmente empleadas para lograr soluciones aceptables en tiempos razonables, pero no poseen las clásicas propiedades que se exige demostrar en el caso de los algoritmos, a saber:

- ◆ Tiempo de corrida acotado y conocido (al menos su valor máximo)⁵.
- ◆ Grado de optimalidad de la solución (o “cercanía” al óptimo)⁶.

Entonces, se puede tratar a las heurísticas de manera similar a los algoritmos, sabiendo que son soluciones menos estudiadas y confiables; a su vez, tienen la ventaja de ser más generales en buena parte de los casos.

Finalmente, la **simulación** aborda los problemas desde un ángulo diferente, intentando imitar el comportamiento del sistema real durante un período definido de tiempo.⁷ El primer beneficio de este enfoque es el gran avance en la comprensión del problema que genera desarrollar y validar un modelo. Muchas veces durante el proceso de modelización pueden hallarse puntos en los cuales apalancar la solución por el mero acto de conceptualizar y sintetizar el problema en sí. Además, el método puede emplearse tanto para realizar un análisis sobre el sistema (generalmente orientado a problemas con interacción entre variables y efectos estocásticos, que no pueden resolverse por técnicas directas) como para intentar una optimización del modelo para luego implementarla en el sistema real.

II.2.b – Algoritmos de Resolución

Los algoritmos de resolución para problemas de scheduling han sido desarrollados principalmente en base al paradigma “**divide y vencerás**”, ya mencionado en la sección anterior. Los mismos pueden dividirse en dos grandes grupos: los exactos y los de aproximación (*véase Figura 9 en página siguiente*).

⁵ Para evitar problemas de tiempo de corrida infinito, generalmente se hace trabajar a las heurísticas durante un tiempo predefinido y luego se les pide que devuelvan la mejor solución hallada en ese período. Si bien es una solución práctica, este hecho le resta eficiencia al procedimiento dado que es abortado abruptamente sin considerar momentos cruciales durante la corrida (como puede ser el fin de un ciclo de movimientos en una heurística local).

⁶ Por esta razón, usualmente se acepta la solución de la heurística y se la compara con otros métodos, sin intentar saber cuán alejada del óptimo se halla la misma; se usa un enfoque práctico en vez de intentar una prueba teórica.

⁷ En el *Anexo VIII.4* se encuentra una ampliación del concepto de simulación.



Figura 9 - Mapa de algoritmos de resolución

Los **algoritmos exactos** son aquellos que, mediante una serie de pasos, logran hallar soluciones óptimas al problema planteado en un tiempo definido. Su principal ventaja es que garantizan que la solución hallada es la óptima, pero entre sus desventajas se cuentan largos períodos de corrida, imposibilidad de considerar ciertas restricciones, necesidad de modificar el problema para adaptarlo a sus reglas de resolución e imposibilidad de trabajar con múltiples objetivos.

Las principales características que requieren los problemas para ser tratados con ayuda de los algoritmos exactos son:

- ◆ Relativamente pocas variables de decisión
- ◆ Objetivo fácil de calcular y único
- ◆ Con pocas o sin restricciones
- ◆ Determinísticos

Uno de los primeros algoritmos exactos desarrollados en relación con la resolución de problemas de scheduling fue el Algoritmo de Johnson, mencionado anteriormente y desarrollado en el *Anexo VIII.1*. Mediante la comprensión de dicho algoritmo se pueden inferir las principales características de los algoritmos exactos “para **casos particulares**”.

En general, todos los algoritmos exactos – exceptuando la programación lineal y la programación dinámica – requieren de una enorme simplificación del problema de scheduling para poder ser aplicados. Por tal razón, su principal interés es a nivel académico. Pueden emplearse en algunas situaciones para comparar soluciones reales con la solución óptima del sistema idealizado, pero no será el caso de este trabajo.

La **programación dinámica** es un caso interesante de algoritmo exacto, ya que por su naturaleza puede atacar problemas más generales⁸. Esta técnica llega a la solución trabajando hacia atrás, partiendo del final del problema hacia el principio y subdividiendo el problema en etapas con ciertas características.

La programación dinámica no cuenta con un algoritmo matemático estándar, sino que plantea una estrategia general de solución de problemas y aporta un procedimiento sistemático para encontrar la combinación de decisiones parciales que optimice el resultado integral.

Las características principales de los problemas en donde puede aplicarse la programación dinámica son:

- ▶ Es posible dividir el problema en etapas, y se requiere una decisión en cada etapa.
- ▶ Cada etapa se relaciona con una cierta cantidad de etapas.
- ▶ En cada etapa se cuenta con posibles estados, información que se necesita para tomar una decisión óptima.
- ▶ La decisión tomada en cualquier etapa describe el modo en que el estado en la etapa actual se transforma en el estado en la etapa siguiente.
- ▶ Dado el estado actual, la decisión óptima para cada una de las etapas restantes es independiente de los estados ya alcanzados o de las decisiones tomadas previamente.

Esta metodología es utilizada para resolver problemas de asignación de recursos, inventarios y redes.

La **programación lineal** tiene el rango de acción más amplio dentro de los algoritmos exactos y, por tal razón, es una de las técnicas más empleadas en la resolución de problemas de planificación a nivel académico e industrial. Debido a este lugar de privilegio, esta técnica será tratada en el apartado siguiente (*véase II.3*).

Los **algoritmos de aproximación** surgen básicamente por la necesidad de resolver problemas de gran complejidad en tiempos razonables. Dado que hoy en día no existen algoritmos que solucionen óptimamente problemas de un grado de dificultad definido – llamado NP-hard – en un tiempo “suficientemente corto” – llamado tiempo polinomial o polinómico⁹ – se emplean algoritmos aproximados que logran dar una solución con error acotado en un tiempo

⁸ El resto de la descripción del algoritmo está adaptado de [5]

⁹ Básicamente, el tiempo polinómico de un algoritmo implica que su tiempo de corrida depende polinómicamente de su número de variables (y no exponencialmente).

mucho menor. Son un paso intermedio hacia las heurísticas, pero estos algoritmos sí cuentan con las demostraciones matemáticas que permiten acotar el error respecto del óptimo y probar que su tiempo de corrida es polinómico.

Una de las formas más habituales de llamar a los algoritmos de aproximación es **algoritmos de factor constante**, que representan aquellas metodologías que garantizan estar a una distancia porcentual constante del óptimo. Por ejemplo, si se busca minimizar algo y se usa un algoritmo de factor 2, se sabe que, cuando mucho, el valor de la función objetivo será el doble que el valor óptimo. Cabe aclarar que usualmente su performance es significativamente mejor que la que asegura la prueba matemática. El algoritmo trabaja principalmente con la relajación del problema de programación lineal (es decir, asumiendo que todas las variables son continuas) y restringe el problema de diversas maneras para especificar esa distancia al óptimo.

Estos algoritmos han ganado terreno en el ámbito de la informática durante el último tiempo; existen versiones que hasta permiten resolver problemas de scheduling,¹⁰ pero no logran incorporar buena parte de las restricciones que requieren los problemas reales. En algún punto, poseen los mismos inconvenientes que los algoritmos para casos particulares, aunque su importante avance hace pensar que irán perfeccionándose y en pocos años se podrá modelar un problema completo de scheduling y resolverlo con un margen de error acotado mediante estas técnicas. Aún así, su idoneidad para el tipo de problema podría discutirse (*véase VI.4*).

II.2.c – Heurísticas de Resolución

Las heurísticas permiten mayor flexibilidad a la hora de definir las restricciones y objetivos de un problema sacrificando algo de calidad en las soluciones; es decir, si bien no garantizan que la solución hallada sea óptima, resuelven un problema más cercano a la realidad.

Remitiéndose a las etapas del proyecto (*revéase I.4*), las heurísticas mejoran la **modelización** y **validación** con respecto a los algoritmos, ya que permiten que el modelo refleje la realidad de una forma más fiel; sin embargo, pierden capacidad de **optimización**, debido a que no logran asegurar la optimalidad de la solución ni acotar su distancia al óptimo. El resultado global de esta ganancia y pérdida suele ser positivo para el proyecto, ya que es mejor resolver bien un problema cercano a la realidad que resolver perfecto un problema alejado de la práctica. Por tal razón, se analizarán algunas de las

¹⁰ Para mayor información refiérase a [26]

heurísticas que mayor relación con el problema de scheduling tienen para emplearlas en el desarrollo del proyecto.

El campo de las heurísticas puede organizarse básicamente en dos grupos bien diferenciados: las heurísticas de construcción y las de mejoramiento (o búsqueda). Dentro de las heurísticas de mejoramiento pueden distinguirse las globales y las locales. Esta clasificación puede apreciarse gráficamente en la *Figura 10* con algunas sub-categorías y los ejemplos que serán abordados en el presente apartado.

La gran diferencia entre las heurísticas de construcción y de mejoramiento es que las primeras evalúan una sola alternativa de solución, la “construyen” mediante reglas, mientras que las de mejoramiento parten de una semilla e intentan “mejorar” esa solución hasta llegar a un pseudo-óptimo.

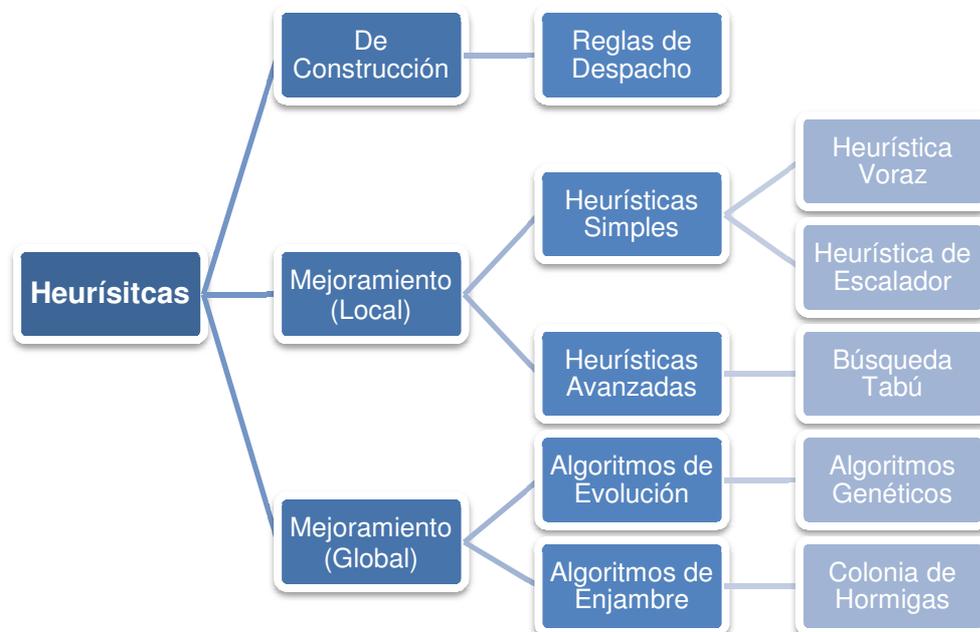


Figura 10 - Mapa de heurísticas de resolución

Dentro de las heurísticas de construcción y refiriéndose al problema de scheduling, sólo pueden mencionarse las llamadas **reglas de despacho**. Esta metodología consiste en definir reglas sencillas que se aplican para generar una solución. La técnica se utiliza frecuentemente desde la experiencia; el enfoque de **evitar las paradas de máquina**, tratado anteriormente, es un ejemplo algo rústico de heurística de construcción.

Entre las principales reglas de despacho empleadas para resolver problemas de scheduling se pueden mencionar:

- ◆ **ERD o FCFS:** Earliest Release Date (fecha de lanzamiento más temprana) o First Come First Served (primero en llegar, primero en ser atendido). Se aplica en problemas muy sencillos como podría ser una máquina que corta un producto estándar, y sólo sirve para ser “justo” con los distintos productos. Requiere que se cuente con los momentos de llegada de las órdenes, hecho que a veces no ocurre o es irrelevante.
- ◆ **SPT:** Shortest Processing Time (el menor tiempo de procesamiento). Se trata de procesar primero las órdenes que menor tiempo lleven minimizando de esta forma el tiempo promedio de espera¹¹. Puede emplearse con dicho objetivo en problemas de una etapa.
- ◆ **LPT:** Longest Processing Time (el mayor tiempo de procesamiento). Al contrario de la regla anterior, ésta sirve para asegurar trabajo continuo a una máquina evitando las paradas, pero penaliza fuertemente las esperas promedio.
- ◆ **ECT:** Earliest Completion Time (el menor tiempo para completar la tarea). Esta regla prioriza siempre los trabajos que requieran un menor tiempo para ser completados, de modo de adelantar todo lo posible la salida de productos terminados. Es útil en situaciones de emergencia donde deben completarse órdenes sin importar la eficiencia de la planta.
- ◆ **WSPT:** Weighted Shortest Processing Time (el menor tiempo de procesamiento ponderado). Se pondera mediante algún parámetro relevante para la empresa y se elige el menor tiempo de procesamiento ponderado, de modo de intentar cumplir con la combinación de dos criterios: el de STP y uno externo que fija la empresa.
- ◆ **EDD:** Earliest Due Date (fecha de entrega más temprana). Consiste en hacer primero lo que se debe entregar primero, con vistas a cumplir con las fechas pactadas de la mejor manera posible.
- ◆ **Reglas de Despacho Compuestas:** consisten en combinar las anteriores mediante un sistema de ponderación. Se elabora una lista de prioridades según las distintas reglas y luego se pondera la posición en las distintas listas para obtener una lista global de la regla compuesta. Permite aplicar varios de los criterios mencionados en una medida controlada.

¹¹ Para mayor detalle, véase VIII. 1.

Las **reglas simples** pueden ser útiles para tratar problemas de scheduling sencillos y con un objetivo bien definido. Por ejemplo, EDD es una buena política que garantiza el mejor cumplimiento de fechas de entrega en sistemas cuasi-ideales de una sola etapa. Sin embargo, es fácil demostrar que por más que exista una sola máquina, la regla no asegura el mejor cumplimiento en casos que existan setups matriciales (es decir, en donde el tiempo de configuración de la máquina depende tanto de la operación precedente como de la posterior al setup). Por tal razón, las reglas de despacho simples no se emplean hoy en día para resolver problemas de scheduling; sin embargo, en algunas ocasiones se usan para programar dinámicamente algunas etapas o máquinas particulares.

Las **reglas compuestas** tienen alguna aplicación industrial en lo que a programación de la producción se refiere pero presentan el gran inconveniente de que, al ponderar objetivos, generalmente la solución propuesta por la heurística no conforma a ninguno de ellos y usualmente está muy alejada del óptimo. Además, el programador suele encontrar grandes dificultades en percibir como su ponderación afecta los resultados que ofrecerá el método.

Las **heurísticas de mejoramiento** trabajan variando la solución actual para intentar aumentar el funcional (en caso de maximización) de una forma “inteligente”, ya que si no caerían en un procedimiento exhaustivo. Su procedimiento se basa justamente en mejorar la solución factible en reiteradas ocasiones. Particularmente, las **heurísticas locales** parten de una solución semilla e intentan variarla levemente para lograr mejorar el funcional, de forma que su alcance es usualmente local; llegan sólo hasta las cercanías de la solución semilla.

Dentro de estas heurísticas locales, encontramos dos ejemplos muy básicos que permiten entenderlas conceptualmente y que se han agrupado como **heurísticas simples** en el mapa de la *Figura 10*. Una de ellas es la **heurística voraz**, que busca simplemente aplicar la mejor solución posible en cada etapa sin contemplar el sistema en su conjunto. Por ejemplo, si se busca minimizar el recorrido del famoso problema del agente viajero, una heurística voraz intentará ir a la ciudad más cercana no visitada en todos los casos. Puede emplearse la idea de la heurística tanto para construir soluciones como para mejorarlas.

El caso que mejor ilustra el concepto de búsqueda local es sin dudas la **heurística del escalador**. Conceptualmente, lo que hace el algoritmo es simplemente moverse en la dirección que crece la función objetivo, de forma similar a un alpinista escalando una montaña. Evalúa el funcional y las

variaciones que tendría al moverse en las distintas direcciones de todas las variables de decisión, y se mueve hacia el lugar de máxima pendiente creciente. De esta forma, logra llegar rápidamente el máximo local y luego se queda sin más para aportar a la optimización.

Es relativamente simple ponerla en práctica, haciéndola una primera opción popular. Aunque algoritmos más avanzados puedan dar mejores resultados, en algunas situaciones la heurística del escalador puede funcionar también. Entre sus principales aplicaciones se pueden mencionar los problemas de redes, asignación de recursos, inteligencia artificial y ruteo.

Aún en el ámbito de la búsqueda local, existen **heurísticas más complejas** que intentan evitar el problema principal de estas técnicas: los atascos en los máximos locales. Una de ellas es la **búsqueda tabú**, que toma de la **Inteligencia Artificial** el concepto de memoria y lo implementa mediante estructuras simples, con el objetivo de dirigir la búsqueda teniendo en cuenta la historia de ésta.

La búsqueda tabú puede considerarse como una meta-heurística que trabaja básicamente monitoreando las soluciones por las que lleva a la heurística local. Confecciona una “lista tabú” de soluciones por las que no puede regresar para evitar el atascamiento en máximos locales, y genera reglas que guían a la heurística local en la búsqueda de mejores soluciones. Dicha lista es modificada dinámicamente según las nuevas soluciones halladas, sus reglas específicas y la cantidad de elementos (usualmente se guardan los últimos **n** elementos como tabú, siendo **n** un valor fijo). Sus principales aplicaciones se centran en la asignación de horarios y tareas para diversas industrias.

Esta metodología constituye un híbrido entre las ya mencionadas heurísticas de búsqueda local y las de búsqueda global. Las **heurísticas de búsqueda global** son aquellas que, si bien trabajan variando la solución para mejorar el funcional como las anteriores, realizan grandes cambios en las variables, explorando de forma más completa el espacio de soluciones factibles. Dentro de las mismas, se distinguen dos categorías que dependen básicamente del origen que tenga la idea madre que da lugar al algoritmo. Si las ideas tienen una procedencia evolutiva, se denominan **algoritmos evolutivos**; si las ideas se toman del comportamiento emergente de grupos de seres vivos, se denominan **algoritmos de enjambre** (existen también otras categorías no tratadas en este trabajo).

Los **algoritmos genéticos**, el ejemplo más difundido de algoritmos evolutivos, establecen una analogía entre el conjunto de soluciones de un

problema y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, llamada **cromosoma**. Los símbolos que forman la cadena son llamados los **genes**. Los cromosomas evolucionan a través de iteraciones, llamadas **generaciones**. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud. Las siguientes generaciones, llamadas **descendencia**, se forman utilizando dos operadores, de **cruce** y de **mutación**.

Un algoritmo genético puede presentar diversas variaciones, dependiendo de cómo se aplican los operadores genéticos (cruces, mutación), de cómo se realiza la selección y de cómo se decide el reemplazo de los individuos para formar la nueva población. En general, el pseudocódigo se compone de los siguientes pasos (véase *Figura 11*):

- [1] **Inicialización:** se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones al problema. En caso de no hacerlo aleatoriamente, es importante garantizar que dentro de la población inicial, se tenga la diversidad estructural de estas soluciones para tener una representación de la mayor parte de la población posible o al menos evitar la convergencia prematura.
- [2] **Evaluación:** a cada uno de los cromosomas de esta población se aplicará la función de aptitud para saber qué tan buena es la solución que se está codificando. Básicamente, lo que se hace es evaluar todas las soluciones con el funcional.
- [3] **Selección:** después de saber la aptitud de cada cromosoma se eligen los cromosomas que serán cruzados en la siguiente generación. Los cromosomas con mejor aptitud tienen mayor probabilidad de ser seleccionados.
- [4] **Cruzamiento:** el cruzamiento es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres. Matemáticamente lo



Figura 11 - Diagrama de flujo de un A. G.

que hace esta operación es combinar ambas soluciones dando lugar a una solución nueva, distinta de sus progenitoras, que tenga algunas características de cada “padre”.

- [5] **Mutación:** modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual. Este paso es fundamental para lograr que el algoritmo no termine en una heurística local.
- [6] **Reemplazo:** una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la generación siguiente.
- [7] **Condición de Término:** el algoritmo genético se deberá detener cuando se alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se deben utilizar otros criterios de detención. Normalmente se usa alguno de los siguientes criterios (o ambos): correr el algoritmo un número máximo de iteraciones (generaciones) o detenerlo cuando no haya cambios en la población.

Los algoritmos genéticos son de probada eficacia en caso de querer calcular funciones no derivables (o de derivación muy compleja) aunque su uso es posible con cualquier función. Entre sus principales aplicaciones industriales se pueden mencionar: diseño automatizado, optimización de carga de contenedores, planificación de producción multicriterio, optimización de layout, construcción de horarios en grandes universidades y problemas de ruteo.

Finalmente, dentro de los algoritmos de enjambre, se puede mencionar uno de sus exponentes más reconocidos por sus aplicaciones en los últimos 15 años: el algoritmo **Colonia de Hormigas** (*Ant Colony en inglés*), que es básicamente una técnica probabilística utilizada para solucionar problemas de cómputo. Este algoritmo está inspirado en el comportamiento que presentan las hormigas para encontrar las trayectorias desde la colonia hasta el alimento.

En la naturaleza, las hormigas vagan aleatoriamente en su búsqueda de alimento, y a lo largo de su camino de regreso a la colonia depositan una hormona denominada feromona. Si otras hormigas encuentran este rastro, lo más probable es que sigan esta trayectoria para depositar el alimento en la colonia. Con el paso del tiempo el rastro de la feromona comienza a evaporarse y se reduce su fuerza atractiva. Las hormigas que siguen el rastro refuerzan la intensidad de la feromona, con lo que logran que perdure por más tiempo. Mientras más hormigas recorren un mismo camino, más intenso es el olor del rastro, lo que estimula a más hormigas a seguir esa trayectoria.

Desde el punto de vista algorítmico, la evaporación de la feromona tiene la ventaja de evitar la convergencia a una solución localmente óptima. Si no hubiera evaporación, todas las trayectorias serían igualmente atractivas para las hormigas. Esta situación haría que la exploración del espacio de la solución fuese demasiado amplia. Así, cuando una hormiga encuentra un buen camino de la colonia a una fuente de alimento (entiéndase corto), es más probable que otras hormigas lo utilicen también. La razón es simplemente que entre una trayectoria corta y una larga la primera tiene una mayor cantidad de recorridos por unidad de tiempo, con lo cual la feromona atrae más fuertemente a otras hormigas. Este comportamiento es la base para el diseño del algoritmo, donde las "hormigas simuladas" caminan alrededor del gráfico que representa el problema a solucionar.

Vale remarcar que a principios de la presente década se probó matemáticamente la convergencia de **Colonia de Hormigas**, con lo cual su *status* de heurística se vio aumentado, sin llegar a ser realmente un algoritmo pero con un grado de formalización mayor al de la mayoría de las heurísticas mencionadas. Por tal razón, generalmente se lo llama **Algoritmo de Hormigas**.

Los algoritmos de optimización de Colonia de Hormigas se han utilizado para producir soluciones cuasi-óptimas en el problema del agente viajero. Otras de sus aplicaciones cubren problemas de ruteo, de redes y de sistemas urbanos del transporte. Los primeros usos de la heurística en scheduling se remontan al año 2000; su aplicación se dio mayormente en problemas con fuerte inclinación a temas espaciales y de layout.

II.2.d – Simulación Como Herramienta De Resolución

La simulación presenta un enfoque relativamente reciente para el problema de programación de la producción. Los primeros desarrollos en el tema se realizaron hace unas dos décadas y la popularización de la técnica no llegó hasta varios años después. Se pretende, en la presente tesis, atacar el problema de scheduling mediante una metodología novedosa – aunque no experimental – por lo que este enfoque resulta muy interesante a tal fin.

Los usos de modelos de simulación para analizar y resolver este tipo de problemas pueden agruparse principalmente en dos categorías (*véase Figura 12 en página siguiente*).



Figura 12 - Mapa de metodologías aplicando simulación

Por un lado, se puede usar el modelo básicamente como un funcional avanzado y más complejo; es decir, se crea un modelo que refleje las restricciones necesarias y se usa para evaluar alternativas de solución, devolviendo un valor definido para cada una de las opciones (por ejemplo, la suma del tiempo de procesamiento total de todas las órdenes). El modelo pasa de ese modo a ser un evaluador de soluciones factibles, tal como el funcional en programación lineal. En estos casos, las técnicas que pueden usarse para optimizar no difieren de las discutidas en apartados anteriores; se trata de las mismas **heurísticas de optimización**. Un ejemplo interesante es el de la **búsqueda tabú**, que por sus características y velocidad puede emplearse para guiar a la solución con el auxilio de la evaluación mediante un modelo.

Por otra parte, el modelo de simulación puede emplearse para analizar el sistema, principalmente cuando los tiempos y operaciones involucrados no son determinísticos, sino que presentan ciertas distribuciones de probabilidad que pueden averiguarse o estimarse. En estos casos, los resultados del sistema son inciertos y de muy difícil – sino imposible – cálculo directo, por lo que un modelo permite observar lo que sucede realmente en la planta mediante un **análisis estocástico**. De esta forma, no se emplea al modelo como optimizador sino como verificador y elemento de corrección, que emula la planta y los resultados que se obtendrán si se aplica una u otra política.

II.3 ♦ LA TÉCNICA MÁS POPULAR: PROGRAMACIÓN LINEAL

En 1947 George Dantzig desarrolló un método efectivo para resolver problemas de programación lineal, el **algoritmo simplex**¹². Desde que surgió dicho algoritmo, la programación lineal se utiliza para resolver problemas de optimización en diversas industrias, tales como bancos, educación, silvicultura, petróleo y transporte de carga. Este algoritmo cuenta con los siguientes elementos:

- ▶ **Función objetivo:** la función que se desea maximizar o minimizar, también llamada “funcional”.

¹² La parte teórica de esta sección está basada en [5].

- ▶ **Variables de decisión:** son las variables que describen por completo las decisiones que se tienen que tomar. Se trabaja con los valores de dichas variables en busca de la combinación que optimice el funcional o la función objetivo.
- ▶ **Coeficientes de costo:** son los coeficientes que acompañan a las variables de decisión en la función objetivo. Estos coeficientes pueden representar simplemente las contribuciones de las variables a la utilidad de la compañía, o el costo de esas variables si se busca minimizar el funcional.
- ▶ **Restricciones:** los valores de las variables están controlados por ciertas limitaciones. Esas restricciones deben expresarse en función de las variables de decisión y todos los términos de la restricción deben tener las mismas unidades. Las restricciones pueden ser del tipo \geq , \leq o $=$. A su vez, cada variable tiene una restricción de signo. En la mayoría de los casos esta restricción obliga a que la variable pueda tomar sólo valores no negativos.
- ▶ **Coeficientes tecnológicos:** son los coeficientes de las variables de decisión en las restricciones. Se los denomina así porque a menudo reflejan las tecnologías utilizadas para producir distintos productos.

Además, la resolución por programación lineal asume los siguientes supuestos:

- ▶ La contribución a la función objetivo de cada variable de decisión es proporcional al valor de ésta.
- ▶ La contribución a la función objetivo de cualquier variable es independiente de los valores de las otras variables de decisión.
- ▶ La contribución de cada variable al primer miembro de cada restricción es proporcional al valor de la variable.
- ▶ La contribución de cada variable al primer miembro de cada restricción es independiente de los valores de las otras variables.
- ▶ Todos los parámetros del modelo (básicamente los coeficientes de la función objetivo y coeficientes tecnológicos) son conocidos con certeza; el modelo no admite distribuciones de probabilidad.

Como puede apreciarse, la programación lineal es un algoritmo que permite resolver un amplio rango de problemas, siempre y cuando cumplan ciertas condiciones. Lo interesante es que, a diferencia de las heurísticas o algoritmos de aproximación mencionados, otorga soluciones óptimas – puede demostrarse la optimalidad de la solución. De esta forma, logra garantizar que la solución que devuelva será tan buena como pueda ser.

El “costo” de esta condición de óptimo también es importante: requiere que el problema se ajuste a ciertas características y supuestos mencionados, posee un tiempo de procesamiento que puede ser excesivo en problemas de tamaño considerable, exige un único funcional, linealidad en la función objetivo, linealidad en las restricciones y comparte otras desventajas con los **algoritmos exactos** (ya que, si bien es una metodología muy general, no deja de ser un algoritmo exacto).

Entonces, teniendo en cuenta la descripción y consideraciones realizadas, ¿por qué es la técnica más popular para resolver problemas de scheduling?

La respuesta es que combina una serie de factores que la hacen atractiva a la hora de resolver problemas de programación de la producción:

- ◆ Su comprensión es simple.
- ◆ Su codificación es sencilla.
- ◆ El algoritmo de resolución figura en cualquier libro de investigación de operaciones y está programado en decenas de paquetes de software.
- ◆ Los parámetros de planta generalmente poseen una variación acotada y pueden modelarse como coeficientes determinísticos.
- ◆ Se cumplen prácticamente siempre los supuestos de independencia y aditividad de variables (salvo tal vez en reacciones químicas donde no se conserva la masa).
- ◆ Otorga una solución siempre óptima, con lo cual evita las dudas en cuanto a si se podría mejorar modificando la programación de forma manual.

Todas estas características llevan a que la primera opción a considerar a la hora de programar la producción sea el famoso **algoritmo simplex**. Sin embargo, se deben tomar varias decisiones más para definir completamente un esquema o metodología de resolución en base a dicho algoritmo.

La programación lineal puede emplearse básicamente de dos maneras para aportar a la solución del problema. La primera de ellas es la desarrollada a partir del paradigma **optimizar el “qué”**, comentado anteriormente. En este caso, lo que se plantea es definir mediante el algoritmo qué producir y, a menudo, dónde, pero no cuándo. Se genera un modelo que prescinde de la variable tiempo, y que permite definir qué trabajos conviene realizar en cada máquina, sin considerar el orden. Basándose en las capacidades de las máquinas, rendimientos, tamaño de las órdenes, algún sistema de prioridades quizás y el listado de órdenes a producir con sus especificaciones, se genera

un modelo relativamente sencillo que resuelve el problema de la asignación. Luego, se deja el problema del secuenciamiento dentro de cada máquina a otra técnica o regla, que en principio es un problema mucho más sencillo que el anterior. De todas maneras, debe observarse que la solución de scheduling en estos casos puede no ser óptima; lo óptimo es la solución al problema de asignación. Sin embargo, suele aportar buenas alternativas con un costo muy reducido, ya que el modelo es simple.

Por otro lado, una aplicación más avanzada de la programación lineal consiste en generar un modelo que, además de las variables mencionadas, considere precedencias, etapas y fechas de entrega. Estos condicionamientos obligan a trabajar con la variable tiempo dentro del modelo, lo que complejiza enormemente tanto su codificación como su resolución (puede requerir un software más avanzado y tardar hasta horas). En contrapartida, se obtiene una resolución óptima a todo el problema de scheduling. Este tipo de modelos es más costoso y, por las limitaciones propias de la programación lineal, a veces impide la codificación de todas las restricciones como para que el problema sea resuelto completamente; es decir, puede ser que se esté resolviendo óptimamente un problema ligeramente distinto al real.

Sea de una u otra forma, un hecho de fundamental importancia a la hora de definir un problema de scheduling por programación lineal es la función objetivo. Dado que usualmente la programación de la producción persigue varios objetivos simultáneamente y el algoritmo de programación lineal admite sólo una función objetivo, se debe seleccionar cuidadosamente como estructurarla, ya que será determinante en los resultados que arroje la metodología. Entre los objetivos más usuales se encuentran:

- ▶ Minimizar los costos de producción
- ▶ Minimizar el lead time
- ▶ Maximizar la utilización de las máquinas
- ▶ Minimizar los tiempos ociosos de las máquinas
- ▶ Maximizar la salida de productos terminados

Como función objetivo de la programación lineal se puede emplear una de las alternativas analizadas o una combinación de algunas de ellas, entrando en la **programación lineal por objetivos múltiples**. Esta técnica simplemente pondera la gravedad de los incumplimientos en distintos objetivos y trata de minimizar la suma de las penalizaciones, con lo cual el sistema termina ponderando y combinando las distintas metas.

Lo dicho en cuanto al análisis de la función objetivo tiene un carácter general aplicable a todas las metodologías de optimización; se incluyó en este

apartado por ser especialmente evidente su influencia en la técnica de programación lineal.

II.4 ♦ CASOS RECIENTES DE APLICACIÓN DE LOS PRINCIPALES MÉTODOS

En esta sección se propone realizar un breve comentario acerca de casos concretos que muestren la dirección que tienen las investigaciones en el tema en la actualidad. Los objetivos de su inclusión son principalmente dos: por un lado, reforzar la síntesis de metodologías realizada en los apartados anteriores; por otro, comprender algunas de las ideas que dominan la escena del scheduling al día de hoy en pos de resolver el problema tomando conceptos y elementos de diversas técnicas.

En primer lugar, es interesante el caso que plantea el autor Gonca Tuncel en una publicación de Diciembre del año 2007 titulada “**Una Aproximación Heurística Basada en Reglas para la Programación de la Producción Dinámica en Sistemas de Manufactura Flexible**”¹³.

El informe se centra en los problemas para definir una buena metodología de scheduling cuando las condiciones del entorno son cambiantes. El autor deduce que debido a las fuertes restricciones y a la variabilidad inherente a este tipo de problemas, la decisión más efectiva es aplicar reglas de despacho. Sin embargo, no se conforma con analizar las reglas clásicas, mencionadas al inicio de *II.2.c*.

El autor define una regla heurística a la medida del problema que logra una performance significativamente superior a la del resto de las reglas (aproximadamente un 30% superior en los indicadores comparados). Estos resultados se obtienen de la aplicación de distintas reglas a un modelo de simulación estocástico que imita las condiciones variables de los sistemas flexibles de manufactura.

El caso es rico desde la perspectiva de esta tesis porque aplica dos de los enfoques tratados con anterioridad para resolver un mismo problema. Por un lado, genera una heurística de construcción y la compara con otras heurísticas de construcción conocidas; por el otro, emplea el análisis por simulación mencionado para evaluar dichas heurísticas en condiciones estocásticas.

¹³ El título original es “A Heuristic Rule-Based Approach for Dynamic Scheduling of Flexible Manufacturing Systems”, corresponde a la referencia [22].

El segundo caso – y último caso a comentar en esta sección – se titula **“Un Modelo de Simulación de Programación de la Producción Restringido por Recursos, Optimizado a Base de Algoritmos Genéticos”**¹⁴. Fue presentado por los autores Sou-Sen Leu y Tzung-Heng Hung en la Universidad Nacional de Taiwan durante el año 2001.

El documento se centra en los problemas de asignación y programación de recursos en la industria de la construcción, bajo evidentes condiciones de variabilidad. Luego de pasar revista a algunas aproximaciones anteriores al tema, describen su modelo basado en dos pilares fundamentales: algoritmos genéticos y simulación.

La simulación que usan es estática, es decir, simula condiciones de variabilidad sin proyectarlas en el tiempo. De esta forma, los autores logran imitar las condiciones de variabilidad (clima y espacio físico, entre otras), relacionarlas, y aplicarlas al modelo.

Por otra parte, emplean un algoritmo genético para el proceso de optimización. El algoritmo genético trabaja testeando soluciones bajo las condiciones de incertidumbre que plantea el modelo de simulación, recogiendo los resultados del mismo y operando en consecuencia.

Este caso representa un excelente ejemplo de lo que anteriormente hemos llamado optimización por simulación. Particularmente, la metodología empleada para optimizar aquí no es una búsqueda tabú sino una búsqueda global, concretamente un algoritmo genético. De esta manera, el algoritmo interactúa con el modelo para “calcular el funcional” de todas las posibles alternativas y efectuar la selección, como se comentó en *II.2.c*.

En la bibliografía figuran otros documentos específicos (*véase VII.2*), en su mayoría publicados en la última década, que ilustran otros casos similares de aplicación de las técnicas mencionadas. La apretada selección que se realizó aquí se basó en que el criterio utilizado en estos dos casos para resolver el problema de scheduling avala el enfoque que se empleará en esta tesis para resolver el problema esbozado (a plantearse concretamente en *III.2*).

¹⁴ El título original es “A genetic algorithm-based optimal resource-constrained scheduling simulation model”, corresponde a la referencia [16].

II.5 ♦ RESUMEN DEL ESTADO DE LA CUESTIÓN

Luego de este recorrido a lo largo y a lo ancho del estado de la tecnología referente al problema de la programación de la producción, puede concluirse que:

- ▶ El problema de scheduling es relativamente reciente pero ha presentado importantes avances en su tratamiento en los últimos años.
- ▶ Para atacar el problema pueden emplearse metodologías muy diferentes que tienen sólo algunos puntos de contacto.
- ▶ No existe ninguna metodología que pruebe tener mejores resultados que el resto; la conveniencia de una u otra técnica depende fuertemente del problema planteado.
- ▶ La programación lineal, por su historia, resultados y sencillez, parece ser una alternativa obligada a la hora de encarar un problema de scheduling.

Como grandes temas pendientes a largo plazo – referentes al problema de scheduling y resultantes del estado actual de la tecnología – se pueden mencionar, principalmente, dos:

- ▶ El desarrollo de metodologías algorítmicas que permitan resultados cuasi-óptimos u óptimos pero que no impongan al problema limitaciones tan severas. Esto aceleraría el regreso a los algoritmos por parte de los que hoy confían sus problemas a una heurística que no puede asegurarles ni siquiera una solución razonable, pero que permite modelizar el problema correctamente.
- ▶ La ampliación del enfoque de simulación al paradigma de **simulación por agentes**, ya que al día de la fecha no se conocen grandes avances en materia de scheduling apoyados en este interesante método.

Estas temáticas quedan fuera del alcance del presente documento, pero serán retomadas hacia el final del trabajo, en *VI.4*.

< III > DEFINICIÓN DEL PROBLEMA

III.1 ♦ CONCEPTOS BÁSICOS Y TERMINOLOGÍA

Para poder definir con precisión el problema y estudiar luego la solución propuesta, hace falta acordar previamente algunos conceptos y terminología referentes al tema. A tal fin, se presenta en este apartado un conjunto de términos tratados en la bibliografía técnica consultada (*véase VII.2*) con sus definiciones más aceptadas y su aplicación al presente trabajo. Vale remarcar que, dado que la enorme mayoría de la bibliografía referente al tema se encuentra publicada en el idioma inglés, algunos de los términos aquí expresados son palabras anglosajonas específicas sin traducción al español.

En primer lugar, hace falta analizar los principales componentes de un problema de programación de la producción. Las **órdenes** por ejemplo, son el input básico de un problema de scheduling, y hacen referencia a los pedidos provenientes de los clientes que pueden contar con algunas de las siguientes especificaciones (u otras tantas):

- | | |
|---------------------|--------------------------|
| ♦ Cliente | ♦ Orden tecnológico |
| ♦ Cantidad | ♦ Insumos necesarios |
| ♦ Tipo de producto | ♦ Lista de materiales |
| ♦ Fecha de la orden | ♦ Tolerancias |
| ♦ Fecha de entrega | ♦ Condiciones de entrega |

El correcto procesamiento de dichas órdenes es la tarea del programador. La mayoría de los atributos de la lista puede comprenderse con facilidad, pero es oportuno destacar la importancia del **orden tecnológico**. El orden tecnológico (o secuencia tecnológica) es la expresión de la ruta que debe seguir la orden por los distintos procesos. Puede expresarse como una lista o vectorialmente (si los procesos están numerados) y representa el “camino” del producto dentro de la planta. Como puede anticiparse, esa secuencia será fundamental para analizar la programación de la producción.

Otro concepto de gran relevancia para el problema de scheduling es el de **centro de trabajo**. Si bien la palabra posee numerosas acepciones, se llamará centro de trabajo – o **centro productivo**, o **centro de procesamiento** – al conjunto de máquinas y operadores que realizan una misma operación (puede ser una sola máquina). Así, el centro de trabajo “inyección” podría ser un conjunto de tres máquinas inyectoras en paralelo que producen envases por inyección en una fábrica de detergentes. En caso de que el orden tecnológico de la mayoría de los productos de la planta sea muy similar y no tenga retornos

significativos (es decir, cada producto pasa sólo una vez por cada centro), el concepto de centro de trabajo se puede asemejar al de **etapa**.

Asumiendo condiciones estándar, para cada pareja {tipo de producto, centro de trabajo} se puede definir un **tiempo de procesamiento**. El tiempo de procesamiento es el tiempo que le tomará a un producto de ese tipo completar su operación en dicho centro de trabajo. En los problemas de scheduling este tiempo suele incluir un estimativo de los transportes internos del producto y el tiempo de máquina. Usualmente se excluyen los tiempos de setup, las esperas en filas por cualquier causa y las contingencias que puedan surgir durante la operación.

Puede deducirse que para n productos y m centros de trabajo, los tiempos de procesamiento pueden ordenarse de forma matricial, dando lugar a una matriz de tiempos de procesamientos que contiene toda la información. Sin embargo, en algunos casos el tiempo de procesamiento es tan variable – aun considerando un único producto y un único centro de trabajo – que su modelización como un valor determinístico se torna irreal; en estos casos se trabaja con **tiempos de procesamiento estocásticos**, resultando en una matriz de distribuciones de tiempos.

Programación de la producción, ordenamiento, sequencing, scheduling, secuenciamiento; todas estas palabras se refieren al problema central de la presente tesis y, en algunos casos, se han usado de manera indistinta a lo largo de los primeros capítulos. Aquí se pretende aclarar algunas diferencias entre estos conceptos en pos de definir el problema de forma más precisa en el apartado siguiente.

Al referirnos a un problema en el cual se debe proponer la mejor manera de procesar un conjunto de órdenes en un conjunto de centros de trabajo, se pueden presentar ciertas dudas en cuanto a la “forma” de la solución, es decir, a la información que debe devolver la metodología para considerar el problema resuelto. Si lo que se busca como respuesta es una lista de trabajos para cada máquina que indique la secuencia de procesamiento de todas las órdenes que deben pasar por ella, se está frente a un problema de **secuenciamiento** (o **sequencing** en inglés). Si, en cambio, la respuesta está integrada por: el momento de inicio de cada trabajo en cada máquina más el momento de finalización de cada trabajo en cada máquina, entonces se tiene un problema de **scheduling** (a veces traducido como **ordenamiento**). Un problema de **programación de la producción** implica la definición de un **programa de producción**, el cual contiene información detallada de las órdenes a realizar por máquina, con todos sus tiempos y recursos asociados.

De las definiciones enunciadas, puede deducirse fácilmente que un problema de programación de la producción contiene, necesariamente, un problema de scheduling; a su vez, un problema de scheduling contiene, sin lugar a duda, un problema de secuenciamiento.

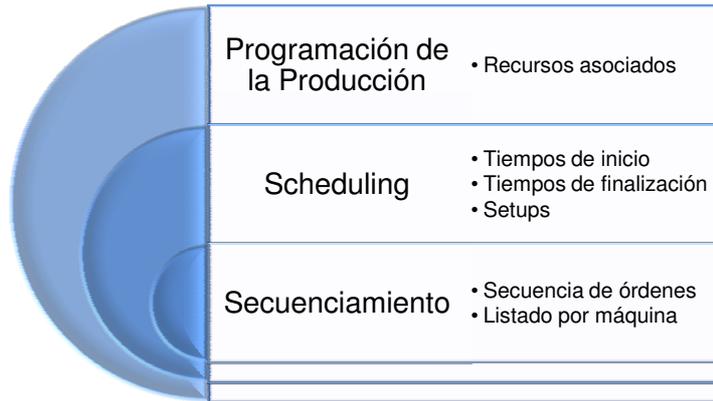


Figura 13 - Distintas formulaciones del problema

En la *Figura 13* se ilustra la situación de estos tres tipos de problema “anidados”. El problema de secuenciamiento es lógicamente el más sencillo de los tres y está contenido dentro de los otros dos. Cuando se consideran los tiempos de setups, precedencias provenientes del orden tecnológico y demás, se halla la respuesta a un problema de scheduling (basada en el secuenciamiento). Si a esa solución se le adicionan los recursos asociados se está frente a un programa de producción, que resuelve un problema más amplio aún.

Si bien los límites parecen claramente definidos, en la práctica estos tres niveles del mismo problema se entremezclan, tal cual sucede con el uso de las tres denominaciones más habituales – mostradas en la *Figura 13* – a menudo empleadas como sinónimos.

Finalmente, es oportuno adelantar aquí la terminología básica de un modelo de simulación empleado para optimización¹⁵. El **modelo de simulación** propiamente dicho es la aplicación o programa que refleja un modelo conceptual previamente discutido y permite imitar el comportamiento de un sistema real durante cierto período de tiempo. El modelo usualmente toma valores de ciertas **variables de entrada (inputs)**, simula, y devuelve valores para ciertas **variables de salida (outputs)**.

Un **experimento** (o **experimento de simulación**) es una forma de introducir las variables de entrada a dicho modelo para analizar sus resultados. Una **corrida** es la simulación de todo el horizonte temporal bajo análisis una única vez. Cuando el modelo posee predominancia de variables estocásticas o cuando se busca optimizar una función objetivo, los experimentos suelen implicar una gran cantidad de corridas.

¹⁵ Para mayor información véase el *Anexo VIII.4*.

III.2 ♦ EL PROBLEMA

El eje de esta tesis – el problema de la programación de la producción en general – ya ha sido objeto de numerosos apartados en la primera parte del trabajo. En esta sección se intentará definir un problema concreto a resolver dentro del contexto ya expuesto.

Dicho problema posee tanto una importancia teórica como una aplicación práctica. La importancia teórica viene dada por la relevancia del problema de scheduling a nivel académico y técnico, mientras que la aplicación práctica se debe a que la resolución del problema constituye la parte central de un trabajo de servicios tecnológicos a una gran empresa.¹⁶

A su vez, la definición precisa del problema surge de la conjunción entre los intereses del autor y los conocimientos aportados por el trabajo realizado para dicha empresa. Sintéticamente expresado, el problema podría definirse como:

“Analizar y definir una metodología de optimización para la programación mensual de una planta, mediante un criterio generalizable a otras plantas y procesos similares.”

Este sencillo problema da lugar a una investigación profunda en técnicas de simulación (motivado por el verbo “analizar”) y al desarrollo de un procedimiento que permita definir correctamente la metodología a seleccionar. Lógicamente, el tema en cuestión es el de programación de la producción, particularmente orientado a una planta de roscado de tubos. Sin embargo, el alcance de la tesis será mayor al de una única planta; se intentará lograr conclusiones generalizables que aporten aunque sea un mínimo avance al tratamiento de estos temas en plantas similares.

III.3 ♦ OBJETIVOS, RESTRICCIONES Y RECURSOS

El principal objetivo del presente trabajo es, en pocas palabras, resolver el problema planteado en el apartado anterior. En una lectura más profunda, deben distinguirse tres metas parciales para solucionar dicho problema:

[1] Analizar correctamente diversas metodologías de optimización factibles para una planta de las características mencionadas

¹⁶ Durante todo este documento se preservará la confidencialidad de la información brindada por la empresa y de los resultados del proyecto. Véase el *Anexo VIII.3*

- [2] Definir una metodología de optimización para dicha planta valiéndose de algún procedimiento racional que permita comparar y seleccionar técnicas
- [3] A partir de un análisis general y un caso particular, lograr conclusiones con cierto alcance, agregando valor al trabajo

Estos tres objetivos son abordados conjuntamente a lo largo de todo el documento, aunque se podría puntualizar que la sección II se centra en el objetivo 1, las secciones IV y V en el objetivo 2 y la sección VI en el objetivo 3. Para el logro de estas metas, este proyecto se apoyará en ciertos recursos e intentará superar las restricciones del sistema.

En buena parte de los casos, las restricciones y los recursos pueden considerarse dos caras de una misma moneda. Por ejemplo: se puede decir que “se dispone de 2 años para terminar la tesis”, considerando al tiempo como un recurso, o que “el tiempo para terminar la tesis no puede excederse de 2 años”, tomándolo como una restricción. En los últimos párrafos de esta sección se intentará mencionar los principales recursos y restricciones del proyecto sin que ello signifique una división taxativa – es decir, algunos recursos son también restricciones y viceversa – sino buscando de alguna manera enmarcar el problema y los objetivos enunciados considerando esta dualidad.

En primer lugar, vale mencionar que una de las **restricciones** fundamentales del proyecto fue el tiempo disponible para modelizar y analizar la planta. Esta labor se llevó a cabo en unas pocas semanas. Si bien el desarrollo de la presente tesis se extendió durante varios meses más, para el trabajo de servicios tecnológicos se contaba únicamente con un mes y medio de tiempo, lo que en su momento fue una restricción severa. El enfoque dado al proyecto, basado en la implementación de modelos de simulación en concordancia con la tendencia actual del prototipado rápido¹⁷, permitió hacer frente a esta limitación.

Otra de las restricciones vinculadas al trabajo realizado fue que, por cuestiones físicas de ubicación geográfica de la planta en cuestión, la misma no pudo ser visitada durante el proyecto. Esto impidió trabajar con datos de primera mano sobre el proceso. Sin embargo, no fue una limitación que haya impedido la realización de ningún análisis; en el peor de los casos, únicamente lo retrasó.

¹⁷ Más conocida como **rapid prototyping**, esta tendencia intenta obtener rápidamente prototipos de modificaciones o renovaciones de diseños a partir de la modularización y la elaboración de modelos virtuales. Su principal campo de acción es el diseño asistido por computadora, pero también se aplica a modelos de simulación y análisis de procesos.

Vale mencionar un conjunto de restricciones particulares que fueron simplificadoras para el proyecto. De manera sintética, lo que sucedió fue que todos los parámetros de recursos vinculados a la programación de la producción se asumieron constantes desde el inicio del proyecto, lo que transformó al problema de programación en un problema de scheduling. No significa esto que los recursos de la planta sean constantes, sino que los parámetros de los que dependen se asumieron invariables (por ejemplo, no se consideró la posibilidad de modificar un método para aumentar la productividad de un centro de trabajo), de modo que la solución al problema de scheduling correspondiente podría traducirse fácilmente a la del problema de programación – en concordancia con la terminología expresada en *III.2*.

Entre los **recursos** centrales del proyecto, se puede mencionar, en primer lugar, la posibilidad de acceder a casi todos los datos necesarios y relevantes de la planta sobre la cual se realizó el estudio. El acceso a esta información permitió un mejor análisis del problema en particular y del tema en general. Por otra parte, también se contaba con las herramientas de simulación adecuadas – principalmente el software y el manejo del mismo – para permitir que este proyecto de servicios tecnológicos se llevara a cabo sin inconvenientes respecto al tema.

Finalmente, fue fundamental para la realización del proyecto de servicios tecnológicos mencionado, la formación de un equipo compuesto por las empresas cliente y asesora que permitió un análisis plural y de gran valor aportando profundidad a la presente tesis.

< IV > CASO DE ESTUDIO

IV.1 ♦ UNA SOLUCIÓN INTEGRADORA

La solución que se dio al problema central de esta tesis fue conceptualmente sencilla pero con un enfoque relativamente innovador. Lo que se buscó fue integrar los conocimientos adquiridos en materia de scheduling, programación, algoritmos, heurísticas y simulación en un eje que permitiera resolver correctamente el problema planteado.

El esquema de la solución fue el siguiente: en vez de decidir la metodología de optimización a aplicar únicamente basados en las características del problema, generar una plataforma virtual de evaluación de técnicas que permita simular lo que habría sucedido si su hubiera aplicado una u otra metodología en los meses pasados. Así, en base a datos reales y concretos de órdenes, tiempos de procesamiento y demás particularidades de la planta, se podría evaluar la factibilidad y los indicadores resultantes de la aplicación de las distintas metodologías para definir la nueva técnica.

La solución propuesta cuenta entonces con un grupo de tareas principales vinculadas a las metodologías de optimización (definirlas o seleccionarlas, desarrollarlas, definir los indicadores de evaluación y luego evaluarlas) que se ordenan alrededor del modelo de simulación. Esto se debe a que, por un lado, algunas de las metodologías serán directamente programadas sobre el modelo, y por otro, la información referente a los indicadores definidos deberá obtenerse de las corridas del modelo, con lo cual requerirá de desarrollos en el mismo.

Todas estas tareas puntualizadas en la *Figura 14* se detallarán de aquí en adelante, en lo que resta del presente capítulo y parte del siguiente.

La solución presentada aporta contenidos sustentados

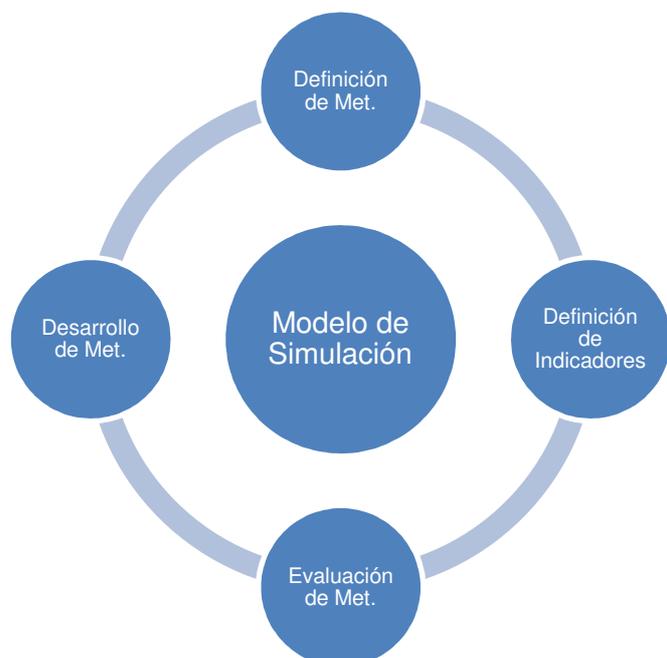


Figura 14 - Esquema de la solución

desde el ámbito académico para su aplicación en este tipo de problemas, como ser heurísticas de búsqueda local y global, algoritmos y optimización por simulación. Basta con revisar algunos de los documentos citados en VII.2 para dimensionar el aval que tiene este tipo de metodologías a nivel internacional. Sin embargo, también se aportan ideas relativamente inexploradas como es el empleo de modelos no sólo para optimizar una metodología sino también para comparar metodologías en una misma planta. Esto permite ganar amplitud y criterio a la hora de definir una técnica adecuada.

En alguna medida, este documento pretende generalizar la línea académica y técnica subyacente en las publicaciones citadas en II.4, que postula que un modelo de simulación puede emplearse no sólo para realizar análisis estocásticos y para evaluar el “funcional más complejo” de una metodología, sino también para probar la efectividad de una técnica concreta, compararla y obtener conclusiones. Las mencionadas demostraciones probablemente no puedan formalizarse matemáticamente, lo que en estos casos no representa un gran inconveniente ya que la aplicación que se busca dar a la herramienta es práctica y no teórica. De hecho, algunas de las heurísticas mencionadas en II.2.c no cuentan con evidencia matemática de su convergencia y/o su nivel de optimalidad y se emplean de todos modos.

IV.2 ♦ DESARROLLO DEL MODELO DE SIMULACIÓN

Previamente a la tarea de generar un modelo de simulación que imite el comportamiento de las órdenes en la planta, se debió realizar un análisis del proceso productivo que se comenta brevemente a continuación.

El proceso de roscado de tubos consta de cuatro operaciones fundamentales:

- [1] El **swage** o “swageado” (a veces traducido como “martelado”), que comprende la uniformización del cilindro formado por la punta del tubo, para luego poder trabajarlo correctamente
- [2] El **torneado**, que corresponde al trabajo realizado mediante tornos que imprimen la rosca a los extremos del tubo
- [3] El **fosfatizado**, que consiste en sumergir la punta de los tubos en una solución química que mejora su resistencia
- [4] Y el **cuplado**, que consiste en colocar cuplas con determinada tensión en una punta de los tubos macho-macho para poder unirlos luego

Estas operaciones se realizan en serie, en ambos extremos de cada tubo (no simultáneamente). Existe una importante cantidad de tipos de rosca

que representan una excepción a este proceso básico, por lo cual alguna o varias de las operaciones mencionadas podrían no ser requeridas para esos tubos.

Una de las características centrales del proceso es que el mismo está centrado en la operación 2. Los tornos son la máquina más cara, el torneado es la operación que más tiempo toma en el ciclo además de ser la que más restricciones impone en cuanto al diámetro de los tubos y el centro de torneado representa generalmente el cuello de botella de la planta. Adicionalmente, el torneado es la única operación que está presente en el orden tecnológico de todas las unidades que ingresan a la planta.

El proceso productivo consiste en aplicar las operaciones necesarias a los tubos de manera secuencial. En general las operaciones se realizan primero a un extremo del tubo y luego al otro, existiendo muy poca distancia entre una máquina y otra. La distribución de la planta es por procesos, lo que significa que el tubo es transportado de un centro de trabajo a otro para llevar a cabo el roscado. Este transporte se realiza por medio de bancales (en esos casos el tubo rueda por gravedad) y gracias al uso de conveyors que permiten trasladar el tubo en dirección axial para cambiarlo de bancal. Además de la posibilidad de almacenar unos pocos tubos sobre los bancales entre un proceso y otro, existen en la planta dos grandes docks de almacenamiento accesibles mediante autoelevadores. Al primero de estos espacios de inventario puede accederse desde el inicio del proceso y desde el transporte intermedio entre la operación 1 y la 2. Al segundo espacio puede accederse desde el transporte entre la operación 2 y la 3 y desde la salida del proceso.

Entre las particularidades más relevantes de esta planta y del problema abordado, se puede comentar lo siguiente:

- ▶ Existen decenas de variables diferenciables para la rosca de un tubo. Sin embargo, a efectos de simplificar el problema y permitir su modelización se eligieron dos cualidades centrales de cada rosca: su diámetro (también llamado “medida”) y su “rosca” (entendiéndose como “rosca” al nombre de la rosca, que representa un conjunto de características fijas de la misma como ser paso, dibujo, etc.). Estos dos atributos son centrales ya que de ellos depende la necesidad de realizar o no setups en las máquinas involucradas en el proceso.
- ▶ El centro de trabajo swage opera con 3 pares de máquinas distintas según el diámetro del tubo. Si las órdenes se alternan de manera que el centro no requiera procesar 2 órdenes seguidas en

las mismas máquinas, se puede realizar el setup de las mismas mientras se trabaja en otra orden, optimizando el tiempo de proceso de dicho centro.

- ▶ La planta cuenta con 4 pares de tornos que pueden operar sobre distintos rangos de diámetro (son pares porque cada uno opera sobre un extremo distinto del tubo). Realmente no hay tornos con rangos coincidentes pero, aproximadamente, 2 tornos operan sobre diámetros pequeños y medianos, y otros dos sobre diámetros medianos y grandes, siendo un único torno el que puede procesar las órdenes del mayor diámetro admisible por la planta.
- ▶ Luego de torneear el tubo es necesario inspeccionar la rosca con un calibre adecuado, distinto para cada diámetro y cada rosca. Se puntualizó que la disponibilidad de estos calibres es a veces una restricción en la planta de roscado debido a que dichos instrumentos de medida circulan por distintas plantas de la misma compañía. Se analizó este impedimento y la estructura necesaria para su simulación pero finalmente no fue incorporado en el modelo.
- ▶ Al cambiar de diámetro o de rosca el torno requiere un considerable tiempo de preparación durante el cual no puede trabajar sobre ninguna orden.
- ▶ Por la disposición física de esta planta en particular, las operaciones 3 y 4 poseen una importante interdependencia que impide que se realicen simultáneamente; de esta manera, se decidió conceptualizar a ambas como un único centro de trabajo que puede realizar tanto el fosfatizado como el cuplado.
- ▶ Todas las máquinas de la planta poseen una productividad calculada mediante estudios de métodos que es afectada, a efectos del cálculo de una programación, por un factor de eficiencia llamado internamente “utilización”, que también depende de la máquina.
- ▶ Las fallas en el swage son muy bajas y pueden solucionarse relativamente rápido transportando el tubo hasta una sierra cercana, cortando el extremo defectuoso y swageándolo nuevamente.
- ▶ La estadística de fallas, scrap y reprocesos se contabiliza por tipo de rosca y diámetro, únicamente para los tornos. Es decir, cada combinación torno-rosca-diámetro posee una tasa de fallos porcentual calculada con datos históricos.

- ▶ Cuando un tubo falla en el torno sigue la operación normal de la planta hasta el final del proceso, procesando únicamente el extremo que no falló; luego se corta con una sierra la rosca defectuosa y se reprocesa ese lado únicamente, repitiendo todo el recorrido por la planta.
- ▶ Las fallas en el fosfatizado y en el cuplado son prácticamente nulas.
- ▶ Las órdenes poseen una cantidad de tubos determinada (generalmente en toneladas) y una tolerancia especificada; de dichos datos puede obtenerse cual es la mínima cantidad de tubos buenos que debe tener la orden al ser terminada.

IV.2.a – Modelo Conceptual

El resultado de la conceptualización del proceso llevada a cabo por el equipo puede consultarse al final de este documento, como *Anexo VIII.6*. A continuación (*Figura 15*) se presenta un esquema simplificado del mismo.

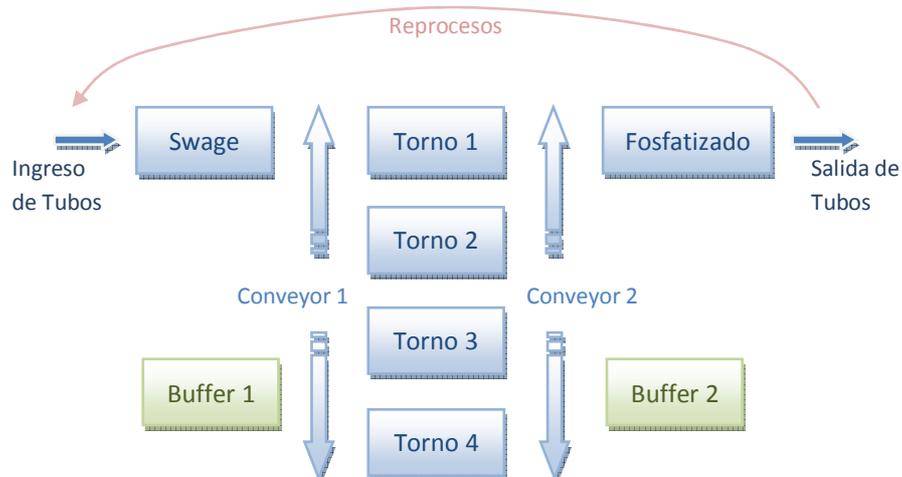


Figura 15 - Esquema simplificado del proceso

Este diagrama centrado en actividades permite visualizar las relaciones entre procesos que tienen lugar en la planta. A su vez, este esquema es el puntapié inicial para el desarrollo de un modelo de simulación que imite el comportamiento de la planta, ocupándose principalmente en los centros de trabajo mencionados e incorporando toda la información técnica de la misma.

Antes de pasar al desarrollo concreto del modelo, es importante hacer la siguiente observación en cuanto a los niveles del problema. Si bien en el

planteo inicial se tiene un problema de programación, el mismo se reduce – debido a las restricciones mencionadas – a un problema de scheduling. De igual forma, dado que el modelo de simulación incorporará todos los tiempos de producción, demoras, precedencias y demás, el input a optimizar se simplifica nuevamente transformándose en un problema de secuenciamiento. Es decir, el modelo permite optimizar el secuenciamiento (en vez del scheduling), y utilizarlo como input para, luego de simular, obtener el scheduling detallado. A su vez, los recursos externos, que son virtualmente constantes, se integran al scheduling para solucionar el problema de programación propuesto (véase *Figura 16*). De esta manera, las técnicas de optimización deberán trabajar fuertemente en el secuenciamiento que es, a priori, un problema más sencillo que el de scheduling.

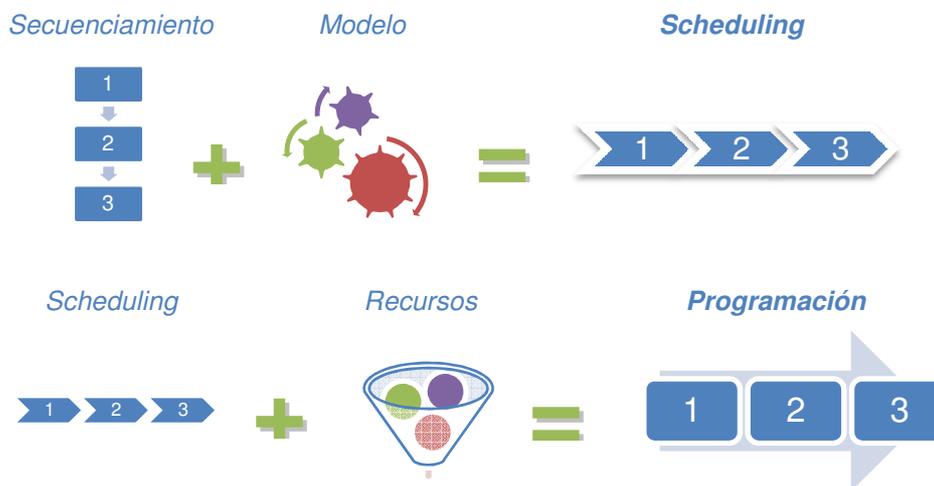


Figura 16 - Niveles del problema

IV.2.b – Codificación del Modelo

La modelización de este tipo de procesos se realiza generalmente mediante simulación discreta. A tal fin, se debió traducir a un programa específico el concepto desarrollado en el apartado anterior, incorporando todas las restricciones y parámetros necesarios para imitar el sistema correctamente. Esta tarea habitualmente se conoce como **codificación**.

El modelo fue codificado en el software de simulación **AnyLogic** (de **XJ Technologies**)¹⁸, que trabaja basado en el lenguaje de programación Java. La principal ventaja que ofrece respecto de otros paquetes es su destacada versatilidad. Por ejemplo, es actualmente el único programa que permite

¹⁸ Para mayor información, remítase a la referencia [24] de *VII.2*.

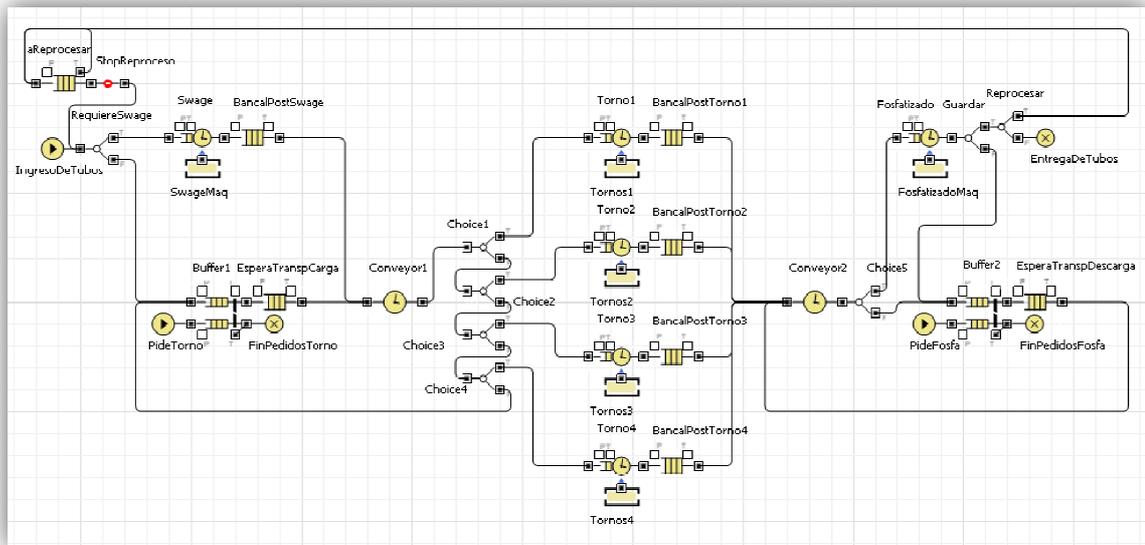


Figura 17 - Captura de pantalla del modelo

combinar y anidar los tres enfoques de simulación¹⁹ en modelos de alta complejidad. En cuanto a la simulación discreta, este software permite una enorme libertad a la hora de programar restricciones particulares, algo interesante para lograr profundidad en el estudio del caso y representar la realidad de la manera más adecuada posible.

Una ventaja adicional que permitió el mencionado software fue la posibilidad de interactuar con archivos de **Excel**, tomando los datos necesarios y devolviendo los resultados obtenidos en el formato de planilla de cálculo más ampliamente usado. De esta manera, se agilizó enormemente la interacción entre los datos reales, las corridas del modelo y la comparación de indicadores.

IV.2.c – Descripción del Modelo

El modelo simula la operación de la planta de roscado durante un mes, considerando los tres procesos principales definidos, los conveyors y los buffers para stock intermedio. En la *Figura 17* se presenta una captura de la pantalla del software para que el lector pueda observar su similitud con el esquema de la *Figura 15*.

Sus límites en cuanto a procesos son los siguientes: se simula desde el almacenamiento en el primer banco (el anterior al **swage**) hasta el proceso de **fosfatizado/cuplado**, quedando fuera del alcance del modelo el almacenamiento de tubos sin roscar y el despacho de tubos terminados.

¹⁹ Véase VIII.4 para una breve reseña sobre los mencionados enfoques.

La modelización de la planta se realiza mediante métodos de simulación discreta, tomando como base la entidad "**tubo**", cuyos atributos se detallan en la *Tabla 1 (página siguiente)*. Las entidades son procesadas en Swage, Torneado y Fosfatizado/Cuplado según requerimiento y se demoran en cada uno de los procesos un tiempo igual al asignado. Además, estos procesos poseen tiempos de setup que se activan al cambiar de diámetro o rosca y dejan la máquina ocupada durante las horas que tome reconfigurarla.

Atributo	Tipo	Significado
Cant	Entero	Cantidad de tubos que pueden transportarse juntos en un autoelevador (similar a un "tamaño de lote")
Conv1	Continuo	Tiempo requerido para ser transportado, en horas
Conv2	Continuo	Tiempo requerido para ser transportado, en horas
Cupla	Binario	Pasó o no por el Cuplado
Destino1	Entero	Lugar a donde se dirige un tubo que pasa por el conveyor 1, siendo 0 el buffer 1 y 1-4 los 4 tornos
Destino2	Entero	Lugar a donde se dirige un tubo que pasa por el conveyor 2, siendo 0 el buffer 2 y 1 el fosfatizado
Diam	Continuo	Diámetro del tubo, en pulgadas
Entrega	Continuo	Fecha de entrega pactada, en horas hábiles contadas a partir del inicio del mes
FallaExtra	Continuo	Tiempo extra requerido en Swage en caso de un fallo, expresado en horas
Fallo	Binario	Falló o no durante el procesamiento actual
Fallos	Entero	Cantidad de veces que el tubo falló en los tornos
Fosfa	Binario	Pasó o no por el Fosfatizado
Orden	Entero	Número de orden a la que pertenece el tubo
Rosca	Entero	Tipo de rosca
SetUpExtra	Continuo	Tiempo extra requerido para setups, en horas
Swage	Binario	Pasó o no por el Swage
TasaFallos	Continuo	Tasa de fallos de la orden a la que pertenece el tubo
TCupla	Continuo	Tiempo requerido para procesar un tubo en el Cuplado, en horas
TFosfa	Continuo	Tiempo requerido para procesar un tubo en el Fosfatizado, en horas
Torno	Binario	Pasó o no por un Torno
TSwage	Continuo	Tiempo requerido para procesar un tubo en el Swage, en horas
TTorno	Continuo	Tiempo requerido para procesar un tubo en el Torno, en horas

Tabla 1 - Atributos de la entidad tubo dentro del modelo

Es oportuno mencionar que algunos de los atributos mostrados se cargan a las entidades al iniciar la simulación (los tiempos de los procesos, número de orden, etc.), mientras que otros son modificados durante la corrida, como ser SetUpExtra, Conv1, Destino2 y Cupla entre otros.

Los tubos se rigen por un sistema "pull" dentro de la planta. Se asume su almacenamiento previo al ingreso (no simulado), lo que da disponibilidad de trabajos para el swage. El swage, de acuerdo a la programación que se le

cargue, solicita tubos del stock para procesarlos y, una vez swageados, los dirige al primer sector de inventario intermedio (buffer 1) haciendo uso del conveyor 1. Los tubos permanecen por defecto en el buffer, hasta que algún torno, de acuerdo a su propia programación, los pida. En ese caso, haciendo nuevamente uso del conveyor, los tubos se dirigen al bancal de ingreso del torno en cuestión, previo chequeo del rango y de disponibilidad de espacio en el bancal. Una vez torneados, los tubos son depositados en el segundo sector de inventario intermedio (buffer 2). De la misma forma, los tubos permanecen allí hasta que el fosfatizado/cuplado, de acuerdo a su programación o a la regla dinámica programada en el modelo, solicite ese trabajo. Mediante la interacción con el segundo conveyor, el tubo es llevado al proceso fosfatizado y luego de realizarle las operaciones necesarias sale del sistema, contabilizándolo como un tubo producido.

Los fallos/scrap se manejan por tasas históricas con implicaciones probabilísticas dentro del modelo. En cuanto al swage, se le asignó una probabilidad de fallas en base a sus valores históricos. Cuando una falla ocurre, se requiere un tiempo extra para reprocesar el tubo en ese centro. En cuanto al torneado, la probabilidad de falla depende de la orden en cuestión. Cuando una falla ocurre en los tornos, el tubo sigue su secuencia de trabajos normalmente pero hacia el final del proceso es almacenado en un stock de reproceso. Esta pieza reingresa al sistema en el caso que el swage este trabajando sobre esa misma orden; en caso contrario, queda separado para reprocesarse el mes siguiente. En el fosfatizado no se consideran fallas.

Para completar la descripción del modelo, es necesario especificar sus **inputs** y sus **outputs**, lo que permitirá comprender mejor su funcionamiento interno. El modelo recibe información tomándola de archivos Excel debidamente identificados o ingresándola en la pantalla de inicio del mismo. Dicha información se compone de:

- ▶ **Listado de órdenes** a procesar, donde figura, para cada orden, su número de orden, cantidad, cantidad mínima, diámetro, rosca, tasa de fallos en el torno, fecha de entrega, tiempos de procesamiento en todos los centros, peso y volumen por tubo.
- ▶ **Secuenciamiento** de todas las máquinas del proceso: swage, tornos 1 a 4 y fosfatizado/cuplado. Si se emplea una regla dinámica – explicada en el apartado siguiente – puede prescindirse del último secuenciamiento.
- ▶ **Datos propios de las máquinas:** productividad neta del swage, fosfatizado y cuplado, productividad bruta de los tornos y

utilización de los tornos. Entiéndase por “neta” luego de multiplicar por la utilización correspondiente.

- ▶ **Otros datos de planta**, como ser: velocidad de conveyors, tasa de fallas del swage, calendario del mes a simular, longitud de bancales, disponibilidad de calibres y tiempos de procesos auxiliares (básicamente la sierra).

Mediante la simulación discreta, el modelo intenta reproducir lo que sucedería en la planta en base a los datos proporcionados, devolviendo luego del procesamiento los siguientes outputs:

- ▶ **Indicadores** de performance de la planta y el secuenciamiento, definidos en *IV.5*.
- ▶ **Diagrama de Gantt** que representa la operación de la planta hora por hora, mostrando los trabajos, setups y tiempos ociosos de todas las máquinas durante el mes simulado.

Este intercambio de información entre el modelo y el usuario puede esquematizarse como se muestra en la *Figura 18*.

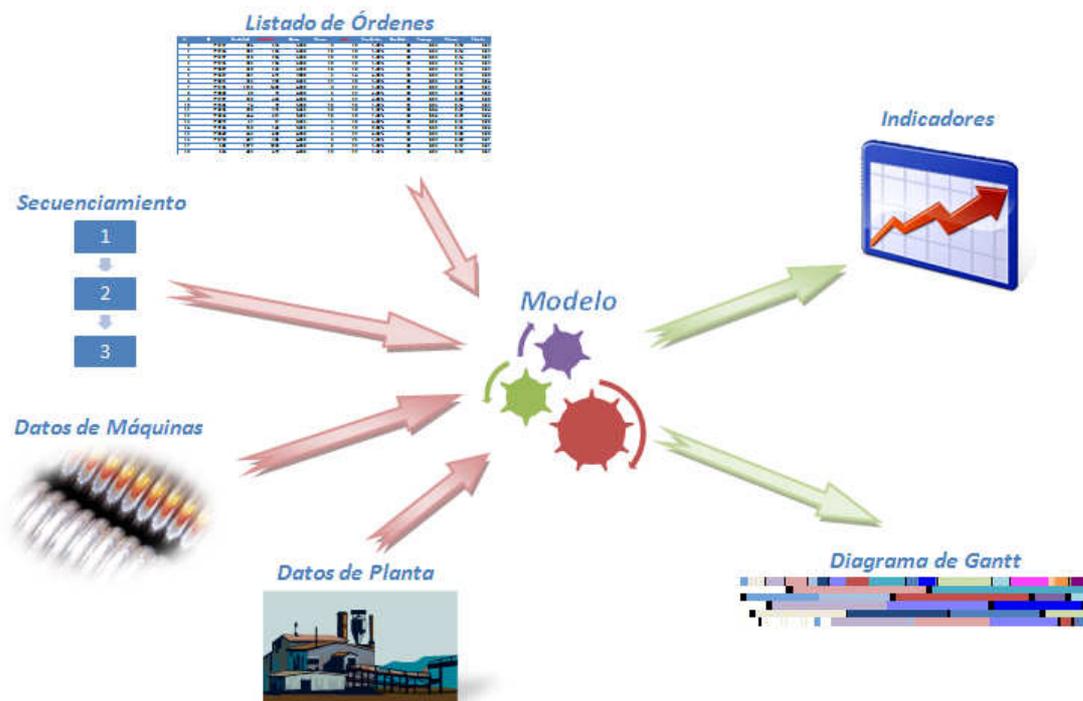


Figura 18 - Inputs y outputs del modelo

Es importante aclarar que, en el caso de emplear el modelo mismo como optimizador (*véase IV.3.d*) la información referente al secuenciamiento de las máquinas no es un input sino que es el resultado de la aplicación de la

heurística de búsqueda programada. En esos casos, dicha información es generada por el modelo y puede obtenerse el secuenciamiento como output.

IV.2.d – Esquemas Lógicos de Trabajo del Modelo

Debido a la mencionada característica de este tipo de plantas que tiende a centrar el problema de scheduling en la programación de los tornos, surgió la posibilidad de trabajar con diversas metodologías de optimización sobre el secuenciamiento de tareas sólo sobre los tornos y aplicar reglas estáticas o dinámicas convenientes a los demás centros de trabajo. A tal fin, se diseñaron esquemas lógicos de trabajo tanto para el swage como para el fosfatizado, ambos asumiendo que ya se ha definido el secuenciamiento de los tornos. El primero consiste en una regla lineal y estática, previa a la corrida del modelo, mientras el segundo es una regla dinámica que se aplica durante la corrida para la toma de trabajos del centro. Vale remarcar que estas heurísticas de construcción pueden utilizarse o no para complementar cualquier técnica y que arrojaron muy buenos resultados en todos los casos.

La **lógica básica de secuenciamiento del swage** pretende hacer explícitas las ideas surgidas en la programación manual del centro. Se aplica antes de correr el modelo, tomando como input la programación de todos los tornos y los datos básicos de las órdenes a analizar, y entregando como output la secuencia de trabajos para el swage.

Los datos que emplea esta lógica son:

- ▶ De cada orden:
 - Cantidad de tubos
 - Productividad del Swage (neta)
 - Productividad de los tornos (bruta)
 - Torno que rosca esa orden
 - Posición de esa orden en el secuenciamiento del torno (es decir, si es la primera del torno en el cual se hace, la segunda, la enésima, la última, etc.)
- ▶ De cada torno:
 - Utilización

Los principales supuestos que aplica esta lógica son:

- ◆ No considera setups ni de tornos ni de swage.
- ◆ Las tareas se hacen, en principio, en un sólo torno. En caso de partirlas, se deberían colocar en el torno que primero las va a requerir para que funcione correctamente.

- ◆ Como la productividad del swage es siempre mayor que la de los tornos, es suficiente que el swage empiece a trabajar en un orden al mismo tiempo que el torno para que este último trabaje de forma continua. Es decir, el momento de mayor importancia es el inicio de los trabajos, tanto en los tornos como en el swage.

El procedimiento que aplica es lineal y puede esquematizarse de la siguiente manera:

- [1] Suponiendo que todos los tornos comenzaran a trabajar en el momento 0, calcular cuándo debería procesarse cada orden para que el torno pueda trabajar en continuo todo el mes. Llamar a ese momento "requerimiento" de la orden.
- [2] Afectar los requerimientos sumándoles una penalidad que multiplica el tiempo que tarda la orden completa en procesarse en el swage. De esta forma, si dos órdenes tienen el mismo requerimiento (típicamente el momento 0), se requerirá la más corta antes y la más larga después (por ejemplo, la primera tendrá requerimiento 0,12 y la segunda 0,43).
- [3] Ordenar los trabajos en el swage de menor a mayor requerimiento.
- [4] Calcular, con esa secuencia, el momento en el que efectivamente se van a realizar los trabajos en el swage, por suma de tiempos de tareas precedentes.
- [5] Calcular nuevamente los requerimientos, pero esta vez iniciando la operación de cada torno en el momento que el swage hace la primera tarea de ese torno (no en el momento 0, como en 1).
- [6] Sumar los retrasos que provoca el ordenamiento del swage, haciendo el siguiente cálculo: si se trata de la primera orden del torno, el retraso es el momento en el que el swage le permite iniciar la operación al torno; en otro caso, el retraso es el momento en el que el swage entrega la orden menos el momento en el que el torno la necesitaría para trabajar en continuo (siempre y cuando la resta sea positiva, sino el retraso es nulo).²⁰

²⁰ Si se aplica rigurosamente el último paso, se puede notar que la lógica penaliza los atrasos de forma diferencial cuando la planificación es muy ajustada. En esos casos, los atrasos más penalizados son los primeros que se encuentran entre tareas. Por ejemplo, si el torno tuviera 4 tareas (A, B, C y D) planificadas en ese orden de manera "just in time" con el swage, el retraso más penalizado sería el provocado entre la A y B, luego entre B y C, luego entre C y D y finalmente el previo a A. En la gran mayoría de los casos, la planificación no es así de ajustada en todas las tareas del torno, con lo cual se penaliza por igual a todos los retrasos.

Penalidad como variable de control: si bien el secuenciamiento del swage ya está definido en el paso 3, se continúa la lógica para poder luego manejar la penalidad y obtener un mejor secuenciamiento.

Si la penalidad es muy baja (entiéndase menor a 0,001), la lógica ordena por requerimiento de tornos con la única salvedad que a igualdad de requerimientos realiza la tarea más corta primero. Si la penalidad es muy alta (entiéndase mayor a 1000), la lógica ordena las tareas de menor a mayor duración de swage, sin importar los requerimientos de tornos. En todos los valores medios, el secuenciamiento considera una combinación de ambos objetivos.

Teniendo en cuenta esta sensibilidad, se puede fijar la penalidad manualmente (los valores entre 1 y 3 otorgan en general los mejores resultados prácticos) o emplear la función "Buscar Objetivo" del Microsoft Excel para que fije la penalidad que minimice los retrasos totales del secuenciamiento. El output de la lógica es simplemente un ordenamiento de tareas en el swage por número.

La **lógica del modelo para el centro de fosfatizado/cuplado** (que se llamará simplemente **fosfatizado**, pero incluye ambas operaciones) es una regla dinámica que toma en cuenta algunas variables de operación del modelo para decidir en qué orden debe trabajar el centro. Se aplica continuamente durante la simulación de la planta, ejecutándose cada 0,3 horas del modelo.

La lógica completa se esquematiza en el diagrama de flujo que figura en el *Anexo VIII.6* tal como está programada en el modelo, pero se resume en los siguientes párrafos.

El centro fosfatizado busca como primera opción seguir trabajando en la orden que se encuentra fosfatizando al momento de hacer la evaluación. Si existen tubos de esta orden en el buffer (entiéndase buffer como el depósito intermedio entre torneado y fosfatizado durante este apartado), se seguirá trabajando en ella cuando se cumpla al menos una de las siguientes condiciones:

- ▶ Los tubos que hay en el buffer aseguran 0,3 horas de trabajo o más
- ▶ Los tubos que hay en el buffer son los últimos que deben fosfatizarse para completar la orden

En caso de no cumplirse ninguna de las condiciones, se evalúa, de entre todas las órdenes que cuentan con tubos en el buffer, en qué casos la orden se

completa al fosfatizar todo el material del buffer. Si alguna orden cumple con dicha condición, se procede a trabajar sobre ella.

En caso de no hallar una orden con tales cualidades, se preseleccionan todas las órdenes que cuentan con más de "n" tubos en el buffer, siendo "n" un lote mínimo definido externamente (el valor asignado actualmente es de 270). De dicha preselección, se elige luego la orden que asegura mayor tiempo de trabajo continuo en el fosfatizado, priorizando aquellas que no requieren setup por poseer tubos del mismo diámetro que la orden actual.

Si la preselección resulta nula, no se hace ningún pedido al buffer y el fosfatizado sigue trabajando si hay tubos en su bancal hasta la próxima ejecución de la lógica.

En todos los casos, antes de pedir tubos, se comprueba la disponibilidad de espacio en el bancal. En caso de encontrarse lleno, no se hace ningún pedido y se espera hasta la próxima ejecución de la lógica.

Las consecuencias de esta lógica tienen cierta sensibilidad al tamaño del lote mínimo. Si bien los resultados son bastante similares para lotes de entre 100 y 400 tubos, se determinó que un lote de 270 tubos otorgaba una lógica razonable y unos resultados aceptables en los meses analizados. De todas formas, puede determinarse el lote mínimo "óptimo" corriendo una optimización de 10 minutos para cada mes, previa a la simulación de dicho mes. Esta tarea es sencilla y está programada en el modelo.

IV.2.e – Supuestos y Restricciones del Modelo

Todos los datos ingresados al modelo son determinísticos, es decir que no existe en ellos variabilidad entre corridas. La única excepción es el porcentaje de fallas, que si bien es un dato concreto e invariable, produce efectos probabilísticos que pueden hacer diferir levemente entre sí a dos corridas con idénticas condiciones iniciales.

Varias de las restricciones que se analizaron durante el proyecto no fueron modeladas en pos de la simplicidad operativa del modelo; de esta forma, pasaron a ser, en algún sentido, supuestos del modelo. Sin embargo, son supuestos flexibles, ya que con leves modificaciones podrían derribarse; el modelo se encuentra preparado para aceptarlos, pero no están programados. Estos supuestos se encuentran especificados en la página siguiente:

- ◆ Se asume que los calibres para el proceso de inspección luego del roscado están siempre disponibles, sin importar el diámetro y el tipo de rosca en cuestión.
- ◆ Se asume que la materia prima para todos los pedidos está en el depósito al iniciar la simulación, por lo cual no constituye una restricción en ningún caso. De igual manera, se aplica el mismo supuesto a los protectores y a las cuplas.
- ◆ La simulación se inicia con la planta "vacía". No hay tubos en los banales ni en los buffers. Ninguna orden está en proceso en el momento 0.

Entre las restricciones más importantes de la planta que fueron incorporadas al modelo, se encuentran las siguientes:

- ▶ **Asignación de tornos a pedidos** (no todos los tornos pueden roscar todos los diámetros): esta restricción ya debería estar implícita en la programación, que es el input del modelo. De todas formas, el modelo verifica esta restricción e impide asignar trabajos que los tornos no puedan procesar.
- ▶ **Turnos de trabajo:** el horario de trabajo se tomó de la programación de la planta.
- ▶ **Tiempos de setup:** se simularon los tiempos proporcionados por la planta, dando la posibilidad a una estructura matricial de setups para los tornos que en principio se dividió en 3 zonas: 0 horas en caso de ser tubos del mismo diámetro y rosca, 2 horas en caso de igual diámetro pero distinta rosca, 4 horas en caso de cambio de medida.
- ▶ **Setups del swage:** en este caso particular, los tiempos de setup dependen del uso de las distintas máquinas disponibles y la posibilidad de hacer setups "a la sombra" de otro trabajo. Esta restricción fue analizada en base a los rangos de todas las máquinas de swage y fue modelizada.
- ▶ **Bancales:** la capacidad de los banales fue modelada de acuerdo a las medidas de la planta estimadas y al diámetro de los tubos.

IV.2.f – Validación del Modelo

La validación del modelo se realizó de forma paralela a la construcción del mismo. Principalmente fue una validación cualitativa consensuada con el equipo de proyecto de la empresa asesorada. Las diversas etapas de validación que acompañaron la creación del modelo fueron las siguientes:

Diagrama IDEF0: El diagrama mostrado en el *Anexo VIII.6* y comentado anteriormente se realizó de manera conjunta entre personal de la empresa cliente y de la empresa asesora. De esta manera, los procesos mostrados y analizados en el mismo se encuentran totalmente validados con la realidad. Por otra parte, el modelo se generó teniendo como base dicho diagrama, de forma que su estructura se encuentra validada por el equipo. Mediante una sencilla comparación visual puede comprobarse que los procesos que se encuentran simulados por el modelo son los mismos que se conceptualizaron en el diagrama, así como los recursos y los controles especificados.

Fuentes de Datos: Todos los datos numéricos empleados en el modelo provienen de la empresa asesorada, tanto de la planta como del análisis del equipo de trabajo.

Metodología Actual: Para el Mes 1 (y sólo para ese mes) se contó con información adicional de planta que especificaba la secuencia en la que se procesaron las órdenes en los distintos turnos. Si bien esta información es incompleta y no permite una validación total del modelo, al ser los turnos el determinante principal de los tiempos de producción pudo recrearse parcialmente la situación real de dicho mes. De esta manera, se intentó validar cuantitativamente el cumplimiento, las fechas de entrega de órdenes, y otros parámetros de planta. Dado que los datos concretos de planta no estaban disponibles, se consensó con el equipo de trabajo que la situación que logró modelarse era razonablemente similar a lo que debería haber ocurrido, validando indirectamente las lógicas internas del modelo.

Diagrama de Gantt: para todas las corridas del modelo se lograron graficar los diagramas de Gantt correspondientes a los distintos centros simulados; de esta manera, se logró interactuar con el equipo y validar indirectamente el modelo mediante la aceptación de los resultados de las distintas metodologías y la comparación con los esperados “haciendo los cálculos por fuera del modelo”. Esta herramienta de comunicación fue crucial durante el proyecto y representó el instrumento de validación indirecta más importante del trabajo. La razón de su relevancia residió en que el diagrama de Gantt fue una suerte de traductor entre el idioma del modelo y el idioma del equipo de trabajo, principalmente de los programadores de la empresa que participaron.

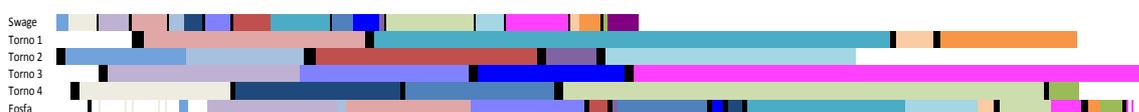


Figura 19 - Ejemplo de un diagrama de Gantt extraído del modelo

IV.3 ♦ DESARROLLO DE METODOLOGÍAS DE OPTIMIZACIÓN

A partir del panorama presentado en el capítulo //, se debieron seleccionar unas pocas metodologías para desarrollar y testear como parte de la solución. A tal fin, se buscó que las técnicas a comparar fueran distintas entre sí; es decir, se evitó por ejemplo usar dos heurísticas de búsqueda local o dos tipos de reglas dinámicas. De esta manera, se logró, en la medida de lo posible, una buena representación del abanico de posibilidades abierto por el problema.

Además de la característica mencionada, también se apuntó a que las técnicas seleccionadas fueran de probada eficiencia en el tema, avaladas por la bibliografía específica. A su vez, se intentó trabajar con metodologías que pudieran ser desarrolladas y dominadas por el equipo de proyecto.

Considerando los plazos del proyecto y el tiempo necesario para la implementación de cada una de las técnicas, se seleccionaron básicamente tres técnicas, las cuales se desarrollan en los cuatro apartados siguientes junto con la metodología actual.

IV.3.a – Metodología Actual

La metodología actual se incluye aquí debido a que requirió cierto nivel de desarrollo, aunque muy inferior al de las técnicas de optimización comentadas luego. Para poder simularla se contó con un único dato concreto: el secuenciamiento de los cuatro tornos durante el Mes 1. Además, mediante comunicaciones con la planta, se investigó la forma de trabajo de todos los centros y se delinearon posibles secuenciamientos para el resto de las operaciones. De esta forma, se diseñó un scheduling que, ingresado al modelo de simulación, permitió validar indirectamente algunos de las principales características del mismo.

La metodología actual contribuyó además proporcionando el “piso” o el nivel base de los indicadores para el Mes 1, y de alguna forma para el resto del proyecto. Si bien no se expresó concretamente en el objetivo, cuando se habla de definir una metodología se busca que sea, por un lado, la que proporcione los mejores resultados globales de entre todas las técnicas evaluadas (no sólo considerando los indicadores de planta, sino también tomando en cuenta el tiempo de corrida, la facilidad de implementación, etc.) y, por otro, que los resultados que arroje superen a los de la metodología actual, ya que sino no tendría sentido su implementación. A tal fin, las primeras corridas de validación fueron de alguna manera las que marcaron el rumbo en cuanto al objetivo cuantitativo a alcanzar.

Concretamente, lo que se hizo para simular esta metodología fue:

- [1] Ordenar las tareas en los tornos según el secuenciamiento real proveniente de la planta.
- [2] Ordenar las tareas en el swage en el orden en el que los tornos las requerían.
- [3] Ordenar las tareas en el fosfatizado según el orden de finalización de las tareas en los tornos (usando una regla básicamente FCFS, con algunas modificaciones manuales asumiendo que, por ejemplo, las órdenes más urgentes se fosfatizan primero).

Los resultados fueron muy buenos. Se logró reproducir razonablemente bien la situación de dicho mes, se validó el modelo y se establecieron los objetivos a superar por el resto de las metodologías. Además, este desarrollo, por ser el primero, sirvió para perfeccionar el manejo de inputs y outputs por parte del modelo.

IV.3.b – Metodología de Programación Lineal + Reglas de Despacho

La primera técnica seleccionada, considerando su amplio respaldo en la resolución del problema de programación de la producción, fue la de programación lineal.

En primer lugar, se intentó aplicarla de la manera más precisa (en referencia a las dos formas básicas comentadas en *II.3*). Se generó un modelo que considerara las principales restricciones y que pudiera devolver el scheduling de todos los trabajos a través del tiempo. Analizando la cantidad de variables requerida²¹ por dicho modelo y su dependencia en cuanto al número de órdenes y el número de tornos, se llegó a la conclusión de que este planteo podría llegar a resolver el problema de esta planta en particular pero no podría generalizarse. La razón fue básicamente que, para problemas levemente mayores, la capacidad de cálculo del software de programación lineal adquirido por la empresa se vería colmada.

Se desarrolló entonces un modelo de programación lineal orientado a “optimizar el qué”²². Este modelo de planificación en programación lineal incluyó las tres etapas del proceso (swage, torneado y fosfatizado/cuplado)

²¹ Recuérdese el hecho de que la cantidad de variables total de un problema de programación lineal está dada por la suma de las variables del modelo más las restricciones del mismo, debido a que cada limitación genera una “variable de slack” sobre la que trabaja el algoritmo.

²² Se agradece la colaboración del Ing. Pablo Ciapanna, perteneciente al CEOS (ITBA), en el desarrollo del modelo de programación lineal.

aunque su output central fue la asignación de los distintos órdenes en los cuatro turnos disponibles. La inclusión de los demás procesos permitió que el sistema detectara casos en los que el cuello de botella principal de la planta no era el torneado, para poder distribuir los órdenes correctamente aún en esas ocasiones.

El objetivo del modelo fue minimizar los tiempos de proceso totales de la planta, para todas las etapas (**e** de **1** a **s**), todas las órdenes (**i** de **1** a **m**) y todos las diferentes máquinas de una misma etapa (**j** de **1** a **n_e+1**) – en este caso sólo aplicable a los 4 turnos. Con tales premisas, el funcional fue el que muestra la *Ecuación 1*:

$$\text{Min} \sum_{e=1}^s \sum_{i=1}^m \sum_{j=1}^{n_e+1} A_{i,j,e} \cdot X_{i,j,e} \quad [1]$$

Las variables **A** representan la inversa de la productividad para la orden **i**, la máquina **j** y la etapa **e**. Las variables **X** – variables de control del modelo – representan la cantidad a procesar de la orden **i** en la máquina **j** de la etapa **e**. Lógicamente gran parte de estos valores serán nulos y la suma tendrá muchos menos términos positivos que **s.m.(n_e+1)**. Nótese que la cantidad de máquinas por etapa varía según etapa siendo 1 para **e=1** y **e=3**, y 4 para **e=2**.

Un hecho fundamental del modelo consiste en que, sin importar los datos de entrada, todas las órdenes son producidas en su totalidad en alguno de los centros. Por tal motivo, no tiene sentido programar una función objetivo vinculada al cumplimiento (siempre resultaría ser 100%) ni a los días de atraso (recuérdese que el modelo no simula tiempos). Cuando la cantidad de toneladas a procesar supera la capacidad de los centros, el modelo envía el material restante a un centro de “overbooking” adicionado intencionalmente a tal fin – esa es la razón del **+1** en la cantidad de máquinas por etapa – y lo produce allí. A fines prácticos, dicho centro tiene una productividad muy inferior al resto para evitar que “compita” con las máquinas reales; es decir, su **A** correspondiente es mucho mayor. De esta forma, las órdenes que no pueden realizarse dentro del mes también son un output de este modelo. Sin embargo, en el caso de que el swage necesite mandar órdenes a overbooking, el programa considera que han sido procesadas en dicha etapa y las distribuye dentro de los turnos sin afectar el output principal del algoritmo. Se asume que el problema de overbooking puede resolverse con horas extra u otros recursos operativos.

Habiendo explicado las características básicas del modelo de programación lineal, resta comentar brevemente sus restricciones. La *Ecuación*

2 muestra la restricción de capacidad para todos los centros y etapas; simplemente no pueden usarse más horas de una máquina que las horas disponibles de la misma (**TD**). En este caso **TD** fue igual para todas las máquinas, en concordancia con los calendarios laborales de los meses analizados.

$$\sum_{i=1}^m A_{i,j,e} \cdot X_{i,j,e} \leq TD_{j,e} \quad \forall j,e \quad [2]$$

Finalmente, resta representar las restricciones de cumplimiento de órdenes, que obligan a que todo el material de cada una de las órdenes sea procesado en cada etapa, sin importar el centro del que se trate.

$$\sum_{j=1}^{n+1} X_{i,j,1} = Q_i \quad \forall i \quad [3]$$

$$\sum_{j=1}^{n+1} X_{i,j,e} = \sum_{j=1}^n X_{i,j,e-1} + QP_{i,e-1} \quad \forall i; e > 1 \quad [4]$$

La *Ecuación 3* es de fácil comprensión, simplemente iguala todas las cantidades procesadas de una orden a su cantidad total **Q** fijada como parámetro del modelo. La *Ecuación 4* introduce el concepto de **QP**, que representa una cantidad pendiente del mes anterior que debe ser procesada de la etapa **e** en adelante, es decir, que no pasa por todas las etapas del proceso. Por tal razón, al igualar las cantidades procesadas en cada etapa con su correspondiente etapa anterior, se suma esta posibilidad del “arrastre” de cantidades de meses pasados.

La salida del modelo de programación lineal desarrollado proporcionó una alocaión óptima de órdenes en los turnos, considerando rangos y productividades. Sin embargo, su secuenciamiento dentro de cada turno se realizó mediante una heurística de construcción, haciendo de esta una metodología combinada. El flujograma de la regla empleada se omite por razones de confidencialidad de la información, pero bastará decir que, partiendo de la lista de órdenes a procesar en cada turno, se secuenciaron priorizando el cumplimiento de las fechas de entrega primero y la minimización de los setups luego, mediante una interesante combinación de ambos objetivos distinta a los clásicos factores ponderados. El resultado de la aplicación de esta regla a los cuatro turnos permitió obtener el secuenciamiento completo del torneado, para luego aplicar los procedimientos ya descritos en *IV.2.d* para el swage y el fosfatizado respectivamente, completando el desarrollo de la metodología.

IV.3.c – Metodología de Algoritmos Genéticos

En pos de la pluralidad de las técnicas a evaluar, se escogió desarrollar una heurística de búsqueda global cuyas aplicaciones en el campo hayan sido de probado éxito en los últimos años: algoritmos genéticos. La implementación de esta metodología fue la más compleja, debido a la necesidad de generar un programa específico para la aplicación del algoritmo al problema.

Para lograr menores tiempos de desarrollo y de corrida, durante la elaboración del programa se debieron realizar varias simplificaciones al problema original. Sin embargo, vale mencionar que, si bien estas reducciones se aplicaron a la lógica que definió el secuenciamiento de la metodología, al correr dicho ordenamiento en el modelo se equipararon las condiciones para todas las metodologías. De esta forma, la evaluación de las técnicas se hizo sobre el mismo problema; la simplificación fue, de alguna manera, parte del planteo del problema por medio de algoritmos genéticos.

Las modificaciones más relevantes al problema original fueron:

- ◆ Se obviaron los procesos de swage y fosfatizado, centrando el problema en la asignación de tareas a los distintos tornos y el secuenciamiento de los mismos.
- ◆ Se asumió que la utilización de todos los tornos era la misma.
- ◆ Se hicieron coincidir los rangos de los tornos 2 a 2, dejando dos grupos de tornos: uno para roscas medianas y grandes y otro para roscas medianas y chicas.
- ◆ Se simplificaron los tiempos de procesamiento en los tornos, haciéndolos depender únicamente del tamaño de la rosca y no de cada orden en particular (a tal fin, se obtuvo un promedio ponderado de los tiempos de todas las órdenes que pertenecían a cada medida).

Teniendo en cuenta la terminología de este tipo de algoritmos y las características del problema, se consideró que los individuos o cromosomas se compondrían de conjunto de listas conteniendo el secuenciamiento de los cuatro tornos. Es decir, un cromosoma contendría cuatro listas de números enteros, representando las órdenes a ser procesadas en cada torno en la secuencia indicada. Justamente esas secuencias serían la característica “única” o diferenciadora del cromosoma, ya que todas las tareas deben realizarse en algún torno. Las generaciones o iteraciones serían entonces las repeticiones del algoritmo programado, contando con unos 25 individuos cada una. La descendencia tendría lugar cruzando y mutando a dichos individuos, operaciones que se materializarían concretamente mediante la permutación de

tareas entre dos listas del cromosoma o el cambio de lista de una o más órdenes. Este fue, en resumidas cuentas, el planteo del problema.

Para la implementación de la técnica se debió diseñar un programa con tres módulos relativamente independientes pero que trabajan en conjunto²³:

- [1] Un generador de población inicial aleatorio que respeta los condicionamientos del problema y, basado en heurísticas específicas, genera los 25 individuos que conforman la primera generación.
- [2] El algoritmo genético en sí, que cruza y muta a los individuos respetando las limitaciones del problema.
- [3] Un método de ranking de individuos orientado a la persecución de dos objetivos: minimizar las paradas de máquina en los tornos completando el conjunto de órdenes indicado en el menor tiempo posible, entendiéndose por tiempo la cantidad de horas que transcurren entre el inicio de la primera orden y el final de la última.

En función de lo comentado en *II.2.c*, estos módulos se aplican del modo ya explicitado. Se genera una población inicial, se cruza y muta, se obtiene el ranking para definir que individuos pasarán a la generación siguiente y se repite el proceso hasta que se cumpla cierta condición.

La condición de término del algoritmo trabaja sobre la estabilidad del funcional evaluado. Si durante varias generaciones se encuentra que los primeros puestos del ranking corresponden a los mismos individuos, se detiene el algoritmo asumiendo que ya ha hallado la mejor solución que puede otorgar (o que está atascado). De todas maneras, se especifica un tiempo máximo de corrida de alrededor de 10 minutos que no puede superarse, de modo que en caso de llegar a tal condición el procedimiento se detiene y otorga la solución perteneciente al primer puesto del último ranking realizado.

Para completar la metodología y evaluarla sobre el modelo de simulación, se debieron definir las operaciones tanto del swage como del fosfatizado, que habían sido excluidas en la definición del problema para esta técnica. A tal fin, se emplearon las lógicas descritas en *IV.2.d*, con las cuales se obtuvieron muy buenos resultados. También se ensayó, sin demasiado éxito, la aplicación de una heurística de búsqueda local para definir el secuenciamiento del fosfatizado.

²³ Se agradece la colaboración del Ing. Enrique Fernández, perteneciente al CAPIS (ITBA), en el desarrollo del programa que implementa el algoritmo genético.

IV.3.d – Metodología de Búsqueda Tabú

La inclusión de la metodología de búsqueda tabú permitió completar el panorama de heurísticas de resolución, empleando un procedimiento avanzado pero local (en contraste con el algoritmo genético). Para el desarrollo de esta técnica se contó con el apoyo del programa de simulación empleado, que cuenta con un paquete de optimización llamado **OptQuest**²⁴. Dicho paquete trabaja atacando el problema mediante distintas heurísticas locales dominadas por una búsqueda tabú que opera sobre las mismas. De esta forma, la programación requerida para aplicar esta técnica fue muy poca; el esfuerzo más grande de desarrollo se centró en la definición de los parámetros a optimizar.

La posibilidad de optimizar directamente sobre el modelo de simulación permitió que esta metodología incorporara todas las restricciones representadas en el mismo de manera automática. De esta forma, la técnica de búsqueda tabú contó con la definición más precisa del problema, obteniendo así una primera ventaja frente al resto de las metodologías seleccionadas. Sin embargo, la optimización misma también implica ciertas restricciones similares a las comentadas en apartados anteriores; la limitación principal es que el funcional debe ser único y de cálculo relativamente sencillo. Por otra parte, el trabajar sobre el modelo también trajo aparejada la desventaja de mayores tiempos de corrida, que deberían ser cuidados y trabajados para hacer a esta técnica competitiva frente a las otras dos.

Existen dos elementos centrales a definir para aplicar la metodología de búsqueda tabú a un modelo de simulación. Por un lado, el funcional a optimizar, que debe expresarse de forma sencilla en función de los parámetros y variables del modelo; por otra parte, los parámetros sobre los cuales trabajará el motor de búsqueda para optimizar.

La definición de un funcional apropiado fue motivo de debate, investigación, pruebas y errores por parte del equipo de proyecto. En concordancia con los principales objetivos de una buena programación de la producción, se ensayaron funcionales tales como: máxima utilización, mayor cumplimiento combinado con menor utilización, menor tiempo de procesamiento, mayor cantidad de tubos producidos, entre otros. Luego de un análisis tanto de los fundamentos teóricos como de los resultados experimentales obtenidos, se definió que el mejor objetivo a programar era el cumplimiento (planteando un problema de maximización), por ser un indicador

²⁴ Para mayor información véase la referencia [27] en VII.3.

clave a la hora de evaluar una técnica y por representar, de alguna forma, uno de los principales objetivos de un scheduling adecuado.

La forma de trabajar sobre los parámetros del modelo por parte del optimizador también fue motivo de un proceso de selección. Se debió armonizar la libertad del programa para modificar las metodologías con la necesidad de acotar el problema de modo que una búsqueda local tuviera resultados exitosos. El procedimiento resultante fue el siguiente: primero se realiza un ordenamiento externo de las tareas mediante la regla SPT, luego se emplea ese orden como semilla para el swage y se subdividen las tareas ordenadas en los distintos turnos, y finalmente se permite que el OptQuest cambie las tareas entre los distintos turnos. El fosfatizado es manejado por la regla dinámica explicada anteriormente.

Nótese que, de los dos objetivos mencionados, se terminó priorizando el acotamiento del problema, debido a que con procedimientos más amplios la metodología tardaba en converger o se atoraba en máximos locales no satisfactorios. De esta forma, si bien la técnica se basa en una búsqueda tabú, se complementa con reglas heurísticas desarrolladas a medida, como en los otros dos casos.

IV.4 ♦ DESARROLLO DE LA INTERFASE

En este apartado se propone comentar brevemente la manera en la que se articuló la interfase entre las metodologías y el modelo de simulación. De alguna manera, se busca explicar la conexión entre el centro y los extremos de la *Figura 14*.

Para las primeras dos técnicas, el esquema fue sencillo: las metodologías otorgaron un ordenamiento que fue traducido a una tabla que el modelo tomó como input desde un archivo de Excel. El aspecto de esa tabla se ejemplifica con la *Tabla 2*.

Oswage	8	4	14	15	9	11	18	1	7	3	17	12	13	5	10	16	6	0	2
Otorno1	4	11	12	6	0	2	3	5											
Otorno2	14	12	6	1	3	13	5												
Otorno3	8	9	7	10															
Otorno4	15	18	9	17	16														
Ofosfa																			

Tabla 2 - Ejemplo de tabla de secuenciamiento

Mediante tablas como esta se interactuó con el modelo durante todo el proyecto. Lo que indica es, sencillamente, en qué orden debe procesar los trabajos cada centro (los números se refieren al número de orden y la posición, de izquierda a derecha, a la secuencia). Nótese que el fosfatizado se encuentra

vacío, lo que implica que en este caso se utilizó la regla dinámica programada en el modelo. También téngase en cuenta que el modelo admite el trabajo paralelo en tornos sobre la misma orden (por ejemplo, las órdenes 3 y 5 aparecen en los tornos 1 y 2); sin embargo, en la mayoría de los casos, las metodologías se restringieron para evitar esa particularidad.

La interacción del modelo con la metodología de búsqueda tabú fue a través de parámetros internos que indicaban, para cada orden, en que torno se debía procesar. Estos parámetros eran cargados externamente basados en el ordenamiento previo del swage y luego se liberaban para que el modelo pudiera manipularlos. Finalmente, una vez hallada la solución pseudo-óptima de la búsqueda, los parámetros mencionados se exportaban a Excel, y mediante un sencillo procedimiento se traducían en una tabla equivalente a la *Tabla 2*. De esta forma, se podía luego ingresar esa tabla al modelo de simulación, sin emplear el paquete de optimización, para corroborar los resultados obtenidos y exportar los indicadores necesarios.

IV.5 ♦ DEFINICIÓN DE INDICADORES

La existencia de indicadores claros, precisos y medibles fue de gran importancia a la hora de evaluar y comparar metodologías sobre el modelo de simulación. Su inclusión en este capítulo se debe a que su definición, incorporación al modelo y posterior evaluación fue sin duda una parte importante de la solución.

La definición de estas medidas de performance fue realizada por todo el equipo de trabajo, pero dirigida principalmente por los miembros de la empresa asesorada, ya que de ellos partían los requerimientos de una buena programación de la producción para su planta. Los indicadores consensuados se dividieron en tres grandes grupos: indicadores de cumplimiento, indicadores de planta e indicadores de inventarios.

Los indicadores relativos al cumplimiento buscan determinar cuán efectiva fue la metodología en lograr terminar las órdenes en el momento que eran requeridas. Este conjunto de parámetros fue:

- ▶ **Cumplimiento** [%]: $C = \frac{\text{cantidad de órdenes terminadas en fecha}}{\text{cantidad de órdenes totales del mes}}$
- ▶ **Órdenes sin cumplir** [lista]: corresponde al listado de órdenes que no se cumplieron en tiempo y forma con la metodología aplicada. Fue de gran utilidad en la comparación entre metodologías ya que permitió contrastar exactamente cómo había trabajado cada técnica.

- ▶ **Órdenes sin cantidad mínima** [lista]: casi igual al anterior, pero aquí se excluyen las órdenes que se cumplieron fuera de tiempo. Es decir, en esta lista figuran las órdenes que a fin de mes no contaban con su cantidad mínima producida, dejando sin efecto la fecha de entrega pactada. Se empleó de forma similar al anterior.
- ▶ **Días promedio de atraso** [días]: este parámetro es calculado por el modelo considerando cuándo se hubiera terminado cada orden y calculando el retraso respecto de la fecha pactada para luego promediarlo.²⁵
$$DPAt = \frac{\sum_{i=1}^n \min(0, FechaFin - FechaPactada)}{n}$$
- ▶ **Días promedio de adelanto** [días]: este parámetro es de alguna manera el “simétrico” del anterior y refleja cuán holgadamente se llegó a cumplir con las órdenes que se entregaron en fecha. No penaliza retrasos.
$$DPAd = \frac{\sum_{i=1}^n \min(0, FechaPactada - FechaFin)}{n}$$

Los indicadores de planta se emplearon para analizar la performance de cada una de las metodologías en relación con el buen uso de las máquinas. La información que sintetizan puede subdividirse en dos grandes grupos: la utilización y los setups. Los indicadores se calcularon de la siguiente manera:

- ▶ $Utilización = \frac{horas\ trabajadas}{horas\ totales}$
- ▶ $Setups = \sum horas\ utilizadas\ para\ setup$

Estos dos índices se aplicaron a los siguientes procesos o conjuntos de procesos:

- ◆ Swage
- ◆ Torno 1
- ◆ Torno 2
- ◆ Torno 3
- ◆ Torno 4
- ◆ Fosfatizado
- ◆ Tornos
- ◆ Planta
- ◆ Conveyor 1
- ◆ Conveyor 2

²⁵ El empleo de la función mínimo se debe a que no se consideran los “adelantos” respecto de la fecha pactada. Si una orden se termina el día que acordó o 10 días antes, para este indicador el efecto es el mismo: promedia un 0.

En el caso de aplicar los indicadores a los **Tornos**, simplemente se suman las horas trabajadas, totales o de setups de todos los tornos; equivalentemente, al calcular los indicadores sobre la **Planta**, se suman las horas de todos los procesos involucrados, excluidos los conveyors.²⁶

Esto generó un conjunto de 18 indicadores de planta – de los cuales se consideraron realmente 15 – para evaluar el desempeño de cada metodología.

Finalmente, se emplearon también algunos indicadores de inventarios para evidenciar la evolución de los stocks al trabajar con las distintas técnicas. Se midió en algunos índices el **WIP** (Work In Progress, Trabajo en Progreso), concepto que toma en cuenta todos los tubos que se encuentran dentro del modelo de simulación, desde los que están en el swage hasta los que se encuentran fosfatizándose, incluyendo bancales, buffers y conveyors. Estos indicadores fueron:

- ▶ Toneladas en Buffer 1, valor máximo durante la corrida
- ▶ Toneladas en Buffer 2, valor máximo durante la corrida
- ▶ Toneladas de WIP, valor máximo durante la corrida
- ▶ Toneladas de WIP, valor medio de la corrida

Adicionalmente, se consideró un último “indicador” – uno de los más importantes para la comparación de metodologías. Este elemento de análisis fue el diagrama de Gantt proporcionado por el modelo en cada caso que, sumando al conjunto de indicadores numéricos desarrollado en este apartado, permitió evaluar y comparar las distintas técnicas seleccionadas.

²⁶ Las horas de setup en los conveyors son siempre nulas, ya que no se programaron tiempos de preparación para los transportes, de modo que esos indicadores no tienen validez.

< V > RESULTADOS OBTENIDOS

V.1 ♦ PRESENTACIÓN DE RESULTADOS

Una vez desarrollado el modelo de simulación y habiendo aplicado las distintas metodologías seleccionadas al problema de secuenciamiento, se procedió a vincular ambas partes de la solución de la forma comentada en *IV.4*. A tal fin, se tomaron los listados de órdenes de los últimos 4 meses disponibles como representativos del trabajo normal de la planta. Se evaluó la posibilidad de representar más meses en el análisis pero se determinó que con esos cuatro sería suficiente; de todas maneras, incorporar un nuevo mes a la herramienta es muy sencillo, basta con cargar el calendario de feriados y el listado de órdenes.

Para los cuatro meses mencionados se evaluaron las distintas metodologías en base a los indicadores definidos en *IV.5*, aunque el análisis más profundo se dio sobre el mes 1, principalmente por la ventaja de contar con datos reales de planta para correr la metodología actual e incluirla también en la comparación. Si bien todos los índices consensuados para evaluar cada metodología fueron tenidos en cuenta a la hora de realizar comparaciones, se mostrarán en este apartado – por cuestiones de simplicidad – dos indicadores de cada grupo. De esta manera, a continuación el trabajo se centrará en los seis indicadores que resultaron ser los más fuertes en el análisis de la performance de una metodología. Para ampliar dicha información, en el *Anexo VIII.5* pueden consultarse los resultados completos de varias corridas a modo de ejemplo, sumados a algunos diagramas de Gantt.

Afortunadamente, en apoyo a la robustez de la metodología empleada, los resultados fueron comparativamente similares para los cuatro meses evaluados. Por las razones expuestas nos centraremos en primer lugar en el Mes 1 pero sin demasiada pérdida de generalidad. En dicho mes, el

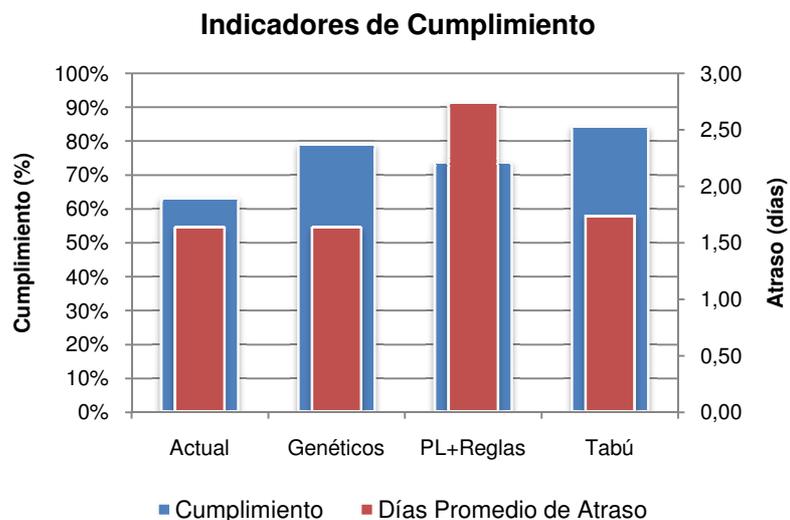


Figura 20 - Indicadores de cumplimiento

cumplimiento de las tres metodologías evaluadas fue superior al de la actual, lo que constituyó un hecho clave para aceptar la mejoría que representan (véase *Figura 20 en página anterior*). Sin embargo, en los días de atraso puede observarse que mientras algunas metodologías mantienen prácticamente igual este indicador, la programación lineal resulta muy inferior (es decir, el indicador es mayor).

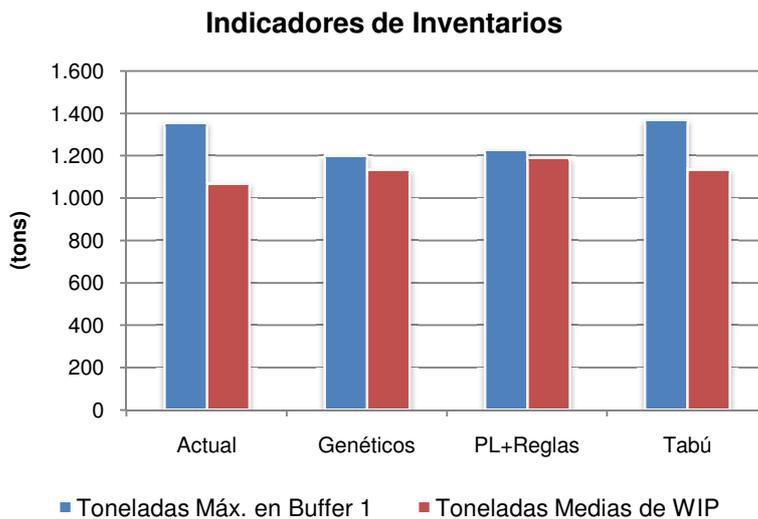


Figura 21 - Indicadores de inventarios

El comportamiento de los indicadores de inventarios fue mucho menos claro; en algunos casos se logró disminuir la cantidad de toneladas máximas en el primer buffer, pero a costa de aumentar el trabajo en proceso de la planta. Se observó, en algún punto, una correlación negativa entre ambos indicadores que queda evidenciada en la *Figura 21*. Además,

en términos generales no se hallaron grandes oportunidades de mejora en este aspecto (no se logró una reducción de toneladas en almacenamiento mayor al 10% en ningún caso).

Con relación a los indicadores de planta, se encontró que la metodología basada en programación lineal contribuía a su mejoramiento, mientras la búsqueda tabú se mostraba inferior a la metodología actual. Los algoritmos genéticos resultaron ser un punto intermedio entre las mencionadas técnicas. En la *Figura 22* pueden observarse los resultados para las cuatro corridas.

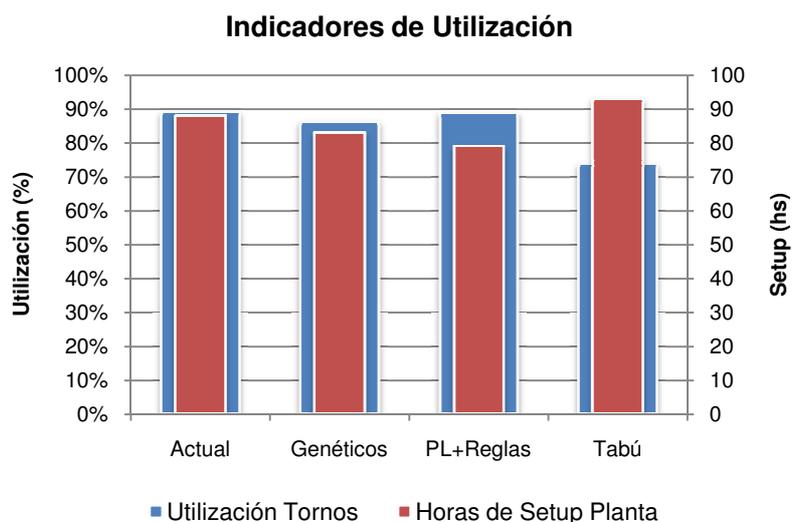


Figura 22 - Indicadores de utilización

A partir de esta primera imagen de la performance de las técnicas evaluadas surgen algunas observaciones, entre ellas:

- ▶ No todos los indicadores tienen el mismo peso. Por ejemplo, el cumplimiento y la utilización de los tornos son parámetros críticos mientras que los indicadores de inventarios no lo son
- ▶ En términos globales las metodologías programadas superan a la actual
- ▶ No existe una superioridad clara de ninguna de las técnicas
- ▶ Existen dos casos de indicadores que representan un outlier²⁷, y serán analizados posteriormente. Éstos son:
 - Los días de atraso de la metodología **PL+Reglas**
 - La utilización de la metodología **Tabú**

La baja utilización de los tornos en la metodología Tabú puede parecer, a primera vista, una debilidad de la técnica frente al resto; sin embargo, un razonamiento más detallado nos lleva a considerar también el cumplimiento. Si dicha metodología logró el mayor cumplimiento utilizando menor cantidad de horas de los tornos, se puede concluir que organizó el trabajo de los tornos de manera más eficiente que el

resto. Este concepto de eficiencia en el uso de tornos se muestra en la *Figura 23*. Allí se suma la utilización de los tornos al incumplimiento (el complementario a 100% del cumplimiento) para determinar, de alguna forma, la ineficiencia de la técnica de forma relativa. Obsérvese que las metodologías figuran ordenadas decrecientemente en función de su ineficiencia (la suma está calculada sobre la barra), es decir, a eficiencia creciente.

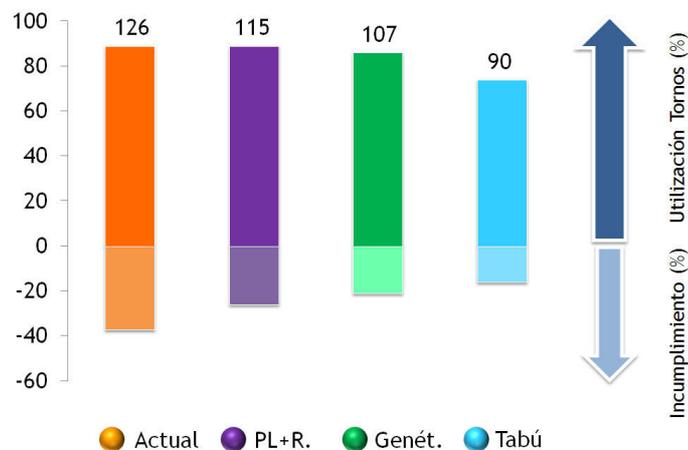


Figura 23 - Ineficiencia de las técnicas

El resto de los indicadores calculados pueden hallarse en el *Anexo VIII.5*. Debe tenerse en cuenta que, además de los parámetros cuantitativos, existieron otros criterios de evaluación de las técnicas como ser: su facilidad de programación y adaptabilidad a distintos problemas, su tiempo de procesamiento, su facilidad de comprensión y actualización y la opinión subjetiva los líderes del equipo de trabajo.

²⁷ En pocas palabras, un resultado que se aleja significativamente del resto.

V.2 ♦ ANÁLISIS Y DISCUSIÓN DE RESULTADOS

V.2.a – Comparación Entre Las Diferentes Metodologías

En función de los resultados obtenidos para cada una de las metodologías, se resolvió que la implementación de una de las seleccionadas sería beneficiosa y se entró en la etapa de selección. A tal fin, se descartó la técnica de algoritmos genéticos por las siguientes razones:

- ♦ Era la más compleja de programar y adaptar a otras plantas
- ♦ En casi todos los aspectos cuantitativos (indicadores) era superada por alguna otra metodología
- ♦ Su tiempo de procesamiento era demasiado prolongado

De esta manera, se redujo el problema a la comparación de dos técnicas entre sí. En la *Figura 24* se muestra un gráfico radial que resume los puntos fuertes y débiles de las dos técnicas restantes evaluadas para el Mes 1. Dado que los indicadores se encuentran en unidades muy diversas, la escala empleada para la comparación es el porcentaje de mejora respecto de la metodología actual (puede ser negativo). El valor nulo, cuando la performance en el indicador es igual al de la metodología actual, se ubica en la tercera línea contando desde el exterior. Los valores más cercanos al centro representan un valor del indicador más negativo que los valores cercanos a los vértices del hexágono.

Este mapa de las técnicas muestra que, mientras PL+Reglas posee puntos fuertes en los indicadores de planta, la metodología Tabú es superior en los indicadores de cumplimiento, siendo relativamente equivalentes para los índices de inventarios. Sin embargo, la gran diferencia en los días promedio de atraso parece inclinar la balanza hacia la técnica basada en Búsqueda Tabú.

Para obtener una visión más completa de las dos técnicas en cuestión, es importante considerar la diferencia en los indicadores

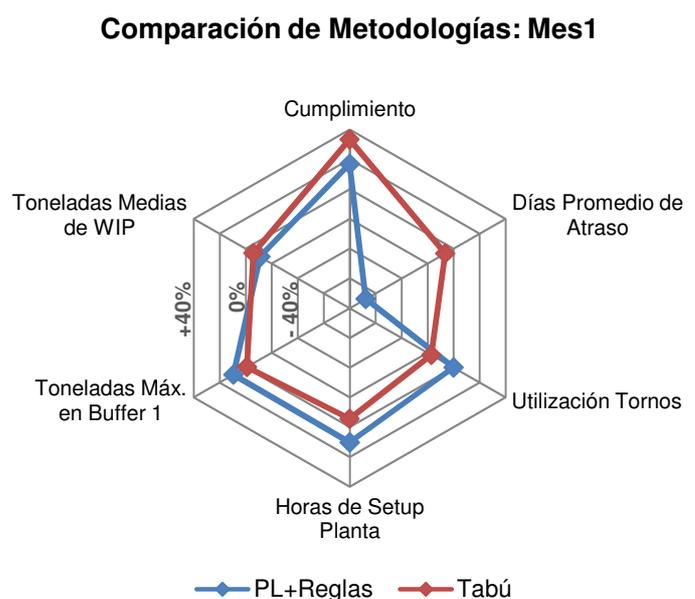


Figura 24 - Comparación de metodologías: Nov

principales para cada uno de los meses simulados. A tal fin, véanse las *Figuras 25, 26 y 27* que resumen, para los cuatro meses, tres indicadores clave.

Comparación de Metodologías: Cumplimiento

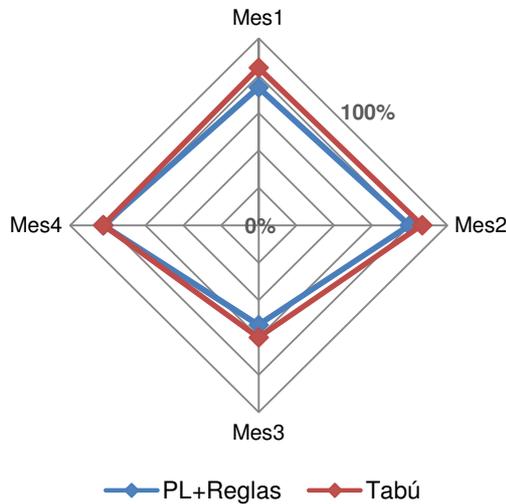


Figura 25 - Comparación de cumplimiento

Comparación de Metodologías: Días de Atraso

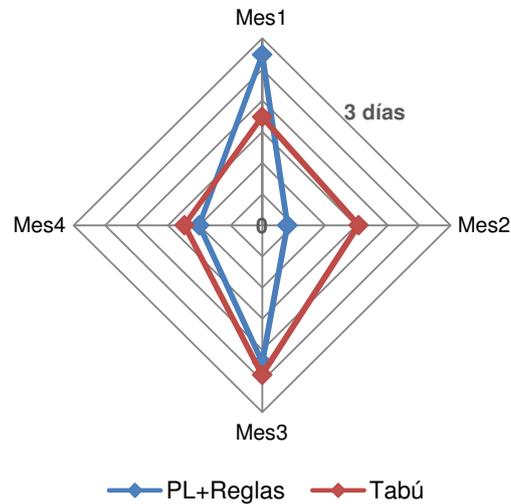


Figura 26 - Comparación de días de atraso

Su lectura es equivalente a la de la *Figura 23*, con la excepción de que en este caso cada indicador está medido en sus unidades.

A efectos de la comparación, la mayoría de los gráficos de los indicadores para los cuatro meses simulados resultan similares a los mostrados en las *Figuras 25 y 27*: la técnica que arroja mejores resultados para un mes suele estar siempre por encima de su competidora (o en el peor de los casos, igualadas).

Comparación de Metodologías: Utilización

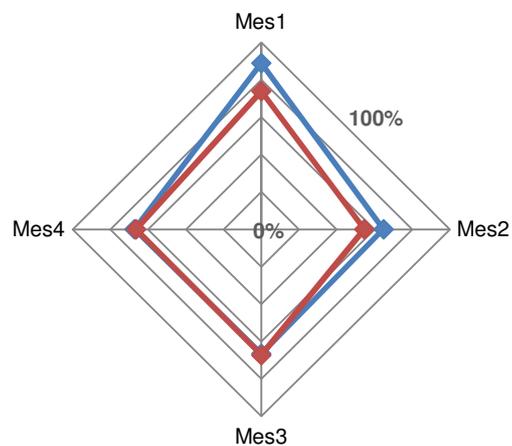


Figura 27 - Comparación de utilización

Sin embargo, el caso de los días de atraso tiene la característica particular de que, según el mes, las técnicas se comportan de manera diferente. Se puede ahondar aún más en este resultado irregular si se observa la *Figura 28* en la página siguiente.

En el eje vertical se muestra la resta de días de atraso de PL+Reglas menos días de atraso de Tabú. Lógicamente el resultado es un número que puede ser positivo o negativo según

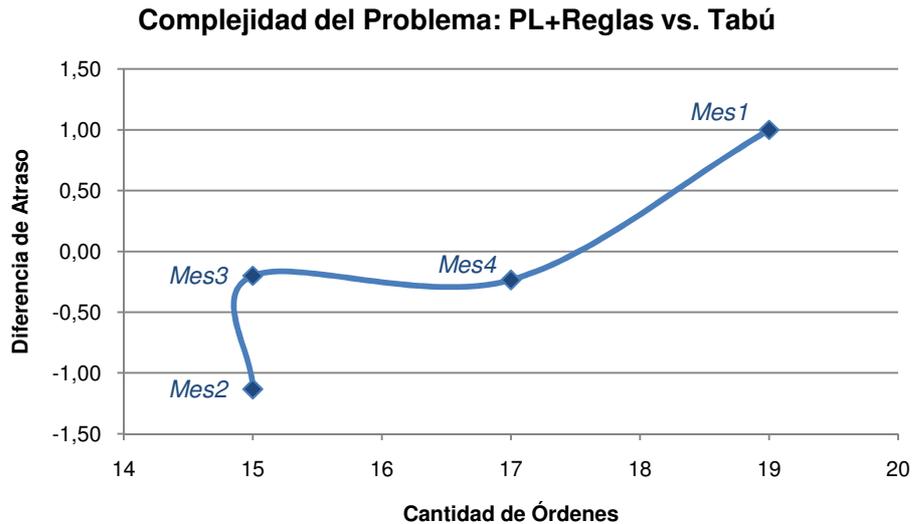


Figura 28 - Complejidad del problema

la performance de cada una de las metodologías en los meses en cuestión. A su vez, esta información se complementa con la cantidad de órdenes que figuraban en el listado para cada mes, como una medida de lo complejo o lo “grande” que era el problema a resolver. Se puede observar que hay una cierta relación directa entre la diferencia de días calculada y el número de órdenes. Esto expresa que a medida que el problema crece, la solución de búsqueda Tabú se vuelve comparativamente mejor a la de programación lineal, con las simplificaciones del caso²⁸.

Este hecho lleva a tratar de definir, al menos de forma cualitativa, una medida de la complejidad del problema, para poder seleccionar la técnica adecuada en función del tamaño no sólo de este caso en particular sino también de los demás problemas que se pretenda resolver con dicha metodología.

V.2.b – Una Medida de la Complejidad del Problema

Cuando se aborda la tarea de definir cuan complejo es un problema de scheduling comparativamente con otro, se encuentra que es muy difícil hallar una variable o un indicador que defina dicha complejidad, principalmente debido a la cantidad de características a considerar y a la diversidad de las mismas. A tal fin, en este apartado se tomarán en cuenta algunos de los parámetros esenciales de un problema de programación de la producción, tal que permitan determinar de manera cualitativa la complejidad del mismo y

²⁸ Es decir, considerando días de atraso como el indicador clave de performance y número de órdenes como el indicador de tamaño del problema.

orienten al lector en la mejor forma de resolverlo. A continuación se presenta el listado de estas características centrales:

- [a] **Cantidad de órdenes.**
- [b] **Cantidad de productos/ítems distintos** a procesar.
- [c] **Cantidad de etapas:** en el caso analizado se trató de un problema de 3 etapas.
- [d] **Cantidad de máquinas distintas por etapa:** en el caso analizado sólo una etapa (el torneado) poseía más de “una máquina” a fines prácticos.
- [e] **Restricciones de procesamiento** entre las distintas máquinas de la misma etapa: como en el caso de los rangos de los tornos, es decir, no todas las máquinas de la etapa pueden procesar todos los productos.
- [f] **Restricciones no matriciales:** restricciones particulares del tipo “esta máquina no puede hacer el mismo producto que aquella porque este operador es inexperto”, u otro tipo de limitaciones que no puedan modelizarse matricialmente, que además sean específicas y que no puedan obviarse por la relevancia de sus efectos.
- [g] **Setups matriciales:** existencia o no de tiempos de setup que dependen tanto del producto precedente como del posterior.
- [h] **Diferencia entre los órdenes tecnológicos** de los distintos productos: básicamente según la semejanza de los vectores que representan dicha característica.
- [i] **Orden tecnológico que repite al menos un centro:** existencia de productos que requieran pasar dos o más veces por el mismo centro en momentos distintos de su orden tecnológico.
- [j] **Parámetros estocásticos:** necesidad de modelizarlos de manera probabilística por su variación intrínseca.
- [k] **Reprocesos:** necesidad de modelizarlos según su importancia.
- [l] **Importancia de fallos** y necesidad de tener en cuenta sus efectos.
- [m] **Buffers intermedios:** cantidad y magnitud de inventarios que permitan “desconectar” etapas del proceso.
- [n] **Partición de órdenes:** posibilidad o no de procesar parte de una orden y cambiar sin haberla terminado.

Las variables enunciadas definen de alguna manera el tipo de problema de scheduling, al menos en lo que respecta a problemas similares al tratado en /V. En la mayoría de los casos, es evidente la dirección en la que cada una de las variables aumenta la complejidad del problema, aunque en algunas

ocasiones podría ser confuso, como por ejemplo en I (buffers). De todas formas, estos parámetros podrían agruparse de distintas maneras para permitir una observación guiada que permita clasificar los problemas de forma más eficiente.

En la *Figura 29* se muestran algunas de las variables del listado anterior agrupadas según la forma en la que aumentan la complejidad del problema. Vale remarcar que, en la dirección adecuada, todas generan un problema más difícil de resolver, pero lo hacen de distintas maneras.



Figura 29 - Distintas complejidades

Las primeras generan un problema de mayor tamaño;

las segundas generan un problema con menos libertades; las últimas dificultan el tratamiento del problema. Según qué grupo de variables sobresalga en el problema, el enfoque adoptado para resolverlo deberá modificarse: si se trata de un gran espacio de solución, las heurísticas de búsqueda global serán las más adecuadas; en problemas muy restringidos, probablemente una programación lineal (o una heurística local) logre los mejores resultados; en el último caso quizá deba recurrirse a técnicas basadas en simulación.

Debe considerarse también otra dimensión de la complejidad de los problemas de scheduling, que es la posibilidad de dividir el problema en subproblemas de menor tamaño (usualmente abarcando una sola etapa cada uno), es decir, el nivel de integración de las etapas. A mayor integración, menor es la posibilidad de subdividir el problema y más complejo se vuelve su tratamiento. En este aspecto, cobran particular relevancia las características de los buffers intermedios, que permiten desconectar las etapas y trabajar sobre la optimización de cada una de manera casi independiente. Además, también se puede vincular la semejanza entre el orden tecnológico de los distintos productos como una variable que colabora con el éxito de la subdivisión de problemas. Sin embargo, si lo que se busca es un óptimo global, vale recordar que con estas técnicas rara vez se podrá hallar (y aún menos demostrar) la optimalidad en una solución al problema completo.

En función de lo comentado, se propone el siguiente esquema de agrupación de variables (*véase Figura 30 en página siguiente*) según la forma

en que complejizan el problema. Para cada aspecto se tomaron en cuenta únicamente las dos variables más relevantes.



Figura 30 - Esquema de agrupación de variables

Se tiene entonces una visión más completa de la complejidad de un problema de scheduling y sus “componentes”, lo que permite clasificar y abordar los distintos problemas según sus características dominantes. En pos de lograr esa vinculación entre tipo de problema y técnica para abordarlo, se analizaron los casos de aplicación referenciados en *VII.2*. A su vez, esa información se combinó con las pruebas realizadas sobre el modelo del caso base de esta tesis y la performance de cada una de las técnicas en los distintos problemas.

Como conclusión principal, se encontró que existe cierta relación entre las características de la complejidad del problema y las técnicas más exitosas para su resolución, especialmente si el universo de problemas a analizar está circunscripto a los de programación de producción de plantas similares a la tratada aquí. Habiendo definido en este apartado una medida cualitativa de la complejidad del problema, resta entonces vincular dicha medida con las técnicas analizadas para obtener un método general de selección de metodologías candidatas a resolver un cierto problema. Dicho objetivo será uno de los ejes de la sección *VI.2*.

< VI > CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

VI.1 ♦ RECOMENDACIONES EN BASE AL PROYECTO

En función de los resultados expuestos, se realizó una recomendación a la empresa asesorada como cierre del proyecto de servicios tecnológicos citado en *IV*, que se resume a continuación a modo de introducción a las conclusiones de la tesis.

En primer lugar, uno de los principales aportes del estudio es el mejor entendimiento del funcionamiento propio de la planta y de las consecuencias del proceso de programación de la producción para la misma. Pudo comprobarse por ejemplo que con los mismos recursos disponibles se pueden lograr cumplimientos mucho mayores, simplemente programando correctamente los centros; en una palabra, se comprobó concretamente la sospecha de que existe un espacio de mejora en el tema.

La combinación de características propias del problema – que podría sintetizarse como “pocas órdenes y muchas restricciones particulares” – genera un hecho algo curioso que fue observado durante el trabajo. Básicamente, las condiciones están dadas para que el ser humano pueda manejar el problema relativamente bien y, conociendo suficientemente las restricciones, obtener soluciones del mismo nivel que las metodologías propuestas – o incluso mejores. Este hecho trae aparejadas dos conclusiones:

- Por un lado, se tiene un punto de comparación razonablemente bueno para las metodologías ensayadas y se pueden comprender todas las decisiones que toman las distintas técnicas analizando las salidas orden por orden (en base a los diagramas de Gantt).
- Por otra parte, surge el interrogante sobre la necesidad o no de un sistema de programación que no pueda superar por mucho al programador mismo. La necesidad del sistema no viene dada entonces por problemas como el de esta planta en particular sino por la escalabilidad, versatilidad y velocidad que proveería un sistema de scheduling global para las plantas de roscado de la compañía.

La recomendación respecto de la definición de la técnica concreta a aplicar se centra en las metodologías de Programación Lineal y Búsqueda Tabú. La *Tabla 3* resume los principales *drivers* a evaluar para la decisión final. Vale aclarar que algunos de los parámetros forman parte de la recomendación pero no se vinculan con el análisis desde el punto de vista académico.

<i>Driver</i>	PL + Reglas	Búsqueda Tabú
Indicadores Objetivos	La técnica mejora significativamente los indicadores respecto de la metodología actual; sin embargo no supera a la búsqueda Tabú.	El programa logra indicadores levemente superiores a los del resto de las metodologías, especialmente cuando se analizan de manera combinada.
Tiempo de Corrida	Suele correr muy rápidamente, lo que es muy ventajoso especialmente para el manejo de problemas de mayor tamaño (más variables).	Posee un tiempo de corrida razonable para encontrar buenas soluciones.
Comprensión del Algoritmo	Es una metodología muy fácil de comprender, lo que permite su entendimiento por parte de quien la aplica y su rápida modificación.	El optimizador es una “caja negra” de difícil comprensión, ya que emplea una combinación de heurísticas avanzadas.
Esfuerzo Requerido para Mejoramiento	La implementación de mejoras requiere un esfuerzo de algún tipo por parte de la compañía.	El software depende de un tercero que realiza mejoras y actualizaciones sin requerir nuevas inversiones de personal o tiempo.
Facilidad de Implementación	Su implementación es relativamente sencilla; sólo requiere el diseño y la codificación del programa.	Su implementación requiere la compra de un software y la capacitación adecuada al personal.

Tabla 3 - Drivers para la comparación de las técnicas seleccionadas

En adición a esta recomendación, se pueden obtener también algunas conclusiones del estudio del problema de ésta planta de roscado en particular, que se sintetizan a continuación.

Como primer punto, cabe destacar que el comportamiento observado en los dos indicadores de inventarios seleccionados (Buffer 1 máximo y WIP medio) no fue casual – *véase Figura 21*. En un análisis posterior se determinó que entre ambos indicadores existe cierta correlación negativa que puede entenderse causalmente. En pocas palabras, la gran acumulación de material en el Buffer 1 permite a los tornos trabajar de forma ininterrumpida, agilizando el paso de los tubos por la planta y disminuyendo el WIP. De esta manera, la existencia de una buena cantidad de órdenes completas en dicho sector permite, en algún punto, “desconectar” el centro de swage del resto de la planta y operar los tornos de forma casi independiente. Por tal razón se considera que es recomendable, en caso de no poder programar la planta adecuadamente, adelantar trabajo en el swage y dar libertad a los tornos para permitir un mejor desempeño global de la planta. En efecto, se investigó que en la realidad estas prácticas eran frecuentes en dicha planta, lo que valida robustamente la conclusión a la que se arribó luego del estudio.

Como segundo punto, cabe mencionar un tema que no se ha tratado con profundidad en el desarrollo del proyecto, pero que tiene cierta importancia: la partición de órdenes. Por partición de órdenes debe entenderse el hecho de que una misma orden sea procesada en dos tornos distintos. Debido a la complejidad que representa aplicar una metodología de optimización a un problema de secuenciamiento que permita la partición de órdenes, se definió dentro del alcance del proyecto que no se incluiría esta posibilidad en todas las técnicas, sino que, en todo caso, se programaría una vez definida la metodología a implementar. De alguna forma, este hecho permitió medir “con la misma vara” a todas las alternativas, pero luego se corroboró que permitiendo la partición de órdenes tanto la programación lineal como la búsqueda tabú mostraban resultados superiores, principalmente reflejados en los indicadores de cumplimiento. Este hecho hace suponer que fue una equivocación no admitir dicha posibilidad en el planteo inicial, error a partir del cual se puede concluir que en problemas muy restrictivos (como este) debe permitírsele a la heurística o algoritmo la mayor libertad posible dentro de los parámetros del sistema; de otro modo, su rango de acción, ya limitado por el problema mismo, se vuelve aún más pequeño por nuestras simplificaciones, y ese hecho se ve reflejado en resultados de inferior calidad.

Como tercer punto, y en contraposición con el anterior, vale destacar el acierto que representó programar una regla dinámica en el centro de trabajo fosfatizado/cuplado. Su performance superadora no fue de ningún modo casual; se debió principalmente a que la regla, a diferencia de una programación fija, le permitía alternar entre distintas órdenes sin completarlas necesariamente, aprovechando el material disponible en el Buffer 2 y evitando esperar a los tornos, cuyo tiempo de procesamiento es mucho mayor al del fosfatizado. Sin embargo, y como era de esperarse, este tipo de reglas no funcionó en absoluto aplicado a los tornos, ya que la complejidad de dicha etapa exige una programación previa que tenga en cuenta el resto del proceso. Como conclusión, cuando la simplicidad y el dinamismo juegan un rol preponderante y los tiempos de setup son relativamente bajos en promedio, puede ensayarse como solución parcial una heurística de construcción.

VI.2 ♦ CONCLUSIONES DEL PROYECTO

Durante esta tesis se han presentado distintas técnicas de optimización para el problema de programación de la producción. Se han mostrado varios resultados obtenidos para un caso base de aplicación, el cual fue analizado en profundidad, y se han discutido posibles formas de encarar el problema en otros casos similares. En este apartado se cierra esta tarea junto con el eje central del proyecto referido a criterios de definición de metodologías.

Como primer punto, cabe destacar que el enfoque aportado aquí para resolver el problema fue satisfactorio desde dos puntos de vista:

- ◆ Permitió resolver adecuadamente el problema y lograr recomendaciones de valor para el caso
- ◆ Dio lugar a un análisis más profundo sobre el tema que el que se hubiera realizado en caso de adoptar una solución menos elaborada

Por tal razón, se entiende que una de las principales conclusiones de este trabajo es la validación del método empleado en la resolución del problema que, si bien no fue totalmente original, fue al menos innovador.

Además, en función de las distintas variables que caracterizan a un problema típico de scheduling, se logró definir un conjunto de ejes que llevan a distintos tipos de complejidad. A su vez, se identificaron las principales técnicas empleadas en la resolución de problemas de scheduling según su grado de avance en cada uno de los ejes, y se completó dicha investigación mediante la resolución de distintos problemas con el modelo desarrollado. Los resultados obtenidos se presentan en la *Figura 31*.

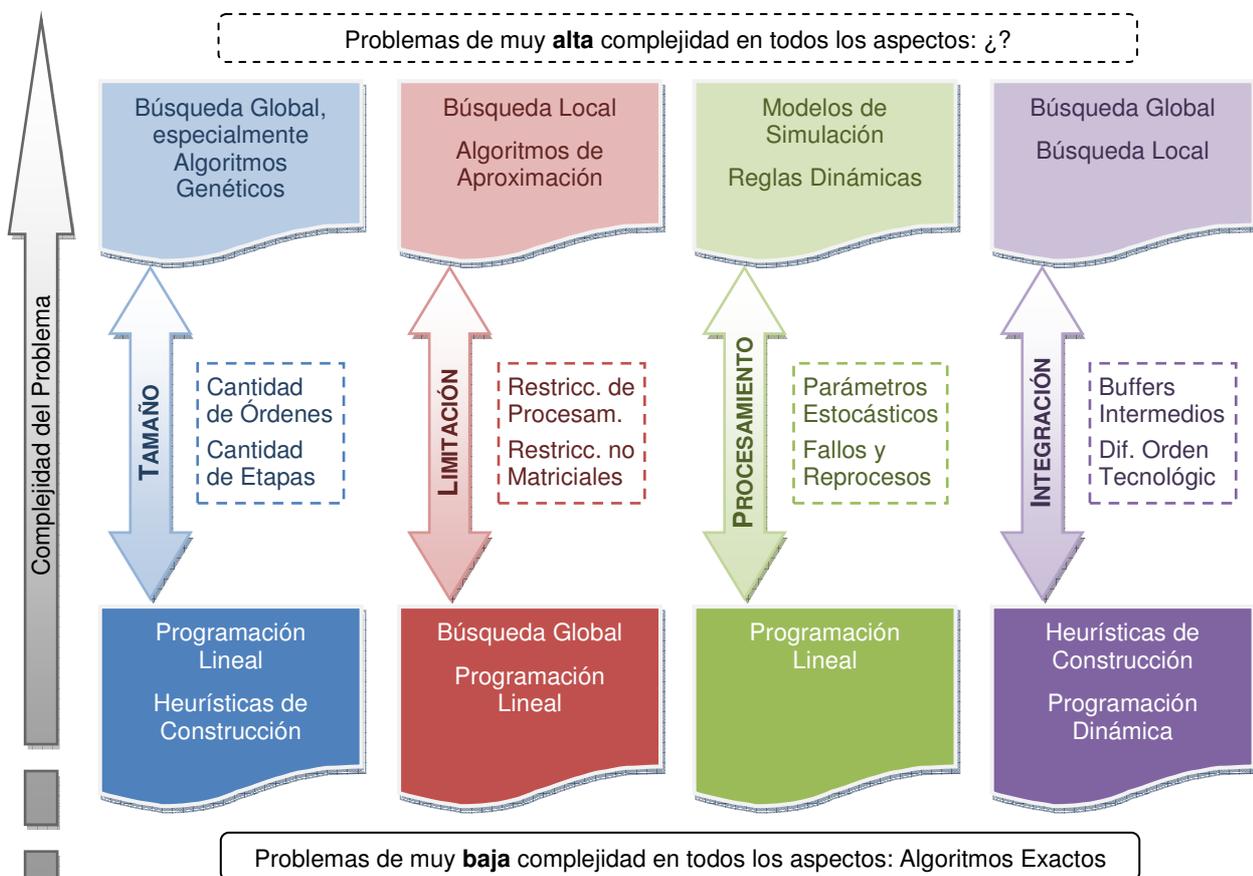


Figura 31 - Técnicas recomendadas según características del problema

La figura anterior representa una síntesis de la relación entre las características del problema y las técnicas a emplear para resolverlo. Nótese que, por ejemplo, en problemas muy sencillos en todos los aspectos lo más conveniente es generalmente desarrollar un algoritmo exacto, como en el caso del Algoritmo de Johnson. Cuando el problema se vuelve más complejo, se debe analizar cuáles son las características que lo hacen difícil de resolver. En función de esas características y lo comentado en *V.2.b*, se deben determinar los aspectos más críticos de dicha complejidad. En el caso base de estudio tratado aquí, el eje central de su complejidad era sin duda el de “limitación”, con lo cual es razonable que con una búsqueda local se hayan obtenido muy buenos resultados. A su vez, se aclaró que para problemas de mayor tamaño sería necesario pasar a heurísticas globales, lo que se condice con la *Figura 31*. Vale remarcar que en **Modelos de Simulación** se hace referencia al modelo como solución al problema y no como plataforma de evaluación (que es el caso de esta tesis).

Otro punto a destacar es el perteneciente a la referencia [22], ya comentado en *II.4*. En problemas como tal, con gran variabilidad y etapas muy fáciles de separar, usualmente son muy convenientes las heurísticas de construcción y las reglas dinámicas como se demuestra en dicho artículo. Sin embargo, es difícil recomendar una técnica para problemas que sean complejos desde todos los puntos de vista. Probablemente no exista una metodología que pueda ocupar ese lugar hoy, pero se cree que en algún momento los algoritmos de aproximación se posicionarán como alternativa. Este tema será tratado en el apartado *VI.4*.

Como última conclusión, cabe decir que en la mayoría de los casos, cuando los problemas de scheduling pasan de cierta dimensión, siempre conviene emplear técnicas que permitan tener en cuenta el problema completo y no sólo una etapa del mismo, sin importar el aspecto específico definido. Esto se debe a que la visión parcial de un proceso complejo puede ser muy engañosa, con lo cual las técnicas locales definidas pueden terminar por actuar de manera contraria a la del objetivo programado. La programación de la producción es un problema centrado en la interacción entre recursos a través del tiempo, y esa interacción debe ser contemplada para obtener una solución satisfactoria al problema.

VI.3 ♦ EVALUACIÓN DEL ÉXITO DEL PROYECTO

Se puede decir que el proyecto ha sido exitoso, en cuanto se ha conseguido cumplir, en mayor o menor medida, con los criterios planteados en *I.5*. En primer lugar, durante el proceso de elaboración de la presente tesis se

adquirió un valioso aprendizaje en la temática y en técnicas de optimización, modelización y simulación. La solución generada para el caso de aplicación fue satisfactoria, con los aciertos y errores mencionados en VI.1; sin embargo, la mejora en los indicadores definidos fue evidente. Finalmente, se considera, desde este humilde lugar, que se ha aportado un enfoque interesante en el tratamiento de problemas de scheduling y una conclusión original en cuanto a su clasificación y complejidad.

Es importante también hacer foco en las acciones que deberían tomarse para hacer de este documento un aporte más sólido a la teoría del scheduling. Principalmente, deberían estudiarse otros problemas mediante la aplicación de los criterios definidos en estos últimos párrafos; medir las variables destacadas, clasificar el problema y verificar – mediante la misma metodología empleada aquí – que las técnicas que mejores resultados otorgan coinciden con las de la *Figura 31*. Si bien la definición de una metodología y la resolución del problema planteado ya se desarrollaron a lo largo de esta tesis, la replicación de este proyecto en algunos casos más permitiría fortalecer las conclusiones obtenidas y generar una potente herramienta de análisis para problemas de programación de la producción. La definición concreta de cuán complejo es un problema de scheduling particular es un tema de gran relevancia académica y técnica que está abierto, y estas líneas de análisis propuestas tenderían a trabajar sobre el mismo.

VI.4 ♦ FUTURAS LÍNEAS DE INVESTIGACIÓN

En el desarrollo de esta tesis se han planteado algunas ideas para futuros trabajos que se desea retomar aquí como cierre del proyecto. Una de ellas se encuentra vinculada al problema de la optimización y otra al de la modelización, ambas temáticas fundamentales de este documento.

Los algoritmos de aproximación representan sin duda el futuro en la resolución de problemas clásicos de scheduling. Investigaciones recientes en el tema demuestran que pueden resolver de forma casi óptima los problemas expresados matricialmente y sin restricciones tan particulares como las tratadas en el caso de esta planta de roscado. Es probable que los avances continúen y se termine reemplazando la solución clásica de programación lineal por algoritmos de aproximación más veloces, versátiles y con un grado de optimalidad casi tan bueno como el de aquella. Sólo hace falta demostrar su idoneidad para problemas de scheduling, y es justamente esa la tarea que se plantea aquí como línea de investigación. Una forma de lograrlo podría ser mediante la misma metodología que se empleó aquí: diseñando una plataforma virtual de evaluación y comparando los resultados de distintos tipos de

algoritmos de aproximación contra los del problema modelizado por programación lineal.

La otra línea de investigación a proponer está mucho menos explorada al día de hoy, pero a su vez representa un quiebre mayor en el estudio del problema de scheduling, probablemente del orden de un cambio de enfoque como los comentados en *II.1*. La idea se trata básicamente de modelizar el problema abandonando la ya clásica simulación discreta y empleando métodos de simulación basada en agentes²⁹. De esta manera, se modificaría conceptualmente el planteo del problema, pudiendo programar las decisiones a tomar por el material mismo durante su procesamiento, las modificaciones particulares de distintos centros de trabajo, y un conjunto de reglas y meta-reglas dinámicas que generaría un sistema capaz de resolver problemas de altísima complejidad. A su vez, este tratamiento debería correlacionarse en la realidad con sistemas automatizados que permitan reproducir los resultados obtenidos en el modelo en la propia planta. Puede parecer algo aventurado, pero dada la creciente complejización de las actividades productivas debido a los grandes avances tecnológicos, es posible que en un futuro la única manera de resolver aceptablemente un problema de scheduling importante sea mediante simulación basada en agentes; y, en ese contexto, quien primero desarrolle la herramienta tendrá una clara ventaja.

²⁹ Para una breve reseña sobre los distintos enfoques de simulación véase *VIII.4*

< VII > BIBLIOGRAFÍA

VII.1 ♦ BIBLIOGRAFÍA GENERAL

- [1] Banks, J. y otros. 2004. **Discrete-Event System Simulation**. 608 páginas. Editorial Pearson Prentice Hall. ISBN: 9780131446793.
- [2] Eco, U. 2001. **Cómo se hace una tesis**. 240 páginas. Editorial Gedisa. ISBN: 9788474328967.
- [3] Kelton, W. y otros. 2003. **Simulation with Arena**. 668 páginas. ISBN: 9780072919813.
- [4] Sterman, J. 2000. **Business Dynamics: Systems Thinking and Modeling in a Complex World**. 1008 páginas. Editorial McGraw-Hill/Irwin. ISBN 9780072389159.
- [5] Winston, W. 2004. **Investigación de Operaciones**. 1424 páginas. Editorial Cengage. ISBN: 9706863621.

VII.2 ♦ DOCUMENTOS TÉCNICOS ESPECÍFICOS

- [6] Abspoel, S. y otros. 2000. **Simulation optimization of stochastic systems with integer variables by sequential linearization**. 2000 Winter Simulation Conference.
- [7] Almeder, C. y otro. 2007. **A hybrid simulation optimization approach for supply chains**. Eurosim. ISBN: 9783901608322.
- [8] Carson, Y. y otro. 1997. **Simulation optimization: methods and applications**. 1997 Winter Simulation Conference.
- [9] Franco, J. y otro. 2005. **Aplicación del simulated annealing al problema de las "n" reinas**. Scientia et Technica. Año 11. Número 29.
- [10] Gilli M. 2004. **An introduction to optimization heuristics**. Seminar: University Of Cyprus, Department of Public and Business Administration.
- [11] Gilli, M. y otro. 2007. **Heuristic optimization methods in econometrics**. Swiss Finance Institute Research Paper. Número 08-12.
- [12] Graves, S. 1999. **Manufacturing planning and control**. Massachusetts Institute Of Technology.
- [13] Gupta, J. 2002. **An excursion in scheduling theory: an overview of scheduling research in the twentieth century**. Production Planning & Control. Volumen 13. Número 2. Páginas 105-116.
- [14] Hincapié Isaza, R. y otros. 2004. **Técnicas heurísticas aplicadas al problema del cartero viajante (TSP)**. Scientia et Technica. Número 24.
- [15] Kumar, S. y otro. 2006. **Integrated simulation application design for short-term production scheduling**. IIE Transactions. Volumen 38. Páginas 737-748.

- [16] Leu, S. y otro. 2002. **A genetic algorithm-based optimal resource-constrained scheduling simulation model**. Construction Management & Economics. Volumen 20. Páginas 131-141.
- [17] Liu, B. y otros. 2004. **Classification rule discovery with ant colony optimization**. IEEE Computational Intelligence Bulletin. Volumen 3. Número 1. Páginas 31-35.
- [18] Lozano, S. 2003. **Breve introducción a GRASP (Greedy Randomized Adaptive Search Procedure)**. Universidad de Sevilla.
- [19] Mahdavinejad, R. 2007. **Multiple job-shop scheduling using hybrid heuristic algorithm**. International Journal of Mathematical, Physical and Engineering Sciences. Volumen 1. Número 1. Páginas 53-58.
- [20] Ríos Mercado, R. y otro. 2001. **Secuenciando óptimamente líneas de flujo en sistemas de manufactura**. Ingenierías. Volumen 4. Número 10. Páginas 38-45.
- [21] Romero R. y otros. 2004. **Modelo de programación de la producción para la industria del aserrío**. Revista Ingeniería Industrial (Universidad Nacional del Bío-Bío, Chile). Año 3. Número 1.
- [22] Tuncel, G. 2007. **A heuristic rule-based approach for dynamic scheduling of flexible manufacturing systems**. Del libro "Multiprocessor Scheduling: Theory and Applications", Itech Education and Publishing, ISBN: 97839026113028.

VII.3 ♦ OTRAS FUENTES DE INFORMACIÓN

- [23] Jonena, A. **Some Notes On Johnson's Algorithm For Scheduling**. Hong Kong University Of Science And Technology – School Of Engineering.
<http://www.ielm.ust.hk/dfaculty/ajay/courses/ieem513/GT/johnson.html>
- [24] Sitio Web de **XJ Technologies**. <http://www.xjtek.com/>
- [25] Google Books: **scheduling**. <http://books.google.com/books?q=scheduling&btnG=Buscar+libros&hl=es>
- [26] Moore, R. y otro. 2005. **A compendium of NP optimization problems**. Mathematics Department, Macquarie University, Sydney.
<http://www.nada.kth.se/~viggo/wwwcompendium/> (traducción a html por Viggo Kann)
- [27] Sitios Web relacionados con el **OptQuest**: <http://www.optquest.com/>,
<http://www.solver.com/xlsoptqeng2.htm>,
<http://www.mesquite.com/products/optfact.htm>.

Todas las páginas web consultadas están vigentes al día 30 de Junio de 2009.

< VIII > ANEXOS

VIII.1 ♦ ALGORITMO DE JOHNSON³⁰

El algoritmo que se presenta a continuación fue demostrado por S. M. Johnson en 1954 y constituye una de las primeras soluciones cerradas al problema de programación de la producción, surgida a partir del paradigma denominado "divide y vencerás". Es un procedimiento conceptualmente muy sencillo, que de alguna manera representa una evolución lógica del criterio más empleado en problemas de scheduling de la vida diaria.

Enumerar los supuestos de este procedimiento en cuanto al problema a tratar sería tedioso y redundante, por lo que bastará decir que es un sistema "ideal" en el sentido más amplio de la palabra, sin roturas, sin tiempos de setup, sin reprogramaciones, sin aleatoriedad y sin ninguna complejidad más allá de lo que implica ordenar trabajos en máquinas.

La idea base del algoritmo es la siguiente: en cualquier fila de personas esperando un servicio – por ejemplo el banco – el mejor desempeño del sistema se obtiene cuando los individuos que requieren servicios más cortos se ubican primero. Esto se da porque el tiempo promedio de espera se minimiza de dicha forma; el primer cliente espera cero minutos, el segundo espera muy poco, el tercero también, y sólo los últimos esperan mucho. Si hubiera un cliente con tiempo de servicio largo al inicio de la fila, todos los que lo suceden deberían esperar mucho, con lo cual el tiempo promedio de espera sería mucho mayor. La *Figura 32* ilustra el hecho con un ejemplo.

Cuando se pasa de un proceso de una etapa (como el del ejemplo) a uno en dos etapas la regla deducida pierde validez, ya que los tiempos de procesamiento en ambas etapas condicionan la velocidad, los tiempos muertos de la segunda etapa y demás variables. El algoritmo de Johnson funciona como una extensión de la regla "el más corto primero" para casos en los cuales hay 2 etapas con una máquina en cada una.

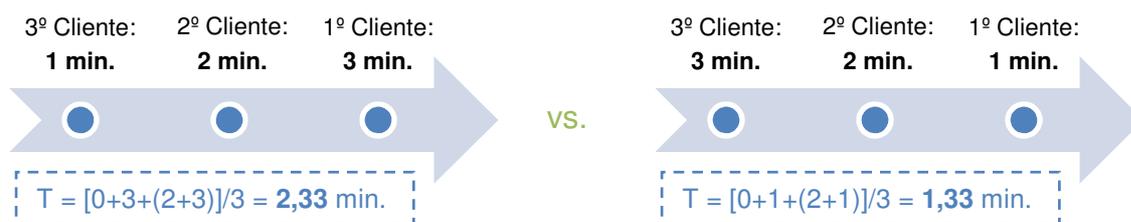


Figura 32 - Ejemplo de un proceso en una etapa

³⁰ Adaptado principalmente de [23]

Se parte de una matriz de datos que indica los tiempos de procesamiento de cada trabajo en cada etapa. Esquemáticamente, se podría representar de la siguiente manera:

Etapa \ Trabajo	1	2	...	i	...	n
1	T_{11}	T_{21}	...	T_{i1}	...	T_{n1}
2	T_{12}	T_{22}	...	T_{i2}	...	T_{n2}

Tabla 4 - Información de entrada para el algoritmo de Johnson

Se trabaja básicamente con 4 listas que pueden expresarse de manera vectorial como sigue:

- ♦ $T = \{T_{11}, T_{21}, \dots, T_{i1}, \dots, T_{n1}, T_{12}, T_{22}, \dots, T_{i2}, \dots, T_{n2}\}$ - es decir, una lista de tamaño $2n$ que contiene todos los tiempos de procesamiento del problema (todos los trabajos cruzados con ambas máquinas).
- ♦ $A = \{1, 2, \dots, i, \dots, n\}$ - lista que contiene todos los productos.
- ♦ $L1 = \{\}$ y $L2 = \{\}$ - listas vacías.

El algoritmo trabaja de buscando el mínimo de la lista T y ordenando el producto en una de las listas L según la regla esquematizada en el diagrama de la *Figura 33*.

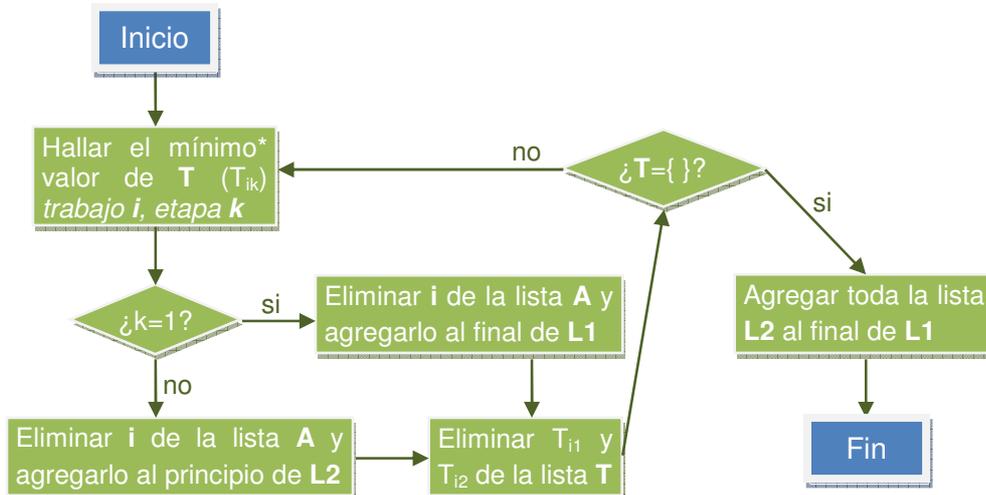


Figura 33 - Diagrama esquemático del Algoritmo de Johnson

* En caso de hallar dos mínimos se puede usar la siguiente regla de selección: si ambos son de la misma etapa, elegir el de mayor tiempo en la otra etapa; si pertenecen a distintas etapas, es indistinto, puede elegirse al azar. La misma regla es generalizable a un empate entre N mínimos y garantiza la optimalidad del algoritmo.

En resumidas cuentas el procedimiento trabaja vaciando la lista **T** de menor a mayor valor y reacomodando en las listas **L1** y **L2**. Una vez concluido el proceso se tiene que **L1** es el orden óptimo de los trabajos, y este hecho puede demostrarse matemáticamente. Es muy interesante comprobar los resultados que arroja y analizar por qué el algoritmo halla el óptimo en todos los casos – aunque no se realice una demostración matemática – pero dicha verificación queda fuera del alcance de este anexo.

Si bien no puede emplearse el algoritmo de Johnson para resolver un problema de la complejidad planteada en el presente trabajo, su comprensión es útil desde los siguientes puntos de vista:

- ▶ Da una idea de los fuertes supuestos y restricciones que debían imponerse para poder resolver problemas de scheduling con un algoritmo cerrado
- ▶ Muestra que en problemas sencillos el “sentido común” puede ser una solución, pero rápidamente deja de serlo a medida que el problema se complejiza
- ▶ Deja en claro que el mejor secuenciamiento depende fuertemente del objetivo que se persiga: si el objetivo fuera minimizar el uso de máquinas por ejemplo, todos los ordenamientos serían equivalentes ya que uno de los supuestos del problema es la ausencia de tiempos de setup (y menos aún de tiempos de setup matriciales).

VIII.2 ♦ MAPA COMPLETO DE METODOLOGÍAS DE RESOLUCIÓN

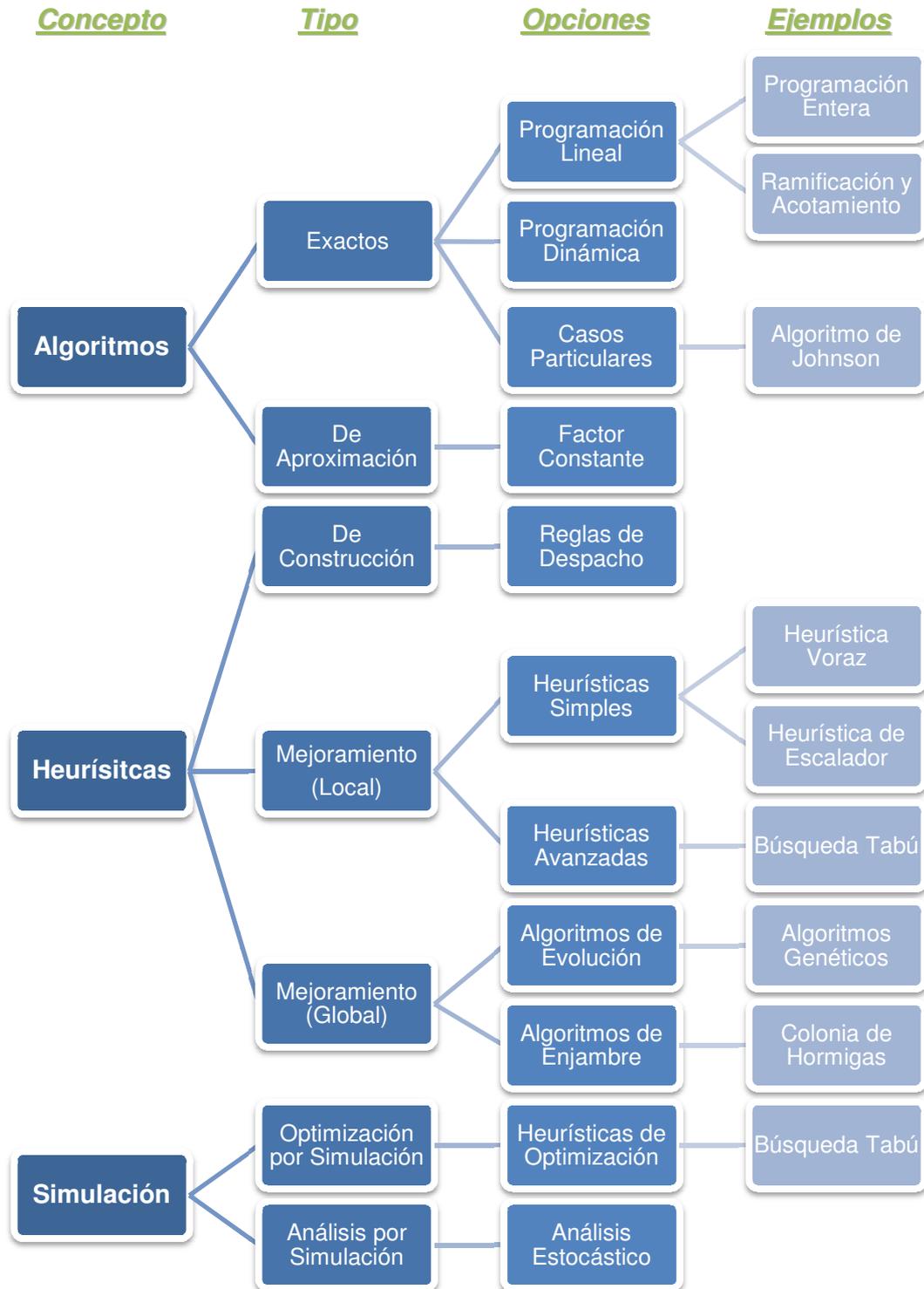


Figura 34 - Mapa completo de metodologías de resolución (elaboración propia)

VIII.3 ♦ CONFIDENCIALIDAD DE LA INFORMACIÓN.

El presente proyecto se basó en un trabajo de servicios tecnológicos realizado a una gran empresa sobre la programación de la producción de una de sus plantas. Para preservar la confidencialidad de dicha empresa se ha omitido mencionar su nombre durante esta tesis y se han modificado algunos de sus datos, como ser nombres de los tornos (omitidos), números orden (modificados), meses analizados (modificados), tiempos y productividades (modificados y en algunos casos omitidos), análisis del layout de la planta (omitido) y conclusión final del trabajo de asesoramiento (no detallada). De todas formas, estas acciones de reserva no han afectado al espíritu y el desarrollo de este proyecto de manera significativa.

VIII.4 ♦ SIMULACIÓN: DISTINTOS PARADIGMAS Y PRINCIPALES CONCEPTOS

La simulación consiste en la imitar el comportamiento de un sistema durante cierto período de tiempo. La forma de hacerlo es usualmente a través de una computadora, aunque también pueden generarse modelos físicos de simulación. Es una técnica surgida a mediados del siglo XX pero popularizada a partir de 1990 gracias a los drásticos avances en el campo de la informática.

Sus aplicaciones son realmente amplias; abarcan desde la industria del entretenimiento hasta la de la salud, pasando por la educación, entrenamiento de personal, usos militares, toma de decisiones, procesos logísticos y de manufactura, análisis de mercados y hasta modelos macroeconómicos. Sin embargo, sus bases son relativamente sencillas: se busca comprender los vínculos causales de un sistema, expresarlos mediante un modelo y simularlos para poder analizarlos mejor (entiéndase evaluar distintos escenarios, modificar parámetros, predecir el comportamiento colectivo, etc.).

Enmarcados en este concepto y con el objetivo enunciado, existen tres **paradigmas** principales que surgieron de diversas corrientes ideológicas y sirven para resolver distintos tipos de problemas.

La **simulación discreta** se basa en la idea de que existen momentos importantes dentro de un sistema, llamados **eventos**, que modifican alguna característica del mismo – o de una parte del mismo – como ser la llegada de un cliente o el final del procesamiento de un producto en una máquina. Durante el tiempo que transcurre entre esos eventos el sistema permanece inalterado. Este tipo de simulación centra su atención en los procesos y maneja **entidades** pasivas que los atraviesan bajo ciertas reglas, modificando en muchos casos sus **atributos**. De esta manera, tanto las cantidades como los tiempos son

discretos: la cantidad de entidades en cualquier proceso es siempre un número entero mientras que el tiempo avanza de evento en evento. Es usualmente el enfoque más adecuado para problemas referentes a procesos productivos, logísticos y otros de un nivel de abstracción medio-bajo.

La **simulación continua**, en cambio, se utiliza para modelizar problemas de largo plazo y alto nivel de abstracción. Es una imitación conceptual del sistema basada en variables que modifican sus valores continuamente con el paso del tiempo virtual. Manejan principalmente el concepto de **acumulador**, que es un tipo de variable que sólo puede ser modificado por **flujos** de entrada y de salida (por ejemplo, la cantidad de viviendas en el país sólo puede ser modificada por su construcción o su demolición). Normalmente es el mejor enfoque para abordar problemas estratégicos donde las características individuales no son determinantes.

La **simulación basada en agentes** es un paradigma mucho más reciente que los anteriores y, en algún punto, consiste en una generalización de la simulación discreta. La gran diferencia es que este caso las entidades que se simulan se llaman **agentes** y son activas; no tienen un camino marcado a través de ciertos procesos, sino que toman decisiones, poseen individualidad, siguen ciertas reglas propias, se relacionan y modifican a otros agentes. Lo que se busca entonces es, estableciendo conductas de agentes particulares, obtener el **comportamiento emergente** de su interacción. Su aplicación más fuerte es la simulación de mercados y sistemas biológicos, aunque su rango de acción es tan amplio que puede usarse tanto para problemas de bajo como de alto nivel de abstracción. Es una forma de simular algo más complicada pero muy completa y versátil.

VIII.5 ♦ RESULTADOS COMPLETOS DE ALGUNAS CORRIDAS

En la página siguiente se presentan los resultados de las corridas comentadas durante este trabajo. En la *Tabla 5* se muestran comparativamente los indicadores cuantitativos para todas las corridas seleccionadas, mientras en las *Figuras 35 a 38* se muestran los diagramas de Gantt para las corridas del Mes 1. Se omiten los números de orden y los indicadores de órdenes sin cumplir y órdenes sin cantidad mínima. De todas formas, puede seguirse el proceso de cada orden en particular mediante el código de colores empleado (el blanco significa “ocioso”, el negro significa “en setup” y los demás colores representan las distintas órdenes; el horizonte temporal de los diagramas es un mes).

Indicadores Principales	Mes 1				Mes 2		Mes 3		Mes 4	
	Actual	PL+R	Tabú	Genét.	PL+R	Tabú	PL+R	Tabú	PL+R	Tabú
Cumplimiento	63,16%	73,68%	84,21%	78,95%	80,00%	86,67%	53,33%	60,00%	82,35%	82,35%
Días Promedio de Atraso	1,63	2,74	1,74	1,63	0,40	1,53	2,20	2,40	1,00	1,24
Días Promedio de Adelanto	8,74	9,68	10,05	8,32	11,33	9,07	4,47	4,27	12,06	7,59
Utilización Swage	51,88%	51,76%	51,76%	51,76%	42,73%	43,00%	49,04%	49,24%	36,09%	36,17%
Utilización Torno 1	94,40%	78,76%	73,68%	85,53%	46,68%	44,65%	26,18%	76,70%	82,25%	48,77%
Utilización Torno 2	97,09%	88,52%	95,27%	71,74%	31,00%	54,79%	83,81%	53,18%	53,22%	98,96%
Utilización Torno 3	74,08%	89,40%	40,48%	95,86%	96,35%	26,57%	64,12%	70,00%	85,50%	63,77%
Utilización Torno 4	90,29%	98,74%	85,80%	91,61%	85,21%	93,05%	92,23%	69,15%	47,70%	54,41%
Utilización Tornos	88,94%	88,87%	73,76%	86,21%	64,80%	54,64%	66,52%	67,26%	67,21%	66,39%
Utilización Fosfatizado	77,62%	88,03%	76,74%	85,91%	58,26%	38,44%	66,48%	72,02%	85,15%	74,91%
Horas de Setup Swage	12	11	11	11	4	7	5	7	7	8
Horas de Setup Torno 1	16	18	18	12	12	12	8	12	12	14
Horas de Setup Torno 2	10	6	14	16	10	8	10	12	12	20
Horas de Setup Torno 3	10	14	12	12	12	12	14	8	14	12
Horas de Setup Torno 4	14	14	16	14	10	18	12	14	18	16
Horas de Setup Tornos	50	52	60	54	44	50	44	46	56	62
Horas de Setup Fosfatizado	26	16	22	18	12	8	18	16	24	20
Horas de Setup Planta	88	79	93	83	60	65	67	69	87	90
Toneladas Máx. en Buffer 1	1354	1225	1368	1198	613	707	710	878	1055	1250
Toneladas Máx. en Buffer 2	651	908	483	799	355	318	520	689	633	479
Toneladas Máx. de WIP	1714	1997	1862	2028	962	997	1128	1499	1358	1710
Toneladas Medias de WIP	1069	1188	1132	1131	498	536	689	762	776	976

Tabla 5 - Indicadores de algunas corridas

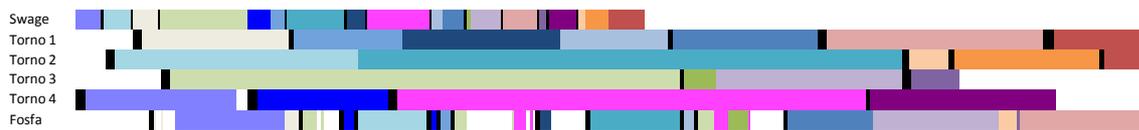


Figura 35 - Diagrama de Gantt para la metodología actual en el Mes 1

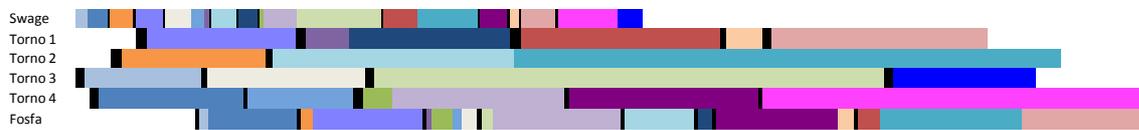


Figura 36 - Diagrama de Gantt para la metodología PL+R en el Mes 1

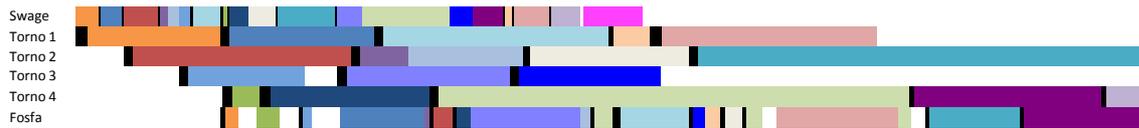


Figura 37 - Diagrama de Gantt para la metodología Tabú en el Mes 1

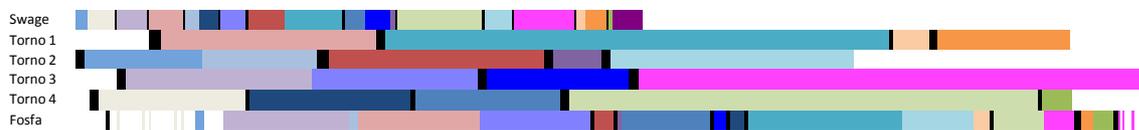


Figura 38 - Diagrama de Gantt para la metodología de alg. genéticos en el Mes 1

VIII.6 ♦ DIAGRAMA IDEF0 DEL PROCESO DE ROSCADO Y FLUJOGRAMAS ADICIONALES

Pueden consultarse como anexos documentales *a continuación*, en las páginas finales de esta tesis.

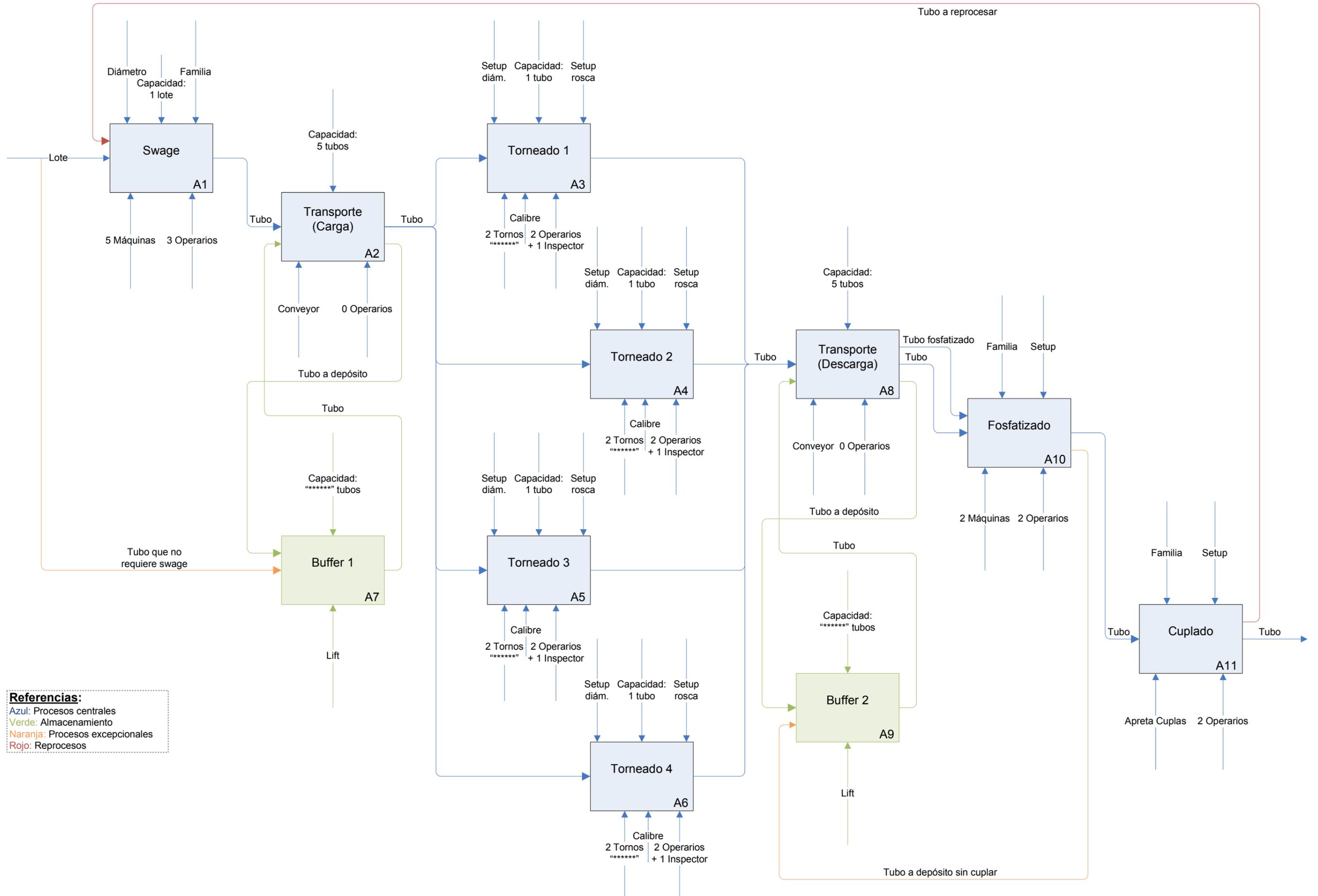


Diagrama de Flujo de la Toma de Trabajos del Fosfatizado

Programado en el modelo

2009

Hoja 1 de 2

Objetivo de la Regla de Decisión

2009

Simular la toma de trabajos del centro Fosfatizado/Cuplado mediante una regla dinámica que permita asignar trabajos al centro en base al monitoreo de algunas variables de operación del modelo, y que además sea simple y clara.

Información monitoreada del modelo:

OrdFosfa: Orden en la cual está trabajando el fosfatizado.

OrdBuffer2(i): Cantidad de tubos en el buffer posterior a los tornos, de cada orden.

OrdCant(i): Cantidad de tubos de cada orden.

OrdCantProducida(i): Cantidad de tubos terminados de cada orden.

OrdDiam(i): Diámetro de cada orden.

OrdT(i)=max(OrdTFosfa(i),OrdTCupla(i)): Tiempo unitario de proceso en el centro en cuestión.

Bancal ubicado antes del fosfatizado: largo total y largo ocupado.

Lote mínimo, fijado como parámetro del modelo.

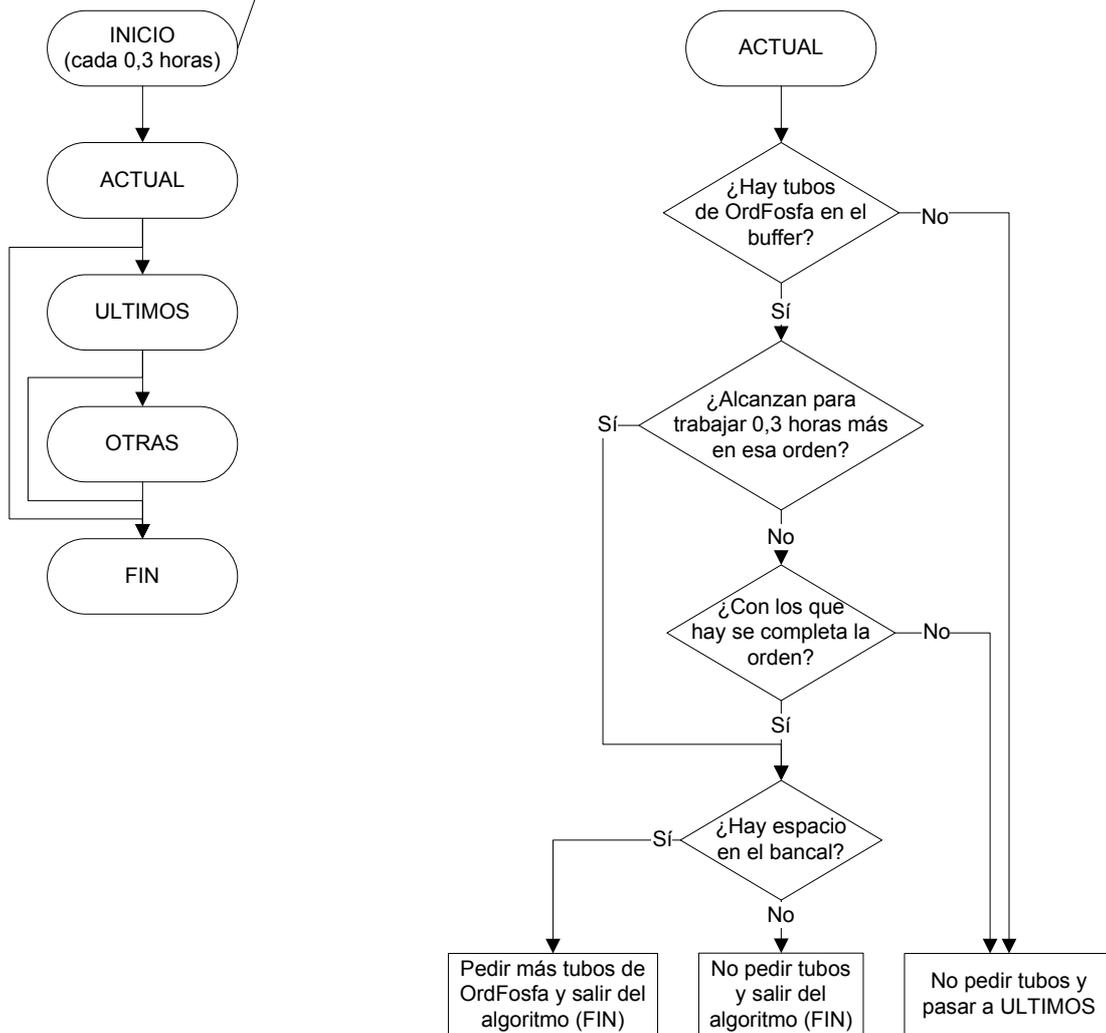
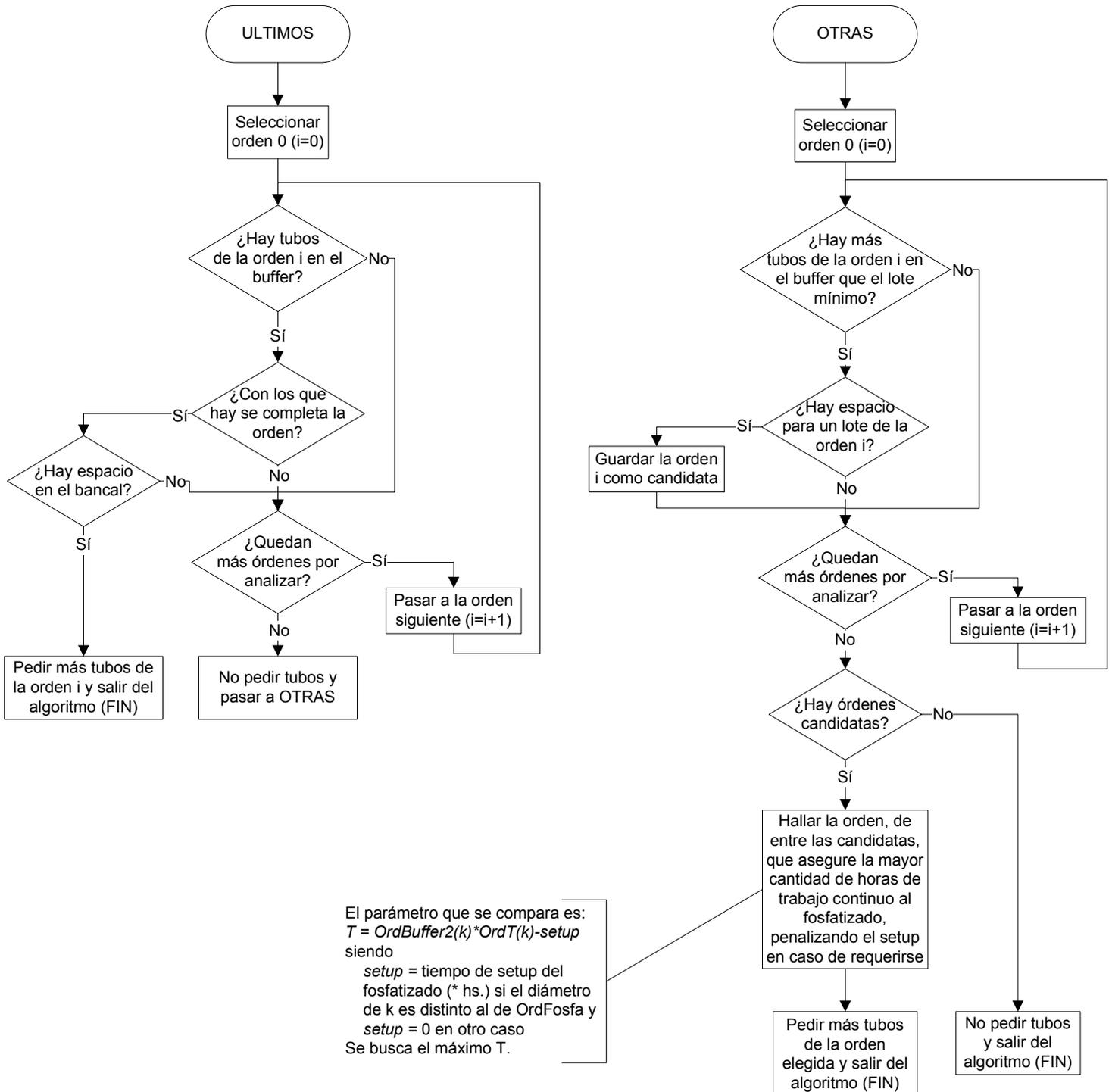


Diagrama de Flujo de la Toma de Trabajos del Fosfatizado

Programado en el modelo

2009

Hoja 2 de 2



Aplicación

El modelo aplica esta lógica cuando se automatiza el fosfatizado y se utiliza la programación por tareas (no por horas).