



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

**DESARROLLO DE UNA SOLUCIÓN LOGÍSTICA PARA LA PROGRAMACIÓN DE
OPERACIONES EN UNA COMPAÑÍA SIDERÚRGICA**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

FELIPE ANDRÉS RIQUELME NIKLITSCHK

PROFESOR GUÍA:
SR. PATRICIO CONCA KEHL

MIEMBROS DE LA COMISIÓN:
SR. JOSE MOSQUERA CADIZ
SR. DANIEL ESPINOZA GONZALEZ

SANTIAGO DE CHILE
ENERO 2009

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL
POR: FELIPE RIQUELME NIKLITSCHK
FECHA: 23/01/2008
PROF. GUIA: SR. PATRICIO CONCA

DESARROLLO DE UNA SOLUCIÓN LOGÍSTICA PARA LA PROGRAMACIÓN DE OPERACIONES EN UNA COMPAÑÍA SIDERÚRGICA

El presente trabajo de título tuvo como objetivo el diseño, desarrollo y evaluación de una herramienta que permitiera apoyar la toma de decisiones respecto de la programación de operaciones en cada una de las dos plantas con que actualmente cuenta una compañía siderúrgica. Se buscó de esta forma encontrar la secuencia en que debieran ejecutarse los distintos trabajos mensuales, minimizando los tiempos de producción así como los tiempos de retrasos en las fechas de entrega.

La investigación se centró, por razones de tiempo, únicamente en el proceso de laminación de la compañía, que es aquel en donde se le da la forma final a los productos siderúrgicos mediante deformación termomecánica. Esta elección no fue al azar y se tomó considerando el hecho de que dicho proceso es hasta el día de hoy el principal “cuello de botella” y por lo tanto representaba las mayores oportunidades de ganancia.

Ahora bien, fue posible demostrar que el problema pertenece a la clase NP-Hard por lo que no se conocen algoritmos capaces de resolverlo en un tiempo polinomial. Como consecuencia, y dado que el tamaño de la instancia es relativamente grande, se hizo necesario incorporar enfoques heurísticos que permitieran obtener resultados suficientemente buenos en un tiempo de computación razonable.

Es así como se optó por un algoritmo de Búsqueda Tabú. La elección se basó principalmente en los buenos resultados reportados en la literatura para otros problemas de programación de operaciones (Lin y Ying, 2006; Gupta y Smith, 2007 y Valente y Alves, 2008).

Gran parte del desempeño de este tipo de heurística depende de dos elementos: la solución inicial y la metodología de generación de vecindades. Es por ello que la estrategia seguida consistió en evaluar un amplio espectro de las técnicas más utilizadas para tales fines, escogiendo finalmente aquella combinación que presentó un mejor desempeño.

Los resultados obtenidos muestran que la aplicación de la heurística propuesta a instancias reales permite obtener reducciones importantes en comparación a la situación actual: un 7% promedio en los tiempos de producción y disminuciones promedio del 35% en lo que se refiere a los tiempos de retraso. Por otra parte, se observa una dramática caída del 82% promedio en lo que al tiempo necesario para determinar la programación respecta.

Finalmente, cabe destacar que la investigación realizada sugiere también que aún hay espacio para futuras mejoras, por lo que se recomienda dar continuidad al estudio y en lo posible ampliarlo a otros procesos de la cadena productiva.

Agradecimientos

Con mucho cariño a mi familia, sin cuyo incondicional apoyo me hubiera sido imposible estar aquí escribiendo estas líneas y culminar esta importante etapa de mi vida. De no ser por el hombro que cada uno de ustedes colocó desinteresadamente para que yo pudiera descansar, probablemente el rumbo de esta historia sería completamente distinto. Simplemente gracias a todos, los amo.

También me gustaría aprovechar esta oportunidad para recordar a todos mis amigos que me acompañaron a lo largo de estos seis años. No puedo sino estar en deuda con ustedes por toda la ayuda y el ánimo que me dieron para seguir adelante.

Finalmente, agradezco a Dios cuya presencia y compañía he sentido a lo largo de toda mi vida. A Él le ofrezco todo el esfuerzo y dedicación colocados en este trabajo.

Índice de Contenidos

	Págs.
1. Introducción.....	1
2. Descripción del Problema.....	1
2.1. Breve Descripción del Proceso Productivo.....	2
2.1.1. Fusión en Acería.....	2
2.1.2. Colado del Acero.....	3
2.1.3. Laminación en Caliente.....	3
2.2. Planteamiento del Problema.....	4
2.3. Diagnóstico de la Situación Actual y Justificación del Proyecto.....	5
2.3.1. Análisis de Demanda.....	5
2.3.2. Análisis Cualitativo.....	6
2.3.3. Análisis Cuantitativo.....	7
3. Objetivos.....	9
3.1. Objetivo General.....	9
3.2. Objetivos Específicos.....	9
4. Alcances del Proyecto.....	10
5. Modelamiento del Problema.....	10
5.1. Modelo de Programación Planta N°1.....	12
5.1.1. Conjuntos de Índices.....	12
5.1.2. Parámetros.....	12
5.1.3. Variables de Decisión.....	12
5.1.4. Función Objetivo.....	13
5.1.5. Restricciones.....	13
5.2. Modelo de Programación Planta N°2.....	13
5.2.1. Conjuntos de Índices.....	14
5.2.2. Parámetros.....	14
5.2.3. Variables de Decisión.....	14
5.2.4. Función Objetivo.....	15
5.2.5. Restricciones.....	15
5.3. Modelo de Programación Planta N°2 Ajustado.....	16
5.3.1. Conjuntos de Índices.....	16
5.3.2. Parámetros.....	16
5.3.3. Variables de Decisión.....	17
5.3.4. Función Objetivo.....	17
5.3.5. Restricciones.....	17
5.4. Determinación de Parámetros.....	18
5.4.1. Tiempo de Procesamiento del Trabajo.....	18
5.4.2. Fecha de Entrega Ideal del Trabajo.....	19
5.4.3. Peso Asociado a la Importancia del Trabajo.....	20
5.4.4. Matriz de Tiempos de Cambio.....	20
5.4.5. Peso Medidas de Desempeño del Sistema.....	21
5.5. Complejidad del Problema Analizado.....	22
6. Enfoque de Solución Heurística.....	25
6.1. Selección del Tipo de Heurística a Aplicar.....	25
6.2. La Metodología de Búsqueda Tabú o Tabú Search (TS).....	27
6.2.1. Vecindad.....	27
6.2.2. Lista Tabú.....	27
6.2.3. Criterio de Aspiración.....	28
6.2.4. Criterio de Detención.....	29

6.2.5.	Algoritmo de Búsqueda Tabú.....	29
6.2.6.	Estrategias de Intensificación y Diversificación.....	30
6.3.	Alternativas Adoptadas para la Implementación de la Búsqueda Tabú.....	30
6.3.1.	Solución Inicial.....	31
6.3.1.1.	Solución Aleatoria (Random).....	31
6.3.1.2.	Weighted Shortest Processing Time (SWPT).....	31
6.3.1.3.	Earliest Due Date (EDD).....	31
6.3.1.4.	Apparent Tardiness Cost with Setups (ATCS).....	32
6.3.2.	Vecindad.....	32
6.3.2.1.	Adjacent Pairwise Interchange (API).....	33
6.3.2.2.	General Pairwise Interchange (GPI).....	34
6.3.2.3.	Top Priority to the Front (TPF).....	34
6.3.2.4.	General Top Priority to the Front (GTPF).....	35
6.3.2.5.	General Insertion Method (GIM).....	35
6.3.3.	Lista Tabú.....	35
6.3.4.	Criterio de Aspiración.....	37
6.3.5.	Criterio de Detención.....	37
6.3.6.	Estrategia de Diversificación.....	37
6.3.7.	Estrategia de Intensificación.....	38
6.3.8.	Consideraciones Adicionales.....	38
7.	Calibración de la Heurística.....	39
7.1.	Desempeño y Calibración Búsqueda Tabú.....	39
7.2.	Análisis de Tendencias de Comportamiento.....	41
7.3.	Calibración Tiempo de Ejecución.....	42
7.4.	Mejoras al Algoritmo Propuesto.....	43
7.4.1.	Estrategia de Diversificación.....	43
7.4.2.	Estrategia de Intensificación.....	45
7.5.	Resumen Parámetros Finales.....	45
8.	Resultados.....	46
8.1.	Caso Planta N°1.....	46
8.1.1.	Desempeño para Pequeñas Instancias.....	46
8.1.2.	Situación Actual <i>versus</i> Solución Propuesta.....	49
8.2.	Caso Planta N°2.....	50
8.2.1.	Desempeño para Pequeñas Instancias.....	50
8.2.2.	Situación Actual <i>versus</i> Solución Propuesta.....	52
8.3.	Valoración de las Mejoras Obtenidas.....	52
9.	Conclusiones.....	54
10.	Referencias Bibliográficas.....	56
11.	Anexos.....	58
	Anexo A. Normalización de Variables para la Función Objetivo.....	59
	Anexo B. Breve Análisis de Sensibilidad Planta N°1.....	61
	Anexo C. Breve Análisis de Sensibilidad Planta N°2.....	63
	Anexo D. Calibración Parámetros Apparent Tardiness Cost with Setup.....	64
	Anexo E. Estimación Beneficios Económicos por Reducción Tiempo Ciclo..	66
	Anexo F. Estimación Beneficios por Reducción Retrasos.....	68

Índice de Tablas

	Págs.
Tabla N°01. Evolución Ventas 2000-2007.....	5
Tabla N°02. Indicadores de Desempeño Sistema de Laminación Planta N°1.....	7
Tabla N°03. Indicadores de Desempeño Sistema de Laminación Planta N°2.....	8
Tabla N°04. Listado de Valores de alfa y beta a Considerar.....	21
Tabla N°05. Tiempo de CPU (seg) para Modelo Planta N°1.....	23
Tabla N°06. Tiempo de CPU (seg) para Modelo Planta N°2.....	24
Tabla N°07. Tamaño Vecindad (GPI).....	34
Tabla N°08. Parámetros Empleados Búsqueda Tabú (1).....	39
Tabla N°09. Desempeño Búsqueda Tabú Tiempo Procesamiento: 20 minutos.....	40
Tabla N°10. Parámetros Empleados Búsqueda Tabú (2).....	42
Tabla N°11. Desempeño Búsqueda Tabú Como Función Tiempo Resolución.....	42
Tabla N°12. Función Objetivo Bajo Estrategia Diversificación: Distintas Combinaciones.....	44
Tabla N°13. Función Objetivo Bajo Estrategia Intensificación.....	45
Tabla N°14. Parámetros Finales Calibrados Búsqueda Tabú.....	46
Tabla N°15. Búsqueda Tabú en Instancias Pequeñas: Planta N°1.....	47
Tabla N°16. Situación Actual vs Solución Propuesta: Planta N°1.....	49
Tabla N°17. Búsqueda Tabú en Instancias Pequeñas: Planta N°2.....	51
Tabla N°18. Situación Actual vs Solución Propuesta: Planta N°2.....	52
Tabla N°19. Función Objetivo frente a Cambios en Rendimiento Promedio por Familia: Planta N°1.....	61
Tabla N°20. Función Objetivo frente a Cambios en Tiempos de Setup por Familia: Planta N°1.....	62
Tabla N°21. Función Objetivo frente a Cambios en Tiempos de Setup por Familia: Planta N°2.....	63

Índice de Ilustraciones

	Págs.
Figura N°01. Flujo del Proceso Productivo.....	2
Figura N°02. Flujo Proceso de Programación de Operaciones.....	6
Figura N°03. Swap(1,5).....	33
Figura N°04. Insert(4,2).....	33
Figura N°05. Estructura de Información de Lista Tabú.....	36
Figura N°06. Comportamiento Función Objetivo en Relación al Tiempo de Ejecución	43
Figura N°07. Comportamiento Búsqueda Tabú: Planta N°1, 13 Trabajos.....	48

1. Introducción

La siderurgia es un proceso industrial de obtención de acero (aleación de hierro y carbono) a partir de hierro o de otro acero en desuso. El acero es el material básico del desarrollo y la industrialización, pues de su abastecimiento dependen los más variados sectores productivos, todos claves en una economía en desarrollo. Así, está presente en la construcción, la industria automotriz y de bienes durables, en la agricultura y hasta en la industria de alimentos, bebida y envasados.

En Chile, y pese al importante dinamismo observado en el mercado del acero durante los últimos años (incremento del 48% en las ventas totales del sector entre el 2000 y el 2007, según cifras publicadas por la SOFOFA), los efectos de la crisis financiera internacional ya comienzan a sentirse producto de una desaceleración de la construcción y la industria metalmeccánica (Orellana, 2008).

Como consecuencia de todo lo anterior, las compañías presentes en el sector siderúrgico nacional se han visto enfrentadas a la necesidad de mejorar sus procesos productivos a fin de poder satisfacer las exigencias del mercado y a la vez asegurar su competitividad estratégica futura. Y, de esta forma, se hace absolutamente necesario hacer el mejor uso posible de los recursos disponibles.

Es en este contexto en el que se enmarca el presente trabajo de título, en virtud del cual se desarrollará una herramienta formal -por medio de una solución de tipo heurística- que permita apoyar la toma de decisiones respecto de la programación de operaciones en una de las compañías del sector y la que a la fecha se realiza de forma completamente manual, con las consecuencias que esto último acarrea. Debido a la solicitud de mantener la confidencialidad respecto de la compañía en la que se realizará el estudio, se utilizará de aquí en adelante el nombre ficticio de Sider S.A. para referirse a ella.

El resto del informe se distribuye como sigue. Primero se realiza una descripción del problema y justificación de la importancia de ejecutar el proyecto, luego se presentan los objetivos del mismo. A continuación, se indica el modelamiento del problema, a nivel de un enfoque de programación matemática como asimismo del algoritmo de tipo heurístico basado en Tabu Search que fue desarrollado. Finalmente se presentan los resultados -incluyendo una comparación de la solución propuesta en relación a la situación actual - y se concluye a partir de ellos.

2. Descripción del Problema

Como se verá en las secciones siguientes, la forma en que actualmente se planifica la programación mensual de actividades en el área de laminación parece ser inadecuada dada la gran cantidad de productos a secuenciar. Como consecuencia, los tiempos de producción se incrementan, al igual que los retrasos en las fechas de entrega.

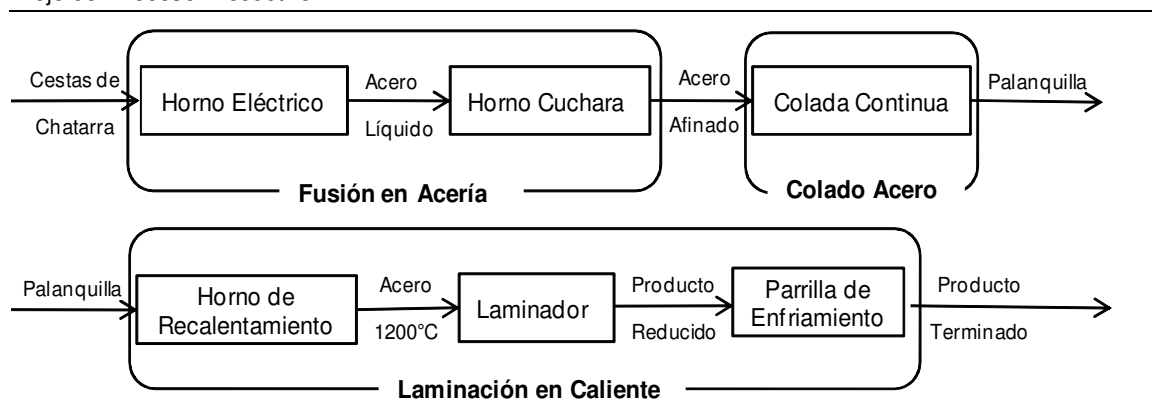
2.1. Breve Descripción del Proceso Productivo

El proceso de fabricación de acero en Sider se inicia con la selección, procesamiento y corte de trozos de acero en desuso (chatarra), los que constituyen la materia prima básica.

El proceso productivo puede resumirse en tres etapas principales: fusión, colado y laminación en caliente; las que son presentadas esquemáticamente en la Figura N°01 y detalladas a continuación.

Figura N°01

Flujo del Proceso Productivo



Fuente: Elaboración Propia

2.1.1. Fusión en Acería

La materia prima se carga en cestas, en proporciones adecuadas para satisfacer las especificaciones del proceso, las que son trasladadas a la Acería para alimentar un horno de arco eléctrico donde el material se funde.

Una vez terminado el proceso de fusión, en donde toda la carga pasa del estado sólido al estado líquido, el acero es trasladado a un Horno de Cuchara, donde se ejecutará la etapa de afino y se procederá a tomar muestras para realizar el análisis de espectrometría, con el propósito de conocer su composición química.

Luego de conocido el informe sobre la composición química, se realizan las correcciones necesarias mediante el proceso de afino, lo que permite obtener la composición y purezas deseadas (calidades buscadas).

Cuando el acero líquido cumple con las especificaciones requeridas, tanto de composición química como de temperatura, es trasladado en la cuchara hasta el equipo de colada continua.

2.1.2. Colado del Acero

Obtenido el acero en su estado líquido, éste debe solidificarse en una forma conveniente para su utilización posterior en los trenes de laminación, lo cual se hace mediante un equipo de colada continua, en el que se transforma el acero líquido en un producto semiterminado denominado palanquilla (barras macizas de 130 mm x 130 mm de sección).

Con este fin, el acero líquido que se encuentra en la cuchara de colada es transferido a un distribuidor. Desde aquí, el acero cae dentro de lingoteras de cobre sin fondo, de doble pared y refrigeradas por agua, donde se inicia la solidificación del acero.

Después de dejar las lingoteras, el acero sólido, es tomado por juegos de rodillos refrigerados con chorros de agua a alta presión, solidificándose completamente y convirtiéndose finalmente en palanquilla.

2.1.3. Laminación en Caliente

La laminación en caliente, es un proceso de transformación termomecánico, en donde se da la forma final a los productos siderúrgicos. El proceso puede describirse como sigue: en la planta de laminación, las palanquillas son seleccionadas según la calidad del acero del producto final y son cargadas a un horno de recalentamiento, donde alcanzan una temperatura uniforme de 1.200°C, lo que permitirá su deformación plástica durante el proceso.

En este proceso, la palanquilla es tratada mecánicamente, haciéndola pasar sucesivamente por los rodillos de los trenes de laminación, los cuales van reduciendo su sección original. De esta forma, se lleva la sección transversal de la palanquilla cada vez más próxima a la forma y diámetro final buscado.

En este punto es necesario resaltar que, en la actualidad, Sider cuenta con dos sistemas de laminación: uno en cada una de las dos plantas productivas que posee. Aunque la operación de ambos laminadores es muy similar, existen algunas diferencias -las cuales serán abordadas en detalle más adelante- que obligan a hacer una distinción entre ambos. De esta forma, y en lo que sigue, se hará referencia a la Planta N°1 y la Planta N°2 para señalar cada laminador respectivamente.

Ahora bien, las barras que han acabado el proceso de laminación se depositan en una gran placa o parrilla de enfriamiento. Desde ahí son trasladadas a las líneas de corte a medida y empaquetado para posteriormente pasar a la zona de almacenamiento y quedar listas para su despacho.

2.2. Planteamiento del Problema

En la actualidad, el “cuello de botella” del proceso productivo se encuentra en la etapa conocida como laminación. Esto último se explica porque aunque el rendimiento promedio de la Acería es de 68 toneladas por hora, el rendimiento de los laminadores es considerablemente menor: en la Planta N°1 alcanza apenas 15 toneladas por hora mientras que en la Planta N°2 llega a las 32 toneladas por hora.

De esta forma, y desde un punto de vista exclusivamente productivo, es necesario poner especial atención en el sistema de laminación y en cómo hacer mejor uso de éste, a fin de reducir el atochamiento. Por otra parte, y asociado a este mismo punto, existe una necesidad de mejorar la calidad de servicio, medida como el porcentaje de cumplimiento de pedidos en las fechas de entrega comprometidas, el que en la actualidad es relativamente bajo (como se verá más adelante).

En este contexto, la principal fuente potencial de mejoras puede encontrarse a nivel del sistema de programación de operaciones en los laminadores; es decir, la manera en que se deben procesar los distintos trabajos y actividades que se presentan mensualmente, debido a que en la actualidad esta programación se hace de forma completamente manual y sin ninguna herramienta o metodología formal.

Luego, si se considera que el *mix*¹ es del orden de los 100 a 200 productos, dependiendo de qué planta se trate, entonces en el mejor de los casos se generan del orden de 10^{157} secuencias distintas. Esto último sugiere que la programación manual se traduce en ineficiencias que originan el incumplimiento de los plazos comprometidos así como un mayor tiempo total de producción, lo que a la larga se traduce en mayores costos. A esto debe sumarse el hecho de que el encargado responsable (que es un recurso valioso) emplea del orden de un día completo de su tiempo en confeccionar el calendario de programación mensual.

El problema anterior plantea un desafío para la empresa, siempre preocupada del mejoramiento continuo de los procesos, a fin de alcanzar una mayor competitividad y eficiencia productiva como así mismo un mejoramiento de los estándares de calidad de servicio. De esta forma, parece atractiva la opción de hacer uso de metodologías formales a fin de encontrar la “mejor” alternativa de programación mensual.

La exitosa experiencia de casos en que se han aplicado estas metodologías, como por ejemplo en la industria electrónica (Gupta y Sivakumar, 2005), del aluminio (Gravel, Price y Gagne, 2000) o las aerolíneas (Butchers y Ryan, 2001), muestran el enorme potencial de las mismas y su contribución a la mejora de desempeño de las organizaciones e industrias en que son introducidas.

¹ Conjunto o combinación de productos.

2.3. Diagnóstico de la Situación Actual y Justificación del Proyecto

A fin de asegurar la relevancia de ejecutar el proyecto, se hace necesario generar un diagnóstico más acabado respecto de la operación actual del sistema de laminación de la Planta N°1 y Planta N°2.

2.3.1. Análisis de Demanda

El pronóstico de demanda es un tema complejo debido a la alta variabilidad que en general se observa en la serie de demanda (ventas). Sin embargo, se hace necesario contar con elementos de pronóstico que permitan determinar un horizonte de tiempo dentro del cual el modelo de programación de operaciones que se implementará será válido.

La Tabla N°01, muestra un resumen del comportamiento de ventas durante los últimos siete años (período 2000-2007). Cabe destacar que, para mantener la confidencialidad de la información, todos los datos han sido amplificados por un factor de escala común.

Tabla N°01

Evolución Ventas 2000-2007

	2000	2001	2002	2003	2004	2005	2006	2007
Ventas (Ton)*	115462	136841	159369	164879	227089	225463	252976	268427
Tasa Crecimiento (%)	-	18,5	16,5	3,5	37,7	-0,7	12,2	6,1

* : Las ventas han sido ajustadas por un factor de escala para mantener la confidencialidad de la información

Fuente: ERP de la compañía

Como se observa en la tabla anterior, el volumen de ventas de Sider se ha incrementado fuertemente durante el período 2000-2007 (132,48%). Esto último responde principalmente a la puesta en marcha, a contar del finales del año 1999, de una nueva planta productiva (Planta N°2). Luego, el aumento de las ventas es consecuencia directa del aumento de la capacidad productiva.

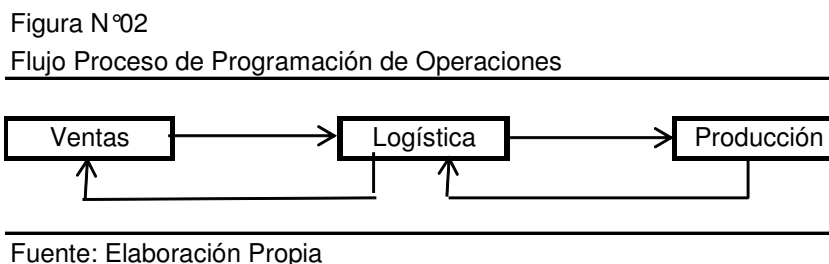
En este contexto, la Tabla N°01 también permite concluir que la tasa de crecimiento de ventas presenta una tendencia decreciente, especialmente durante los últimos dos años, lo cual parece ser un indicador de que recién se está comenzando a alcanzar el estado de régimen del sistema. Si adicionalmente se considera el escenario económico actual y las proyecciones futuras en cuanto a una reducción o al menos un estancamiento de la producción para los próximos años², entonces parece razonable concluir que el horizonte de tiempo dentro del cual el modelo de programación de operaciones a implementar será válido es de al menos tres a cinco años.

² DIRECMIN. Crisis Financiera Obliga a Vale a Reducir Producción y Gastos. 2008. [en línea] <http://www.direcmin.com/Noticias/Whats-Up-Empresas/Crisis-Financiera-Obliga-a-Vale-a-Reducir-Produccion-y-Gastos.html> [consulta: 27 noviembre 2008].

2.3.2. Análisis Cualitativo

Como ya ha sido mencionado con anterioridad, la programación de operaciones de las actividades de laminación en Sider es realizada de forma completamente manual por una persona encargada, que además es un recurso valioso.

El flujo del proceso de programación de operaciones puede ser descrito como se muestra en la Figura N°02.



En primer lugar, el Departamento de Ventas envía los requerimientos mensuales al Departamento de Logística. Ahí, el encargado responsable genera manualmente un programa en base a los requerimientos recibidos. Y una vez que la programación se encuentra lista, ésta es enviada al Área de Producción para que sea ejecutada.

Pese a que el proceso suele concluir en dicho punto, existen algunas ocasiones en las cuales el Área de Producción sugiere algunos cambios en la programación, con base exclusivamente en criterios de eficiencia productiva. Es por ello que se solicita incrementar los volúmenes de producción o alterar la secuencia original a fin de disminuir los tiempos de cambio entre un producto y el siguiente.

Cuando lo anterior acontece, el encargado de Logística hace llegar las solicitudes de cambio al Departamento de Ventas. Luego de una reunión con el encargado de Logística, se toma la decisión final respecto de qué cambios aplican y cuáles no. Sin embargo, las fechas de entrega solicitadas no son modificables, por lo que la producción debe tratar de hacer lo posible por respetarlas.

En base a lo anterior se puede concluir que al momento de determinar la programación mensual de actividades, se suele poner énfasis tanto en la eficiencia productiva (medida en términos de toneladas producidas por hora) como en la calidad de servicio (medida por los tiempos de retraso en las fechas de entrega).

2.3.3. Análisis Cuantitativo

La Tabla N°02 y la Tabla N°03 presentan un resumen con los principales indicadores de desempeño de cada planta respectivamente.

Tabla N°02

Indicadores de Desempeño Sistema de Laminación Planta N°1

Indicador	Valor
Determinación Programación	4,5 hrs.
Tiempo Total Producción	24 días
Trabajos con Retraso	15,00%
Trabajos con Retraso (Ponderado)	20,00%
Retraso Promedio por Trabajo	1 día
Retraso Promedio por Trabajo (Ponderado)	1 día

Fuente: Elaboración Propia

En donde la primera medida corresponde al tiempo aproximado empleado por la persona encargada para determinar la secuencia de producción. El segundo indicador se refiere al tiempo de ciclo sobre la base de los requerimientos recibidos al inicio del mes. La razón de que este valor no sea de 30 días es que en la práctica existen detenciones programadas debido al ahorro energético como asimismo trabajos de último momento, todos los cuales han sido descontados a fin de contrastar adecuadamente la situación actual (programación al inicio del mes) con la solución que se propondrá a lo largo de la presente investigación.

Los trabajos con retraso indican el porcentaje de trabajos que finalizan con posterioridad a la fecha ideal de entrega, independientemente de la cantidad de días de atraso. El siguiente indicador es exactamente igual al anterior, salvo que ha sido ajustado por el peso relativo de cada producto dentro del *mix* total. Esto último se debe a la necesidad de considerar la existencia de ciertos productos que son más relevantes que otros (como se verá en el capítulo 5).

Finalmente, el retraso promedio por trabajo busca cuantificar la amplitud promedio del tiempo de retraso por trabajo. Y el retraso promedio por trabajo ponderado busca incorporar el efecto que genera la importancia relativa de cada producto.

Por otra parte, en este punto es necesario hacer notar que, aunque la compañía cuenta actualmente con un sistema ERP³, éste registra información detallada respecto de las operaciones únicamente en Planta N°2, mientras que para la Planta N°1 sólo es posible obtener información agregada en base a familias de productos.

³ Enterprise Resource Planing. Sistema de administración de la información de una organización, orientado a la planificación de empresas. Suele llevar registro de información relevante de las distintas actividades productivas y áreas de negocio de una organización.

Luego, en este caso, y dada la inexistencia de información precisa respecto de la planificación de operaciones, se han considerado valores estimados en función del criterio y experiencia del actual encargado de realizar la programación mensual de actividades (juicio experto).

Tabla N°03
Indicadores de Desempeño Sistema de Laminación Planta N°2

Indicador	Valor
Determinación Programación	3 hrs.
Tiempo Total Producción	28,62 días
Trabajos con Retraso	43,84%
Trabajos con Retraso (Ponderado)	57,97%
Retraso Promedio por Trabajo	5,38 días
Retraso Promedio por Trabajo (Ponderado)	6,69 días

Fuente: Elaboración Propia

Para el caso de la Planta N°2, es posible contar -haciendo uso del sistema ERP de que dispone la compañía- con información variada de la programación. Por ello, los indicadores anteriores han sido calculados como los valores promedio de los últimos tres meses sobre la base de la programación observada.

Ahora bien, existen dos fuentes importantes que afectan el tiempo total de producción: el tiempo de procesamiento (exclusivo de cada trabajo) y el tiempo de cambio o de *setup* (depende del trabajo actual y de aquel inmediatamente anterior). El tiempo de procesamiento es relativamente fijo y depende del rendimiento promedio del laminador para cada producto específico. Por su parte, los tiempos de *setup* pueden aumentar o disminuir dependiendo de la programación que se aplique (secuencia). Y es precisamente esta fuente la que representa la principal oportunidad de mejora ya que con un *mix* de 100 productos -como en el caso de la Planta N°2- se generan del orden de 10^{157} combinaciones o secuencias distintas, por lo que la toma de decisiones de forma manual se vuelve sumamente compleja y probablemente ineficiente. Cabe agregar que en el caso de la Planta N°1 existen del orden de 10^{374} secuencias de productos distintas.

Por otra parte, se observa una gran cantidad de retrasos con respecto a las fechas de entrega comprometidas. Y, de existir la posibilidad de reducir los tiempos de producción, haciendo un mejor uso de los recursos al generar una secuencia eficiente, entonces parece razonable pensar que al tener los trabajos más rápido también se reduce el tiempo promedio de retraso.

De esta forma se justifica, a lo menos, el estudio del problema y la exploración de distintas alternativas que permitan reducir los indicadores propuestos, mediante el desarrollo de un sistema formal de apoyo a la toma de decisiones.

3. Objetivos

3.1. Objetivo General

Se persigue generar una herramienta formal de apoyo a la programación mensual de operaciones para las actividades de laminación en las dos plantas productivas con que actualmente cuenta una compañía siderúrgica, que permita decidir la secuencia en que se ejecutarán los trabajos, a fin de minimizar los tiempos de producción así como los retrasos en los plazos y fechas de entrega comprometidos.

3.2. Objetivos Específicos

El objetivo anterior se descompone en los siguientes objetivos específicos:

- a. Reducir retrasos en fechas de entrega. Parte importante del desarrollo del trabajo de memoria pasa por determinar una programación de actividades que reduzca la cantidad de trabajos con retraso o, a lo menos, la magnitud de dichos retrasos si es que éstos se producen.
- b. Reducir tiempos de producción. Otro elemento esencial del presente proyecto consiste en encontrar una secuencia que se traduzca en una disminución de los tiempos de producción, es decir, que el instante en que cada trabajo es completado se vea reducido; especialmente aquellos asociados a los productos más importantes.
- c. Reducir el tiempo empleado en determinar la programación. Se espera desarrollar un modelo que entregue soluciones de calidad aceptable en un tiempo de proceso razonable, no superior a un par de horas. De esta forma, se busca contar con una herramienta que sea capaz de ejecutarse en cualquier instante. Esto último entrega la flexibilidad de poder reprogramar la producción frente a cambios o eventualidades que pudieran suceder en el transcurso del mes.
- d. Implantación de metodologías formales para determinar la programación. Se busca un cambio organizacional a nivel de la metodología para determinar la programación mensual que permita incorporar herramientas para la evaluación de distintos escenarios y que justifique las decisiones tomadas. Para ello, se elaborará un prototipo capaz de recibir las entradas del modelo (*input*) y generar el calendario de programación mensual lo más cercano al óptimo posible (*output*).

4. Alcances del Proyecto

Dada la complejidad del problema y también por razones de tiempo, el trabajo se limitará únicamente a la programación de operaciones de los laminadores de acero en ambas plantas.

Esto último se sustenta en que el principal “cuello de botella” del proceso productivo se encuentra hoy en día en lo que respecta precisamente a la actividad de laminación y que la principal oportunidad de mejora se encuentra a nivel de la planificación de la programación (tal y como ya se detalló en el capítulo 2.2). Por lo tanto, para efectos del tema de memoria, no se considerará el resto de los procesos productivos, tales como la acería o la colada.

Por otra parte, para efectos prácticos, no se realizará un análisis de las políticas de inventario de productos que se llevan en la planta y sólo se considerará como un *input* dado del sistema. Del mismo modo, la demanda de ambas plantas estará basada en el pronóstico de ventas elaborado por el departamento del mismo nombre por lo que para todo efecto se considerará dada (determinística). Esto último se basa fundamentalmente en la forma empleada por la compañía para determinar el pronóstico: se consulta a la lista de clientes por los requerimientos mensuales de los distintos productos. En general, las desviaciones de este pronóstico no son significativas -salvo pedidos urgentes de último momento- por lo que se considera una fuente de información confiable.

La elaboración de la solución contempla su implementación sólo hasta una etapa funcional, es decir, la construcción de un *software* prototipo sencillo que lea las entradas del modelo y entregue una solución, pero sin tener en cuenta requerimientos respecto de la interfaz gráfica de usuario, diseño personalizado, etc. Estos elementos pueden ser fácilmente incorporados por un experto en programación de *software* una vez se tenga el modelo base.

Finalmente, elementos como la capacidad física y la fuerza de trabajo actual de las plantas se considerarán fijos e invariantes durante el período de planificación (equivalente a 1 mes).

5. Modelamiento del Problema

El problema general planteado en el capítulo 2.2 y contextualizado en el capítulo 2.3 puede enunciarse como sigue. Se debe procesar un conjunto de n -trabajos (o productos) en una única máquina (el laminador), la cual sólo puede procesar un trabajo a la vez. Asociado a cada trabajo j existe un tiempo de procesamiento, una fecha de entrega para la cual el trabajo debería estar idealmente finalizado y un tiempo de *setup*, así como también una importancia relativa en relación al resto de los trabajos (definida en términos de su promedio mensual de ventas).

Una consideración adicional es que no es posible interrumpir un trabajo para reiniciarlo con posterioridad. Esto último se debe a que el proceso de laminación debe ser realizado en caliente y por lo tanto, si éste es interrumpido, no sólo se perjudican las propiedades plásticas del material sino que también se arriesga la composición química del mismo (calidad).

El objetivo del modelo es determinar la secuencia en que deben ser ejecutados los trabajos de modo de minimizar los tiempos de producción así como los retrasos en los plazos y fechas de entrega. De esta forma, se definen dos medidas de desempeño del sistema:

- a. Tiempo de Producción C_j , asociada al trabajo j . Esta medida representa el instante de tiempo en que el trabajo j es completado.
- b. Tiempo de Retaso T_j , asociada al trabajo j . Representa el retraso del trabajo j en relación a su fecha de entrega (si es que existe) y se define como $T_j = \max\{C_j - d_j, 0\}$.

Cabe destacar que la primera medida puede ser considerada como un indicador de la eficiencia productiva; pues mientras menores sean los tiempos de producción de cada uno de los trabajos a procesar, mejor uso de los recursos se estará haciendo. Por otra parte, la segunda medida presentada puede ser considerada como un indicador de la calidad de servicio, es decir, menores retrasos se traducen en un mayor cumplimiento de cara al cliente, lo que con el tiempo mejora la confianza y credibilidad.

De esta forma, resulta claro que uno de los principales objetivos del modelo a desarrollar consiste en encontrar un equilibrio adecuado entre ambas medidas de desempeño del sistema. En consecuencia, tanto el tiempo de producción (C_j) como el tiempo de retraso (T_j) serán normalizados a valores en el intervalo $[0,1]$ mediante el procedimiento descrito en el Anexo A. Con ello, sólo basta asignar a cada una de las medidas de desempeño una proporción, definidas por los coeficientes α y β , de modo tal que $\alpha + \beta = 1$.

Ahora bien, en la literatura este tipo de problemática de secuenciamiento de trabajos con tiempos de *setup* es conocido como “*Single Machine Scheduling Problema with Sequence-Dependent Setup Times and Due Dates*” o “Programación de Operaciones un una Máquina con Tiempos de Preparación Dependientes de la Secuencia y Fechas de Entrega”. La investigación respecto de este problema específico es bastante reducida, especialmente porque la mayoría de los autores suele trabajar con versiones idealizadas en las que los tiempos de preparación se suponen nulos, no se consideran las fechas de entrega o ambas. Sin embargo, durante los últimos años la preocupación por incorporar estos elementos más sofisticados ha ido en aumento, especialmente por la necesidad de las grandes industrias de contar con un mayor grado de realismo para mejorar sus procesos productivos. Destacan en este sentido los trabajos de Armentano y Mazzini (2000); Gupta y Smith (2005); Eren y Güner (2006); Lin y Ying (2007) y Valente y Alves (2007).

Finalmente, es necesario recordar en este punto que el problema bajo estudio abarca la confección de una solución para la programación de operaciones en cada una de las dos plantas de producción con las que actualmente cuenta la compañía. Y, aunque el problema general es el enunciado anteriormente, existen pequeñas diferencias en las características de operación de cada planta que obligan a plantear dos modelos de programación matemática distintos, los que son presentados a continuación.

5.1. Modelo de Programación Planta N°1

En este caso, el problema es exactamente idéntico al planteado con anterioridad. No se consideran por tanto, restricciones adicionales especiales de ningún tipo. Luego, es posible desarrollar un modelo de programación lineal entera mixta, con base en aquel propuesto por Eren y Güner (2006), el que toma la forma que se presenta a continuación.

5.1.1. Conjuntos de Índices

$T = \{1, \dots, n\}$ = Conjunto de trabajos a secuenciar

5.1.2. Parámetros

p_j = Tiempo de procesamiento del trabajo j , donde $j \in T$.

d_j = Fecha de entrega ideal del trabajo j , donde $j \in T$.

w_j = Peso asociado a la importancia del trabajo j , donde $j \in T$.

$s_{j,k}$ = Tiempo de *setup* del trabajo k si es precedido por el trabajo j en la secuencia, donde $j, k \in T$.

$M = \sum_{j \in T} p_j + \sum_{j \in T} \sum_{k \in T} s_{j,k}$ = Número real suficientemente grande.

α = Peso para medida de desempeño del tiempo total de producción ($0 \leq \alpha \leq 1$).

β = Peso para medida de desempeño del tiempo total de retraso ($\alpha + \beta = 1$).

5.1.3. Variables de Decisión

$$x_{j,k} = \begin{cases} 1 & \text{si el trabajo } j \text{ es asignado antes que el trabajo } k \text{ en la secuencia} \\ 0 & \text{en caso contrario} \end{cases}$$

C_j = Instante en que se completa la producción del trabajo j .

$T_j = \max\{C_j - d_j, 0\}$ = Tiempo de retraso del trabajo j .

5.1.4. Función Objetivo

$$\text{Min } \alpha \sum_{j \in T} w_j C_j^* + \beta \sum_{j \in T} w_j T_j^*$$

Donde el tiempo de producción (C_j) y el tiempo de retraso (T_j) han sido normalizados a valores en el intervalo $[0,1]$ para asegurar la consistencia de la proporción asignada a cada una de las medidas de desempeño, definidas por los coeficientes α y β .

5.1.5. Restricciones

a. Tiempo de completitud del trabajo j .

$$C_j \geq p_j \quad \forall j \in T \quad (1)$$

b. Solo un trabajo puede ser procesado al mismo tiempo.

$$C_j - C_k + (M * x_{j,k}) \geq s_{k,j} + p_j \quad \forall j \in T \mid j = 1, \dots, n-1 \quad \forall k \in T \mid k = 2, \dots, n \quad k > j \quad (2)$$

$$C_k - C_j + (M * (1 - x_{j,k})) \geq s_{j,k} + p_k \quad \forall j \in T \mid j = 1, \dots, n-1 \quad \forall k \in T \mid k = 2, \dots, n \quad k > j \quad (3)$$

c. Tiempo de retraso del trabajo j .

$$T_j \geq C_j - d_j \quad \forall j \in T \quad (4)$$

d. Naturaleza de las variables.

$$x_{j,k} \in \{0,1\} \quad \forall j \in T \mid j = 1, \dots, n-1 \quad \forall k \in T \mid k = 2, \dots, n \quad k > j \quad (5)$$

$$C_j, T_j \in \mathfrak{R}^+ \quad (6)$$

Donde la restricción (1) especifica que el tiempo de producción asociado al trabajo j debe ser mayor o a lo sumo igual que su tiempo de procesamiento. Las restricciones (2) y (3) aseguran que sólo un trabajo pueda ser procesado al mismo tiempo. La restricción (4) especifica que el tiempo de retraso del trabajo j es mayor o igual que la diferencia entre el tiempo de producción y la fecha de entrega de ese trabajo. Finalmente, las restricciones (5) y (6) indican la naturaleza de las variables.

Es fácil ver que el modelo propuesto tiene $(n^2 - n)/2$ variables binarias, $(n + 1)$ variables continuas y $(n^2 + n)$ restricciones; donde n es el número total de trabajos a secuenciar.

5.2. Modelo de Programación Planta N°2

La operación de esta planta considera, además de los elementos presentados para la Planta N°1, otras restricciones especiales. Éstas son enumeradas a continuación.

- Existe un conjunto de trabajos (o productos) que tienen una fecha máxima de completitud, adicionalmente de la fecha ideal, para la cual deben estar finalizados obligatoriamente.
- Existe un grupo de trabajos que sólo pueden realizarse inmediatamente a continuación de otro que pertenezca a una determinada familia de productos.

- c. Se cuenta con un grupo de trabajos que deben estar distanciados al menos una semana en su fecha de completitud.

Tomando en cuenta todas estas consideraciones, es posible confeccionar un modelo de programación matemática como el que sigue. Cabe destacar que se ha utilizado como base aquel propuesto por Armentano y Mazzini (2000).

5.2.1. Conjuntos de Índices

$T = \{1, \dots, n\}$ = Conjunto de trabajos a secuenciar.

$T_0 = \{0\} \cup T$ = Conjunto de trabajos a secuenciar, donde se ha incluido un trabajo o nodo ficticio para representar el primer y último trabajo en la secuencia.

$D \subseteq T$ = Conjunto de trabajos que exigen una fecha máxima de completitud.

$B \subseteq T$ = Conjunto de trabajos que pertenecen al grupo B.

$A \subseteq T$ = Conjunto de trabajos cuya campaña anterior debe pertenecer al grupo B, tal que $|A| \leq |B|$.

$E \subseteq T$ = Conjunto de trabajos que deben estar distanciados al menos una semana en su completitud.

5.2.2. Parámetros

p_j = Tiempo de procesamiento del trabajo j , donde $j \in T$.

d_j = Fecha de entrega ideal del trabajo j , donde $j \in T$.

\bar{d}_j = Fecha de entrega máxima para la cual el trabajo j debe estar finalizado, donde $j \in D$.

w_j = Peso asociado a la importancia del trabajo j , donde $j \in T$.

$s_{j,k}$ = Tiempo de *setup* del trabajo k si es precedido por el trabajo j en la secuencia, donde $j, k \in T_0$.

$M = \sum_{j \in T_0} p_j + \sum_{j \in T_0} \sum_{k \in T_0} s_{j,k}$ = Número real suficientemente grande.

α = Peso para medida de desempeño del tiempo total de producción ($0 \leq \alpha \leq 1$).

β = Peso para medida de desempeño del tiempo total de retraso ($\alpha + \beta = 1$).

5.2.3. Variables de Decisión

$$x_{j,k} = \begin{cases} 1 & \text{si el trabajo } k \text{ sigue inmediatamente después que el trabajo } j \\ 0 & \text{en caso contrario} \end{cases}$$

C_j = Instante en que se completa la producción del trabajo j .

$T_j = \max\{C_j - d_j, 0\}$ = Tiempo de retraso del trabajo j .

5.2.4. Función Objetivo

$$\text{Min } \alpha \sum_{j \in T} w_j C_j + \beta \sum_{j \in T} w_j T_j$$

Donde el tiempo de producción (C_j) y el tiempo de retraso (T_j) han sido normalizados a valores en el intervalo $[0,1]$ para asegurar la consistencia de la proporción asignada a cada una de las medidas de desempeño, definidas por los coeficientes α y β .

5.2.5. Restricciones

a. Relaciones de sucesión.

$$\sum_{j \in T_0, j \neq k} x_{j,k} = 1 \quad \forall k \in T \quad (1)$$

$$\sum_{k \in T, j \neq k} x_{j,k} = 1 \quad \forall j \in T_0 \quad (2)$$

b. Solo un trabajo puede ser procesado al mismo tiempo.

$$C_k \geq C_j + (s_{j,k} + p_k)x_{j,k} - (1 - x_{j,k})M \quad \forall j \in T_0 \quad \forall k \in T \quad (3)$$

c. Tiempo de retraso del trabajo j .

$$T_j \geq C_j - d_j \quad \forall j \in T \quad (4)$$

d. Cumplimiento fechas límite.

$$C_j \leq \bar{d}_j \quad \forall j \in D \quad (5)$$

e. Campaña anterior.

$$\sum_{j \in B} x_{j,k} = 1 \quad \forall k \in A \quad (6)$$

f. Separación de al menos una semana.

$$|C_k - C_j| \geq 10080 \quad \forall j, k \in E \quad (7)$$

g. Naturaleza de las variables.

$$x_{j,k} \in \{0,1\} \quad \forall j, k \in T_0 \quad (8)$$

$$C_j, T_j \in \mathfrak{R}^+ \quad (9)$$

Donde las restricciones (1) y (2) aseguran que cada trabajo tiene sólo un predecesor y un sucesor. La restricción (3) especifica que sólo un trabajo puede ser procesado al mismo tiempo. La restricción (4) indica que el tiempo de retraso del trabajo j es mayor o igual que la diferencia entre el tiempo de producción y la fecha de entrega de ese trabajo. La restricción (5) asegura que se cumpla la fecha máxima de completitud del trabajo $j \in D$. La restricción (6) establece que la campaña anterior a la de los trabajos $k \in A$ debe pertenecer al grupo B . La restricción (7) asegura que los trabajos $j \in E$ deben estar distanciados al menos una semana en su tiempo de completitud. Finalmente, las restricciones (8) y (9) indican la naturaleza de las variables.

Por otra parte, el modelo propuesto tiene $n(n+1)$ variables binarias, $(n+1)$ variables continuas y, al menos, $(n^2 + 4n + 1)$ restricciones; donde n es el número total de trabajos a secuenciar.

Respecto de lo anterior, y si se lo compara con el modelo de la Planta N°1, se puede ver que la cantidad de variables binarias se ha incrementado. Esto último se explica porque las particularidades de operación de la Planta N°2 (restricciones) exigen seguir un rastro más preciso de cada trabajo y aquel inmediatamente siguiente.

Finalmente, se hace necesario notar que la existencia de la restricción (7) da origen a un modelo de tipo no lineal. Sin embargo, es posible generar una versión ajustada de este modelo a fin de contar con uno de tipo lineal, comparable al desarrollado para la Planta N°1, el cual es presentado a continuación.

5.3. Modelo de Programación Planta N°2 Ajustado

En este caso, las consideraciones y restricciones son exactamente idénticas a las expuestas en la sección anterior. Sin embargo, se busca generar una versión lineal de tipo entero mixto para el problema de la Planta N°2. Para cumplir con este objetivo, se hace necesaria la incorporación de variables adicionales, tal y como se muestra en lo que sigue.

5.3.1. Conjuntos de Índices

$T = \{1, \dots, n\}$ = Conjunto de trabajos a secuenciar.

$T_0 = \{0\} \cup T$ = Conjunto de trabajos a secuenciar, donde se ha incluido un trabajo o nodo ficticio para representar el primer y último trabajo en la secuencia.

$D \subseteq T$ = Conjunto de trabajos que exigen una fecha máxima de completitud.

$B \subseteq T$ = Conjunto de trabajos que pertenecen al grupo B.

$A \subseteq T$ = Conjunto de trabajos cuya campaña anterior debe pertenecer al grupo B, tal que $|A| \leq |B|$.

$E \subseteq T$ = Conjunto de trabajos que deben estar distanciados al menos una semana en su completitud.

5.3.2. Parámetros

p_j = Tiempo de procesamiento del trabajo j , donde $j \in T$.

d_j = Fecha de entrega ideal del trabajo j , donde $j \in T$.

\bar{d}_j = Fecha de entrega máxima para la cual el trabajo j debe estar finalizado, donde $j \in D$.

w_j = Peso asociado a la importancia del trabajo j , donde $j \in T$.

$s_{j,k}$ = Tiempo de *setup* del trabajo k si es precedido por el trabajo j en la secuencia, donde $j, k \in T_0$.

$M = \sum_{j \in T_0} p_j + \sum_{j \in T_0} \sum_{k \in T_0} s_{j,k}$ = Número real suficientemente grande.

α = Peso para medida de desempeño del tiempo total de producción ($0 \leq \alpha \leq 1$).

β = Peso para medida de desempeño del tiempo total de retraso ($\alpha + \beta = 1$).

5.3.3. Variables de Decisión

$$x_{j,k} = \begin{cases} 1 & \text{si el trabajo } k \text{ sigue inmediatamente después que el trabajo } j \\ 0 & \text{en caso contrario} \end{cases}$$

$$y_{j,k} = \begin{cases} 1 & \text{si el trabajo } j \text{ es asignado antes que el trabajo } k \text{ en la secuencia} \\ 0 & \text{en caso contrario} \end{cases}$$

C_j = Instante en que se completa la producción del trabajo j .

$T_j = \max\{C_j - d_j, 0\}$ = Tiempo de retraso del trabajo j .

5.3.4. Función Objetivo

$$\text{Min } \alpha \sum_{j \in T} w_j C_j + \beta \sum_{j \in T} w_j T_j$$

Donde el tiempo de producción (C_j) y el tiempo de retraso (T_j) han sido normalizados a valores en el intervalo $[0,1]$ para asegurar la consistencia de la proporción asignada a cada una de las medidas de desempeño, definidas por los coeficientes α y β .

5.3.5. Restricciones

a. Relaciones de sucesión.

$$\sum_{j \in T_0, j \neq k} x_{j,k} = 1 \quad \forall k \in T \quad (1)$$

$$\sum_{k \in T, j \neq k} x_{j,k} = 1 \quad \forall j \in T_0 \quad (2)$$

b. Solo un trabajo puede ser procesado al mismo tiempo.

$$C_k \geq C_j + (s_{j,k} + p_k)x_{j,k} - (1 - x_{j,k})M \quad \forall j \in T_0 \quad \forall k \in T \quad (3)$$

c. Tiempo de retraso del trabajo j .

$$T_j \geq C_j - d_j \quad \forall j \in T \quad (4)$$

d. Cumplimiento fechas límite.

$$C_j \leq \bar{d}_j \quad \forall j \in D \quad (5)$$

e. Campaña anterior.

$$\sum_{j \in B} x_{j,k} = 1 \quad \forall k \in A \quad (6)$$

f. Separación de al menos una semana.

$$C_j - C_k + (M * y_{j,k}) \geq 10080 \quad \forall j, k \in E \quad k > j \quad (7)$$

$$C_k - C_j + (M * (1 - y_{j,k})) \geq 10080 \quad \forall j, k \in E \quad k > j \quad (8)$$

g. Naturaleza de las variables.

$$x_{j,k} \in \{0,1\} \quad \forall j,k \in T_0 \quad (9)$$

$$y_{j,k} \in \{0,1\} \quad \forall j,k \in E \quad (10)$$

$$C_j, T_j \in \mathfrak{R}^+ \quad (11)$$

Donde las restricciones (1) y (2) aseguran que cada trabajo tiene sólo un predecesor y un sucesor. La restricción (3) especifica que sólo un trabajo puede ser procesado al mismo tiempo. La restricción (4) indica que el tiempo de retraso del trabajo j es mayor o igual que la diferencia entre el tiempo de producción y la fecha de entrega de ese trabajo. La restricción (5) asegura que se cumpla la fecha máxima de completitud del trabajo $j \in D$. La restricción (6) establece que la campaña anterior a la de los trabajos $k \in A$ debe pertenecer al grupo B .

Las restricciones (7) y (8) aseguran que los trabajos $j \in E$ deben estar distanciados al menos una semana en su tiempo de completitud. Finalmente, las restricciones (9), (10) y (11) indican la naturaleza de las variables.

El modelo así propuesto tiene al menos $n(n+1)$ variables binarias, $(n+1)$ variables continuas y, al menos, (n^2+4n+1) restricciones; donde n es el número total de trabajos a secuenciar. De esta forma, la cantidad de variables y restricciones se incrementa con respecto al modelo planteado con anterioridad pero se gana una formulación más simple, desde el punto de vista que se pasa de un modelo de tipo no lineal a un modelo de programación lineal, comparable con aquél desarrollado para la Planta N°1.

5.4. Determinación de Parámetros

Con base en los modelos anteriormente expuestos, se hace necesario determinar y calcular la información correspondiente a los parámetros que éstos necesitan para solucionar una instancia cualquiera dada (*input*).

5.4.1. Tiempo de Procesamiento del Trabajo

Para la determinación de este valor es necesario hacer una salvedad. Dado que, como se explicó en el capítulo 2.3, el sistema ERP de la compañía lleva un registro detallado de operaciones únicamente en la Planta N°2 y un registro agregado a nivel de familias para la Planta N°1; la recolección de datos se realiza en dos etapas.

En una primera etapa, se calcula el tiempo de procesamiento de cada trabajo para la Planta N°2. Éste queda definido como:

$$p_j = \left(\frac{\text{requerimiento}_j}{\text{rendimiento}_j} \right) * 60$$

Donde

$requerimiento_j$ = Requerimientos de Ventas para el mes actual [Ton].

$rendimiento_j$ = Rendimiento promedio del producto j [Ton/hr].

Siendo $requerimiento_j$ un parámetro variable que proviene del pronóstico realizado por el Departamento de Ventas y $rendimiento_j$ un valor entregado por los reportes de información del sistema ERP de la organización. Cabe destacar que el cálculo ha sido realizado sobre una base-tiempo de minutos, escogida arbitrariamente.

En una segunda etapa, se calcula el tiempo de procesamiento de cada trabajo para la Planta N°1. Éste queda definido de igual manera que para el caso de la Planta N°2, sin embargo, la diferencia radica en que el rendimiento promedio a nivel de producto es desconocido.

Pese a ello, se cuenta con información respecto del rendimiento promedio por familia de producto (en total 15 familias para la Planta N°1) así como también la pertenencia de cada producto en cada familia. Una familia agrupa productos que comparten características similares en cuanto a la configuración del laminador a la hora de procesarlos. Luego, se utiliza como una aproximación razonable el rendimiento del producto como el rendimiento promedio de la familia a la cual éste pertenece. Cabe destacar que algunas mediciones en terreno han permitido estimar los errores debidos a esta aproximación en torno al 4%-5%, lo que se traduce finalmente en una desviación de la función objetivo de alrededor de un 6% aproximadamente, tal y como muestra el análisis de sensibilidad del Anexo B.

5.4.2. Fecha de Entrega Ideal del Trabajo

El cálculo de este parámetro involucra una consideración especial. Actualmente, la compañía divide el *mix* total de productos en dos tipos: *Make-to-Stock* (MTS)⁴ y *Make-to-Order* (MTO)⁵, contándose con la información respecto de a qué categoría pertenece cada producto (tanto en la Planta N°1 como en la Planta N°2).

Así la determinación de la fecha de entrega ideal de un trabajo del tipo MTS se realiza en base a una estimación del punto de quiebre de ese producto, suponiendo una demanda lineal. Esto último puede ser expresado mediante la siguiente relación:

$$d_j = \left[\left(\frac{stock_j}{requerimiento_j} * 30 \right) + 1 \right] * 24 * 60$$

Donde

$stock_j$ = Stock del producto j al inicio del período de planificación [Ton].

$requerimiento_j$ = Requerimientos de Ventas para el mes actual [Ton].

⁴ Hecho para Almacenar. Ambiente de manufactura en el que la empresa produce de forma continua artículos por los cuales aún no hay una demanda explícita por parte de algún cliente.

⁵ Hecho a Pedido. Ambiente de manufactura en el que la empresa produce un determinado artículo sólo bajo pedido.

Siendo $stock_j$ un parámetro disponible tanto para la Planta N°1 como para la Planta N°2 en todo momento y a nivel de cada producto. Cabe destacar que la fecha de entrega ha sido calculada sobre una base-tiempo de minutos.

Por otra parte, la fecha de entrega ideal de un trabajo tipo MTO corresponde a un valor entregado directamente por el Departamento de Ventas una vez concretada la operación. En este sentido, el área de Logística debe intentar ajustar la producción para cumplir con dichos compromisos.

5.4.3. Peso Asociado a la Importancia del Trabajo

El modelamiento realizado reconoce el hecho de que dentro del *mix* total existen ciertos productos que son más importantes que otros. Por ejemplo, el retraso de un producto altamente prioritario respecto de su fecha de entrega es mucho más perjudicial que el retraso de un producto menos trascendente.

Es por ello que es importante construir un parámetro que mida la importancia relativa de cada producto dentro del *mix* total. En este sentido, el principal indicador de la relevancia de un determinado producto, lo constituye su volumen de ventas.

De esta forma, es posible definir el peso asociado a la importancia de cada trabajo mediante la expresión:

$$w_j = \frac{ventas_j}{\sum_{j=1}^{j=n} ventas_j}$$

Donde

$ventas_j$ = Promedio de ventas del producto j durante los últimos 6 meses [Ton].

Finalmente, la información necesaria para el cálculo de este parámetro está disponible tanto para la Planta N°1 como para la Planta N°2.

5.4.4. Matriz de Tiempos de Cambio

En la práctica, la calibración del tiempo de *setup* entre un producto y otro distinto es imposible de lograr para todas las combinaciones de productos existentes. Esto se debe a que es conocido por el encargado de realizar la programación el hecho de que existen ciertas secuencias de productos que son ineficientes y por lo tanto nunca son planificadas ni observadas.

Con base en lo anterior, y para poder determinar un tiempo de cambio fidedigno, se ha optado por hacer uso de tablas disponibles que contienen la información a nivel agregado para la Planta N°1 y a nivel semiagregado para la Planta N°2.

En el caso de la Planta N°1, y como ya se ha señalado, es posible contar con la matriz de tiempos de cambio a nivel de familias de productos. Por lo tanto, se ha utilizado como estimación que el tiempo de *setup* entre un producto y otro corresponde

a aquel existente entre las familias a que pertenecen dichos productos. El análisis de sensibilidad del Anexo B permite concluir que la función objetivo no es tan sensible frente a pequeños cambios en los tiempos de *setup*, por lo que la aproximación considerada parece razonable.

Del mismo modo, para la Planta N°2 se cuenta con la información de tiempos de cambio a un nivel semiagregado. Y, a fin de ser consistente con el planteamiento realizado para la Planta N°1, se opta por desagregar esta información a nivel de los distintos tipos de productos que son fabricados en esta planta. Los errores incurridos en la función objetivo producto de esta aproximación son despreciables, tal y como se muestra en el análisis de sensibilidad del Anexo C.

5.4.5. Peso Medidas de Desempeño del Sistema

El valor a asignar a los parámetros adimensionales α y β depende única y exclusivamente de la importancia relativa que se le quiera asignar a la medida de desempeño de eficiencia productiva respecto de la medida de desempeño de calidad de servicio.

En este sentido, el encargado actual de realizar la programación manifiesta que *a priori* no existe ninguna medida que sea dominante. Es decir, existen ocasiones en las que, por determinadas razones, es necesario poner mayor énfasis en reducir el tiempo de producción y, del mismo modo, existen otras ocasiones en las que el énfasis se debe colocar mayormente en cumplir con las fechas de entrega. Sin embargo, en general, se intenta mantener un equilibrio entre ambas medidas de desempeño. Cabe destacar que la discusión anterior es válida tanto para la Planta N°1 como para la Planta N°2.

De esta forma, parece natural considerar un amplio rango de valores para α y β a fin de evaluar la calidad de las soluciones encontradas bajo distintos escenarios. Se propone entonces la estructura de la Tabla N°04.

Tabla N°04

Listado de Valores de α y β a Considerar	
Pesos ($\alpha - \beta$)	0,00 - 1,00
	0,25 - 0,75
	0,50 - 0,50
	0,75 - 0,25
	1,00 - 0,00

Fuente: Elaboración Propia

Sin embargo, como se verá en el siguiente capítulo, la complejidad del problema estudiado requiere del desarrollo de una solución de tipo heurística. Y, debido a la estrategia de resolución adoptada, la evaluación para los distintos valores propuestos en la Tabla N°04 se vuelve sumamente engorrosa.

Prestando atención a lo anterior, se opta por considerar para efectos exclusivamente de calibración de la heurística, los valores $\alpha = 0,5$ y $\beta = 0,5$. Esta elección se basa fundamentalmente en dos razones. Por una parte, al emplear dichos valores se asigna igual importancia a cada una de las medidas de desempeño del sistema, con lo que se estaría cumpliendo el objetivo de mantener el equilibrio. Por otra parte, y tal como se verá en el capítulo 5.5, los tiempos de resolución tienden a ser mayores a medida que α se hace más cercano a β , lo que sugiere que el problema es aún más complejo para estos valores.

Finalmente, se debe señalar que aún cuando la calibración del modelo se realizará en base a los valores antes mencionados de α y β , la evaluación de resultados considerará el desempeño de las soluciones bajo los distintos escenarios descritos en la Tabla N°04.

5.5. Complejidad del Problema Analizado

Los modelos propuestos tanto para la Planta N°1 como para la Planta N°2 contienen una gran cantidad de variables binarias así como también de restricciones, las que evidentemente se incrementan de manera considerable a medida que la cantidad de trabajos a secuenciar crece. A modo de ejemplo, al hablar de 10 trabajos se deben considerar unas 45 variables binarias en el caso de la Planta N°1 y al menos 110 en el caso de la Planta N°2; mientras que con 100 trabajos se tienen casi 4.500 variables en el caso de la Planta N°1 y más de 10.000 en el caso de la Planta N°2. Esto último sugiere, en primera instancia, una gran complejidad para el problema a medida que el tamaño de la instancia (número de trabajos) aumenta.

Es más, desde el punto de vista de la Teoría de Complejidad de los algoritmos, aquella que se ocupa de estudiar y analizar la cantidad de recursos y tiempo requeridos para la ejecución de un determinado algoritmo⁶, el problema de minimizar la suma ponderada del tiempo de completitud y el retraso en una única máquina con tiempos de *setup* es de tipo NP-Hard dado que sus versiones más simples de minimizar la completitud en una única máquina con tiempos de *setup* y minimizar el retraso en una única máquina con tiempos de *setup* también lo son, tal y como fue demostrado por Rinnoy (1976) y Du y Leung (1990) en cada caso respectivamente.

Por lo tanto, el caso bajo estudio es al menos tan complejo como un problema de tipo No-Polinomial (NP). En otras palabras, no se conoce un algoritmo que sea capaz de resolver el problema de manera exacta en tiempo polinomial (y posiblemente no exista tal algoritmo).

Si a esto último se suma que el tamaño de la instancia (número de trabajos) es de aproximadamente 200 trabajos para la Planta N°1 y unos 100 trabajos para la Planta N°2, entonces parece ser que el problema no es abordable de manera exacta con la estrategia de programación entera, al menos no en un tiempo de computación razonable.

⁶ Para un mejor tratamiento del tema, véase GAREY, M. R. y JOHNSON, D. S. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, W. H. Freeman. 338p.

En efecto, ambos modelos (Modelo Planta N°1 y Modelo Planta N°2 Ajustado) fueron compilados y ejecutados para distintos tamaños del problema en un entorno Windows, utilizando el software OPL Studio 3.7 con el solver Cplex en un PC Pentium 4 de 2.4GHz con 512 MB RAM. Los principales resultados -en lo que respecta al tamaño del problema y el tiempo de resolución- se resumen en la Tabla N°05 y la Tabla N°06.

Cabe destacar para ejecutar las distintas corridas del modelo, primero se seleccionaron los 6 productos con mayor volumen de ventas durante los últimos seis meses; a continuación se ejecutó el modelo con los 8 productos de mayor volumen de ventas durante los últimos seis meses y así sucesivamente, como se muestra en la tabla.

Tabla N°05
Tiempo de CPU (seg) para Modelo Planta N°1

Pesos ($\alpha - \beta$)	Número de Trabajos	Tiempo CPU (seg)
0,00-1,00	6	0,96
	8	1,09
	10	1,09
	12	1,33
	13	3,05
0,25-0,75	6	1,01
	8	1,28
	10	5,48
	12	69,58
	13	2742,11
0,50-0,50	6	1,31
	8	1,19
	10	3,98
	12	9,34
	13	9775,23
0,75-0,25	6	1,16
	8	1,24
	10	5,36
	12	803,24
	13	1975,10
1,00-0,00	6	1,14
	8	1,22
	10	6,34
	12	640,76
	13	5045,06

Fuente: Elaboración Propia

Como se ve en la Tabla N°05, los tiempos de resolución para el modelo de la Planta N°1 crecen considerablemente a medida que se incrementa el número de trabajos a secuenciar.

Las soluciones para problemas de tamaño mayor a 13 trabajos no pudieron ser obtenidas dentro de 18.000 segundos de procesamiento (5 horas), por lo que el proceso fue interrumpido al cabo de este tiempo.

Tabla N°06

Tiempo de CPU (seg) para Modelo Ajustado Planta N°2

Pesos ($\alpha - \beta$)	Número de Trabajos	Tiempo CPU (seg)
0,00-1,00	6	1,14
	8	9,95
	9	67,06
	10	613,45
0,25-0,75	6	1,44
	8	38,82
	9	355,45
	10	4048,50
0,50-0,50	6	1,41
	8	44,99
	9	330,34
	10	5361,89
0,75-0,25	6	1,66
	8	39,30
	9	291,77
	10	3092,47
1,00-0,00	6	1,29
	8	32,07
	9	303,93
	10	4016,13

Fuente: Elaboración Propia

Al igual que en el caso anterior, de la Tabla N°06 se desprende que el tiempo de resolución para el modelo de la Planta N°2 crece de forma dramática a medida que aumenta el tamaño del problema. Adicionalmente se observa que, dada una instancia de tamaño n cualquiera, los tiempos de resolución tienden a ser mayores a medida que la relación α sobre β se hace más cercana a uno. Esta relación también parece darse en el caso de la Planta N°1, sin embargo, es menos clara que para el caso de la Planta N°2.

Por otra parte, las soluciones para problemas de tamaño mayor a 10 trabajos no pudieron ser obtenidas dentro de 18.000 segundos de procesamiento (5 horas), por lo que el proceso fue interrumpido al cabo de este lapso. De esta forma, se concluye que la aplicación del modelo de programación entera propuesto resulta impracticable para una instancia de tamaño real del problema -tanto en el caso de la Planta N°1 como en el caso de la Planta N°2-, lo que a su vez demuestra la complejidad del problema bajo análisis.

6. Enfoque de Solución Heurística

Debido a la imposibilidad práctica de aplicar los modelos de programación lineal entera mixta antes propuestos para solucionar instancias reales del problema, parece necesario buscar nuevas estrategias de resolución. En este contexto, se hace deseable desarrollar un algoritmo de tipo heurístico que sea capaz de obtener soluciones lo más cercanas al óptimo posible en un tiempo computacional de proceso razonable.

6.1. Selección del Tipo de Heurística a Aplicar

Los enfoques de solución heurística para problemas de programación de operaciones se pueden clasificar en dos tipos: heurísticas de construcción y heurísticas de mejora. Las heurísticas de construcción utilizan reglas de secuenciamiento para construir una solución al fijar un trabajo en una posición en cada etapa.

Este tipo de enfoques han demostrado ser capaces de obtener programaciones en muy poco tiempo, sin embargo, la calidad de las soluciones es inferior a la obtenida por heurísticas de mejora, especialmente en problemas de gran tamaño (número de trabajos a ser programados).

Las heurísticas de mejora generalmente permiten obtener mejores soluciones que las heurísticas de construcción, pero al mismo tiempo requieren de un mayor esfuerzo computacional. Este enfoque parte de una secuencia inicial y mediante métodos de intercambio en la secuencia busca mejorar la solución. Las principales heurísticas de este tipo se presentan a continuación.

- a. *Simulated Annealing* (SA). En cada etapa, el algoritmo considera un vecino⁷ de la solución actual y probabilísticamente decide entre moverse hacia el vecino o permanecer en el estado actual. Este proceso se repite iterativamente hasta que se alcance una solución “suficientemente buena”, o bien, durante un número fijo de iteraciones.
- b. *Genetic Algorithm* (GA). Los algoritmos genéticos tienen la ventaja de analizar un conjunto de soluciones simultáneamente (en cada etapa). Los nuevos candidatos a solución se generan por combinaciones y “mutaciones” del set de candidatos anteriores. Un criterio probabilístico similar al del SA es empleado para realizar la combinación. El algoritmo termina después de un número fijo de iteraciones o tiempo de computador.
- c. *Beam Search* (BS). Es una adaptación del método *Branch-and-Bound* en la que sólo los nodos más prometedores son analizados en cada nivel del árbol y conservados para una búsqueda futura en los niveles posteriores, mientras que el resto de los nodos son eliminados permanentemente. De esta forma, se logra una heurística cuyo tiempo de resolución es polinomial, pero que no asegura optimalidad para el problema.

⁷ Solución que puede ser obtenida por pequeños cambios o “perturbaciones” de la solución actual.

- d. *Tabu Search* (TS). El algoritmo comienza en una solución inicial e iterativamente se mueve desde la solución actual X a la mejor solución Y encontrada en la vecindad de X (conjunto de vecinos de X), aún cuando la nueva solución sea peor que la anterior. El proceso se repite hasta que algún criterio de detención se cumpla. A fin de evitar comportamientos cíclicos (volver recurrentemente a la solución anterior) y caer en óptimos locales, las soluciones visitadas recientemente son prohibidas o declaradas tabú durante un número determinado de iteraciones. Para ello, se guardan los atributos de estos movimientos en una lista denominada lista tabú.

De todos los enfoques descritos con anterioridad, se ha optado por aplicar el *Tabu Search* como método de resolución para el problema bajo análisis. Esta decisión se fundamenta principalmente en las siguientes cuatro razones:

- a. Simplicidad y flexibilidad. Es una heurística sencilla pero poderosa y que tiene la habilidad de poder comenzar con una implementación simple pero que puede ser mejorada a lo largo del tiempo mediante la incorporación -sin grandes dificultades- de elementos más avanzados y especializados.
- b. Uso de estructuras de memoria. Está diseñada como una alternativa para poder escapar de la trampa del óptimo local a la vez que evitar comportamientos cíclicos en la búsqueda de soluciones. Esto le proporciona ventajas sobre las otras alternativas de búsqueda local analizadas.
- c. Buenos resultados en problemas de programación de operaciones. Este tipo de heurística ha sido satisfactoriamente aplicada a una gran variedad de problemas relacionados directamente con la programación de operaciones, obteniéndose soluciones superiores a todos los otros métodos en un 80% de los casos (Glover, 1989).
- d. Relación calidad de la solución v/s tiempo de resolución. Pese a que hasta la fecha no existen registros en la literatura respecto de comparaciones que se hayan realizado entre todas las heurísticas descritas para el problema específico bajo estudio, sí es posible encontrar algunas comparaciones para otros problemas de programación de operaciones similares. En este sentido, Lin y Ying (2006), Gupta y Smith (2007) y Valente y Alves (2008) reportan resultados que indican que el *Tabu Search* es, en general, uno de los métodos que mejor desempeño tiene en cuanto a la relación calidad de la solución en comparación al tiempo de resolución.

Finalmente, cabe destacar que el último punto señalado es especialmente relevante para cumplir con el objetivo propuesto. Esto se debe a que la implementación del prototipo se debe realizar -por razones de seguridad y permisos impuestos por la organización- íntegramente en *Visual Basic for Applications* (VBA), un lenguaje que en tiempo de ejecución es reconocidamente más lento que sus principales competidores (como JAVA o C++). Es por ello que se requiere de una heurística que permita obtener resultados suficientemente buenos, pero al menor costo de procesamiento posible. Y de todos los enfoques propuestos, *Tabu Search* parece ser la opción más adecuada.

6.2. La Metodología de Búsqueda Tabú o Tabu Search (TS)

Una vez definido el enfoque heurístico a desarrollar, se hace necesario explorar más acabadamente ciertos aspectos del mismo.

El *Tabu Search* o Búsqueda Tabú es un algoritmo heurístico cuya principal característica es el uso de memoria adaptativa y de estrategias especiales de resolución de problemas. Su filosofía es derivar y explotar una colección de estrategias “inteligentes”, basadas en procedimientos de aprendizaje.

Esta técnica fue propuesta originalmente por Glover (1986) y desde entonces ha sido aplicada con gran éxito a distintos problemas de optimización combinatorial complejos, especialmente aquellos que surgen en aplicaciones del mundo real.

Previo a caracterizar el procedimiento general del *Tabu Search*, es necesario conocer y entender algunos de los elementos claves que componen la búsqueda.

6.2.1. Vecindad

A fin de determinar las soluciones que serán evaluadas a lo largo del proceso de búsqueda, se define -en primera instancia- un conjunto de soluciones $N(s)$ las que son obtenidas a partir de pequeñas perturbaciones a la solución s , típicamente a través de algún tipo de intercambio entre algunos elementos del vector de solución s . De esta forma, “una vecindad es construida para identificar soluciones cercanas, que puedan ser alcanzadas desde la solución actual” (Reeves, 1993). El conjunto $N(s)$ se conoce como la vecindad de s y sus elementos se denominan vecinos de s .

Ahora bien, refinamientos más sofisticados de la Búsqueda Tabú consideran que la vecindad puede depender no sólo de la solución actual s , sino que también de la iteración en que se encuentra la búsqueda. Se habla entonces de un conjunto $N(s, k)$, donde k es la iteración, que intenta reflejar el hecho de que después de un determinado número de iteraciones se pueden preferir soluciones que incluyan ciertas características.

Finalmente, es necesario hacer notar que, dependiendo del tipo de vecindad escogida, puede suceder que ésta contenga una gran cantidad de elementos. En estos casos, y dado el costo computacional al examinar todos los elementos, se puede considerar un subconjunto de la vecindad $V^* \subseteq N(s, k)$ tal que $|V^*| \ll |N(s, k)|$ a fin de permitir una evaluación más rápida en cada iteración y mantener el carácter agresivo de la búsqueda.

6.2.2. Lista Tabú

Para evaluar las soluciones vecinas a s y escoger el mejor movimiento dentro de la vecindad, se pueden comparar los valores de la función objetivo de cada vecino y escoger la menor (en caso de estar minimizando). De esta forma se ha encontrado un mínimo local s' para la vecindad.

El proceso así descrito corre el riesgo de volver recurrentemente a la solución anterior s , es decir, el mejor elemento de la vecindad de s es precisamente s . En tal caso, el algoritmo queda atrapado en un óptimo local irremediablemente. A fin de evitar este problema, el método presenta un procedimiento adaptativo de búsqueda que le permite continuar explorando el espacio factible y aceptar soluciones, aún cuando éstas sean peores que la solución actual.

Es por ello que se hace necesario registrar no sólo la información local (valor actual de la función objetivo) sino que además llevar un registro del itinerario recientemente realizado, de forma de restringir los posibles vecinos sobre los que va a avanzar el proceso de búsqueda. En este caso, una solución ya elegida será marcada como tabú (prohibida o penalizada) por un determinado número de iteraciones (*tenure*), de modo de impedir que el algoritmo retorne a una solución previamente visitada y pueda ocurrir un comportamiento cíclico.

Para llevar a cabo tal consideración, se crea una lista T -denominada Lista Tabú- que guarda las últimas $|T|$ soluciones visitadas y las marca como soluciones tabú. *Tabu Search* incorpora de esta manera una estructura de memoria reciente o de corto plazo (*short term memory*) como una de sus principales características para superar la antes descrita “trampa del óptimo local”. Cabe destacar que en tal caso es posible asegurar que se pueden prevenir ciclos sólo durante las $|T|$ iteraciones siguientes.

La implementación de la Lista Tabú puede ser explícita, es decir, se guarda toda la información de la solución ya visitada; o bien, por atributos. En este último caso, se guardan únicamente ciertos atributos o características importantes de la solución recientemente visitada.

Por otra parte, si bien existe un mecanismo que prohíbe ciertos movimientos, el número de iteraciones por las cuales una solución será considerada tabú es de igual importancia. Si el tamaño de la lista tabú es muy grande, existe la posibilidad de que todas las soluciones vecinas sean prohibidas; mientras que si el tamaño de la lista es muy pequeño, existe una mayor probabilidad de volver a una solución anterior y obtener un comportamiento cíclico en torno a un mínimo local.

6.2.3. Criterio de Aspiración

Aunque la Lista Tabú puede guardar la información completa de las últimas $|T|$ soluciones visitadas, es recomendable -por razones de eficiencia- guardar sólo una parte de la información que describe a las soluciones recorridas. El problema de guardar sólo una parte de la solución es que pueden existir otras soluciones aún no visitadas que en esa parte de la información sean iguales y, en particular, una solución no visitada puede ser atractiva. Lo anterior hace necesario definir un criterio de aspiración, que permita aceptar soluciones, aún cuando sus atributos estén en la lista tabú.

En la práctica, el criterio de aspiración más utilizado consiste en eliminar la condición tabú de un movimiento cuando éste conduce a una solución mejor que la mejor encontrada hasta el momento. Sin embargo, existen otros criterios que pueden utilizarse, como por ejemplo, si todos los movimientos disponibles está clasificados como tabú entonces se selecciona el movimiento “menos tabú” (Glover y Melián, 2003).

6.2.4. Criterio de Detención

La condición de parada de la Búsqueda Tabú queda expresada en el criterio de detención. Aunque esta condición puede ser cualquiera definida por el usuario, típicamente se utilizan algunos criterios tales como un determinado número de iteraciones o tiempo de computador (CPU), no encontrar soluciones factibles para continuar iterando, un determinado número de iteraciones sin mejora en la solución, alcanzar una solución que sea mejor que un cierto valor fijado al inicio o incluso distintas combinaciones de éstos criterios.

6.2.5. Algoritmo de Búsqueda Tabú

Con todos estos elementos en mente, ya es posible delinear formalmente el algoritmo de Búsqueda Tabú. Se define en primer lugar.

s, s' = Dos soluciones.

k = Índice de la iteración.

$N(s, k)$ = Vecindad de la solución s en la iteración k .

V^* = Subconjunto de soluciones.

$f(s)$ = Valor de la función objetivo para la solución s .

Entonces, el algoritmo de Búsqueda Tabú queda descrito de la siguiente forma.

Paso 1. Escoger una solución inicial s en el espacio de soluciones factibles S . Hacer $s^* = s$ y $k = 0$.

Paso 2. Hacer $k = k + 1$ y generar un subconjunto V^* de soluciones en $N(s, k)$, de modo que ningún elemento está incluido en la lista tabú o al menos cumpla con el criterio de aspiración.

Paso 3. Escoger el mejor s' en V^* (con respecto a la función objetivo f o una función \tilde{f} escogida de antemano y que permita evaluar la calidad de las distintas soluciones). Hacer $s = s'$.

Paso 4. Si $f(s) < f(s^*)$, en caso de un problema de minimización, entonces hacer $s^* = s$.

Paso 5. Actualizar la Lista Tabú y el criterio de aspiración.

Paso 6. Si el criterio de detención se cumple, entonces parar. En caso contrario, volver al Paso 2.

6.2.6. Estrategias de Intensificación y Diversificación

El algoritmo recién expuesto, es denominado *Standard Tabu Search* o Búsqueda Tabú Estándar. Sin embargo, es posible incorporar algunas sofisticaciones que, haciendo uso de otras estructuras de memoria, permitan mejorar el proceso de búsqueda.

En efecto, la Búsqueda Tabú puede también ser utilizada como un proceso de aprendizaje al incorporar estructuras de memoria de largo plazo (*long term memory*).

En este contexto, y habiendo visitado una cantidad considerable de soluciones, es posible volver a buenas regiones ya exploradas para estudiarlas más a fondo. Para ello se restringe la vecindad $N(s, k)$ o se altera el criterio para evaluar los movimientos, a favor de la aparición de aquellos atributos que se han encontrado frecuentemente en buenas soluciones previamente visitadas. Esto representa un esquema de intensificación para la búsqueda.

Por el contrario, la diversificación persigue una búsqueda más efectiva sobre el espacio de soluciones, permitiendo que el procedimiento explore nuevas regiones no visitadas hasta el momento. Tal estrategia puede ejecutarse modificando las reglas de elección para incorporar a las soluciones atributos que no han sido usados frecuentemente.

6.3. Alternativas Adoptadas para la Implementación de la Búsqueda Tabú

El comportamiento y desempeño de los algoritmos de tipo heurístico depende tanto de las características del problema como del tipo de instancia considerada. Es por esto que, aún en problemas muy similares, una heurística puede ser de gran calidad al solucionar uno de los dos problemas pero sistemáticamente deficiente al aplicarla al otro.

Por otro lado, gran parte del éxito de la Búsqueda Tabú radica en la forma o estructura asignada a cada uno de los distintos componentes del algoritmo (descritos en el capítulo 6.2). En particular, existen dos elementos clave que tienen gran injerencia en el resultado final: la solución inicial y el método de generación de la vecindad. Esto último se debe a que la efectividad y eficiencia de la búsqueda se incrementan al comenzar en una solución lo más cercana al óptimo posible (solución inicial) a la vez que los movimientos se realizan en la dirección correcta (vecindad).

Después de una investigación minuciosa, se ha elaborado un listado de las metodologías más populares para determinar soluciones iniciales y vecindades, las que empíricamente han mostrado tener los mejores resultados en la mayoría de los casos.

Tomando en consideración todo lo anteriormente mencionado, la estrategia a utilizar para resolver el problema estudiado consiste en contrastar el comportamiento de la Búsqueda Tabú bajo distintas combinaciones de metodologías de obtención de soluciones iniciales y estrategias de generación de vecindades, a fin de escoger finalmente aquella que mejores resultados presente.

Finalmente, existe un tercer elemento que no puede ser ignorado en el proceso: aquel que dice relación con la estructura y largo de la Lista Tabú. Es por ello que, adicionalmente, se hace necesario estudiar las distintas combinaciones para diferentes largos.

6.3.1. Solución Inicial

Tal y como fue antes mencionado, la estrategia de resolución del problema considera la implementación de distintas metodologías de obtención de soluciones iniciales, a fin de evaluar su desempeño -en conjunto con una estrategia de generación de vecindades dada- tanto para la Planta N°1 como para la Planta N°2.

A continuación, se presentan los distintos mecanismos de obtención de soluciones iniciales a utilizar. Cabe destacar que todas las metodologías consideradas pueden ser clasificadas como del tipo heurística de construcción (véase capítulo 6.1), por lo que permiten obtener soluciones muy rápidamente.

6.3.1.1. Solución Aleatoria (Random)

Consiste en determinar un orden para la secuencia de manera completamente aleatoria, sin importar si la solución obtenida es o no factible. Tiene la ventaja de ser una solución muy fácil de implementar.

6.3.1.2. Weighted Shortest Processing Time (WSPT)

Consiste en ordenar los trabajos en forma decreciente, de acuerdo al índice:

$$w_j/p_j$$

Donde

w_j = Peso o importancia asociada al trabajo j .

p_j = Tiempo de procesamiento del trabajo j .

Cabe destacar que esta heurística entrega la solución óptima para el caso de programar trabajos en una sola máquina donde el objetivo es minimizar únicamente el tiempo total de ciclo.

6.3.1.3. Earliest Due Date (EDD)

Consiste en ordenar los trabajos de acuerdo a la proximidad de su fecha de entrega, en forma ascendente. Esta heurística entrega la solución óptima para el caso de programación de trabajos en una sola máquina donde el objetivo es minimizar únicamente el retraso máximo de entre todos los trabajos.

6.3.1.4. Apparent Tardiness Cost with Setups (ATCS)

Heurística desarrollada por Lee, Bhaskaran y Pinedo (1997), la que puede derivar una solución muy rápidamente. El costo aparente de retraso con tiempos de cambio está definido como sigue:

$$ATCS_j(t,i) = \frac{1}{p_j} \exp\left(-\frac{\max(d_j - p_j - t, 0)}{k_1 * \bar{p}} - \frac{s_{i,j}}{k_2 * \bar{s}}\right)$$

Donde

t = tiempo actual.

i = trabajo que se acaba de completar.

p_j = tiempo de procesamiento del trabajo j .

d_j = fecha de entrega del trabajo j .

\bar{p} = tiempo promedio de procesamiento.

\bar{s} = tiempo promedio de *setup*.

$s_{i,j}$ = tiempo de *setup* incurrido si el trabajo j sucede al trabajo i .

k_1, k_2 constantes calibradas según la metodología expuesta en el Anexo D.

Entonces la regla indica que el siguiente trabajo j para agregar en la secuencia es simplemente:

$$j = \arg \max_j ATCS_j(t,i)$$

La gran ventaja de este tipo de solución es que es la única heurística de construcción que incorpora los tiempos de *setup* como una consideración importante en la programación de operaciones. Es más, Lin y Ying (2006) la consideran la mejor heurística de construcción para el problema de programación de operaciones en una sola máquina con tiempos de preparación dependientes de la secuencia. Sin embargo, los mismos autores se adelantan en aclarar que “aunque ATCS puede derivar una solución factible rápidamente, la calidad de la solución es pobre, especialmente para grandes problemas”.

6.3.2. Vecindad

La estrategia de generación de vecindades que se aplique determina la forma en que se “moverá” el algoritmo dentro del espacio de solución, es decir, la dirección que tomará la búsqueda.

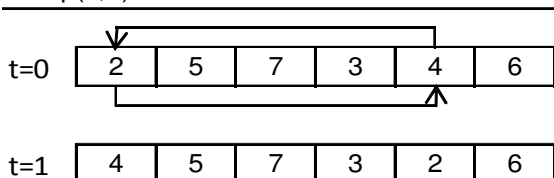
Es por ello que, en lugar de optar *a priori* por un mecanismo único como podría ser el popular API, lo más razonable parece ser evaluar el desempeño de distintas estrategias y después seleccionar la más adecuada.

Los distintos mecanismos de generación de vecindades se basan en dos tipos fundamentales de movimiento. Un movimiento hace referencia a una perturbación de una solución s dada. Estos dos movimientos son:

- a. Intercambio (*Swap*). Consiste en intercambiar los elementos ubicados en la i – *esima* y j – *esima* posición respectivamente, donde $i \neq j$. Una representación del movimiento es mostrada en la Figura N°03, en la cual se han intercambiado los elementos en las posiciones (1,5).

Figura N°03

Swap(1,5)

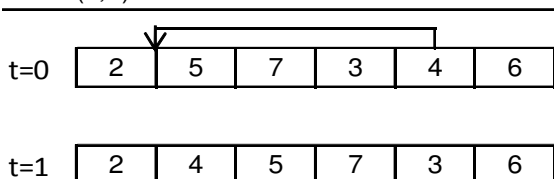


Fuente: Elaboración Propia

- b. Inserción (*Insert*). Consiste en remover el elemento en la i – *esima* posición e insertarlo en la j – *esima* posición, donde $i \neq j$. Una ilustración del movimiento es presentada en la Figura N°04, en la cual se ha realizado una inserción entre los elementos en las posiciones (5,2).

Figura N°04

Insert(5,2)



Fuente: Elaboración Propia

Pues bien, en base a estos tipos básicos de movimiento es posible definir más formalmente las distintas estrategias de generación de vecindades a utilizar en la Búsqueda Tabú.

6.3.2.1. Adjacent Pairwise Interchange (API)

Por lejos, el método más común para generar una vecindad a partir de una solución actual s es el API. Esta operación consiste en realizar movimientos de tipo *swap* entre cada elemento de la secuencia y aquel inmediatamente posterior.

Debido a que intercambiar los elementos en las posiciones i y j es equivalente a intercambiar los elementos en las posiciones j e i , no es difícil notar entonces que una secuencia cualquiera de largo n dará origen a una vecindad de tamaño $(n - 1)$. Luego, se concluye que el tamaño de la vecindad es relativamente pequeño $O(n)$ y fácil de generar. Sin embargo, al mismo tiempo, se limita el número de nuevas opciones en las cuales se puede realizar la búsqueda; por lo que evidentemente se establece un *trade-off*.

6.3.2.2. General Pairwise Interchange (GPI)

Este es el segundo método más utilizado para generar vecindades en la práctica. El procedimiento considera la vecindad generada por cada posible intercambio entre los elementos de la secuencia (y no sólo los adyacentes), lo que da origen a una vecindad que contiene $n(n - 1)/2$ nuevas secuencias. Luego, el tamaño de la vecindad generada por el GPI es $O(n^2)$, razón por la cual es considerada como una vecindad de gran escala.

En la práctica, sin embargo, el esfuerzo computacional derivado del cálculo y análisis de este tipo de vecindad dificulta su uso en problemas de gran envergadura. Es más, para el problema específico bajo estudio, la Tabla N°07 muestra el tamaño de la vecindad vía GPI asociada al problema de la Planta N°1 y la Planta N°2.

Tabla N°07

Tamaño Vecindad (GPI)

	Número Trabajos	Tamaño Vecindad
Planta N°1	217	23.436
Planta N°2	109	5.886

Fuente: Elaboración Propia

Una posible solución al problema -y la más común- consiste en analizar simplemente una parte de la vecindad en cada iteración. Es por ello que se ha optado por seguir la siguiente estrategia: en la primera iteración se ejecutarán todos los intercambios GPI disponibles entre el primer elemento de la secuencia y hasta un límite determinado por la aproximación entera del resultado de dividir el largo de la secuencia en tres. Luego, durante la segunda iteración, se realizarán todos los intercambios GPI posibles entre el límite anterior y este valor multiplicado por dos. Para la tercera iteración se ejecutarán todos los intercambios GPI entre el límite anterior y el último elemento de la secuencia.

Finalmente, en la cuarta iteración se vuelve a analizar el primer tercio de intercambios tipo GPI, durante la quinta iteración el segundo tramo y así sucesivamente.

De esta forma, se obtiene un tamaño de la vecindad considerablemente más pequeño y, en consecuencia, menos costoso en tiempo de procesamiento computacional, asumiendo claro está, la reducción de posibilidades de búsqueda. Sin embargo, la estrategia indicada tiene la ventaja de considerar el movimiento parcial entre las distintas regiones de la vecindad con gran dinamismo a lo largo de todo el proceso de búsqueda.

6.3.2.3. Top Priority to the Front (TPF)

Una estructura de vecindad distinta consiste en considerar todos los posibles trabajos j que se pueden insertar en la primera posición. Luego, esta vecindad considera un movimiento de tipo inserción hacia la primera posición, que es aquella que debería tener la más alta prioridad en la secuencia.

Por otra parte, es directo que una secuencia cualquiera de largo n dará origen a una vecindad de tamaño $(n - 1)$. De esta forma, se obtiene una vecindad relativamente pequeña $O(n)$ y fácil de generar.

6.3.2.4. General Top Priority to the Front (GTPF)

Este tipo de vecindad es considerada una generalización de la anterior. En primer lugar, se prueban las $(n - 1)$ alternativas de inserción en la primera posición. Luego, se fija la mejor primera opción y a continuación se prueban las $(n - 2)$ opciones en la segunda posición; y así sucesivamente.

La operación antes descrita produce una vecindad de alrededor de $n(n - 1)/2$ nuevas posibles secuencias, por lo que puede ser considerada de gran escala. Es por ello, que se ha optado por seguir exactamente la misma estrategia de reducción de tamaño de vecindad adoptada para el *General Pairwise Interchange*. Con ello, se obtiene un tamaño de la vecindad considerablemente menor, lo que reduce el esfuerzo computacional.

6.3.2.5. General Insertion Method (GIM)

Este tipo de estructura de vecindad considera cada posible inserción entre los elementos de la secuencia, y no sólo aquellas realizadas hacia la primera posición. Al igual que en el caso anterior, el tamaño de la vecindad así generada es $O(n^2)$, lo que evidentemente dificulta el cálculo computacional.

Es así que, como en todas las vecindades de gran escala antes propuestas, se opta por aplicar la estrategia de reducir su tamaño y buscar alternadamente entre las distintas regiones que componen dicha vecindad, tal y como se describe en el capítulo 6.3.2.2.

6.3.3. Lista Tabú

Con la intención de lograr una implementación eficiente en términos de consumo de recursos de memoria, es necesario poner énfasis en otros aspectos importantes de la Búsqueda Tabú y, en particular, en lo que respecta a la estructura de la Lista Tabú.

En consecuencia, se ha optado por hacer uso de una Lista Tabú en base a una memoria atributiva. Es decir, en lugar de memorizar cada solución s por completo, se guardará la información respecto del movimiento que dio origen a esa solución (atributos).

De esta forma, cada solución que ingrese a la Lista Tabú quedará descrita por el par (i, j) ; el que representará ya sea el intercambio entre el elemento en la i -ésima posición con el elemento en la j -ésima posición, o bien la inserción del elemento en la i -ésima posición en el elemento en la j -ésima posición.

6.3.4. Criterio de Aspiración

El problema de la estructura de información de la Lista Tabú presentada en el capítulo anterior es que al guardar sólo la información de dos atributos -definidos por el movimiento de los elementos (i, j) - pueden existir otras soluciones aún no visitadas que en esa parte de la información sean iguales.

En otras palabras, puede existir la posibilidad de que un movimiento que está prohibido (clasificado como tabú) sea atractivo, porque al ser elegido genere una solución mejor que todas las encontradas hasta el momento.

A fin de evitar el problema anterior, se define el siguiente criterio de aspiración: eliminar la condición tabú de un movimiento cuando éste conduce a una solución mejor que la mejor encontrada hasta el momento.

6.3.5. Criterio de Detención

Tal y como se vio en el capítulo 6.2.4, existe una gran cantidad de criterios de detención para el algoritmo de Búsqueda Tabú. Sin embargo, para la resolución del problema específico bajo estudio se ha optado por fijar un determinado tiempo de ejecución.

La elección anterior se basa fundamentalmente en que, por una parte, la implementación de tal criterio es rápida y simple; y, por otra parte, se obtiene la flexibilidad de poder variar arbitrariamente el tiempo empleado y estudiar el comportamiento de la heurística si así se desea.

6.3.6. Estrategia de Diversificación

Como se mencionó en el capítulo 6.2.6, un esquema de diversificación consiste típicamente en dirigir la búsqueda hacia regiones que aún no han sido exploradas. En palabras de Beausoleil (2000) "el objetivo es producir un nuevo punto de partida desde donde el proceso de búsqueda pueda continuar".

Basándose en este concepto, la forma más fácil de generar un nuevo punto de partida es simplemente hacerlo de manera aleatoria. Sin embargo, se propone aquí una alternativa sencilla, posiblemente más inteligente y poco costosa en términos de implementación, la cual consiste en alcanzar una nueva región mediante la perturbación de una solución con un método de generación de vecindad distinto al que está siendo empleado.

Luego, la estrategia puede enunciarse como sigue. Si durante un determinado número de iteraciones la mejor solución encontrada hasta el momento no ha cambiado, entonces se gatilla el mecanismo de diversificación en virtud del cual se aplica un método de generación de vecindad distinto -por otro número fijo de iteraciones- para después volver al que se ha definido originalmente para la búsqueda estándar.

Finalmente, cabe mencionar que si bien la estrategia propuesta es sencilla y fácil de implementar, existen varios elementos que deben ser calibrados en el proceso: el número de iteraciones previas sin mejoras antes de pasar a la etapa de diversificación, las iteraciones durante las cuales se perturba la solución actual y el método de generación de vecindad con que esto se lleva a cabo. Debido a que a *priori* no es posible determinar qué combinación tendrá mejores resultados, se opta nuevamente por probar varias y estudiar el desempeño del algoritmo.

6.3.7. Estrategia de Intensificación

La técnica denominada *Path Relinking* ha sido sugerida por Glover y Laguna (1997) como un buen mecanismo para integrar la intensificación al proceso de Búsqueda Tabú. Aquí se hará una breve descripción de la técnica poniendo especial énfasis en los detalles de implementación, pero para un mejor tratamiento del tema puede consultarse Laguna and Martí (1997). En el presente proyecto, la técnica ha sido incluida al final del algoritmo como un procedimiento de post-optimización.

De esta forma, la implementación del *Path Relinking* queda como sigue. A lo largo del proceso de búsqueda se almacena un pequeño conjunto de las mejores soluciones encontradas (denominadas soluciones *elite*) para luego, una vez terminada la búsqueda, escoger una de ellas (denominada solución de inicio o *initiating solution*) y generar nuevas soluciones al explorar trayectorias que la conectan con otra “buena” solución (denominada solución guía o *guiding solution*) del conjunto. Esto último se logra utilizando movimientos de inserción, que introducen los atributos contenidos en la solución guía. Ahora bien, si la función objetivo mejora, entonces se guarda la solución. El procedimiento anterior se repite para todas las combinaciones entre solución de inicio y solución guía que se puedan generar sobre el conjunto de soluciones elite.

Cabe destacar que el principal parámetro de la estrategia propuesta consiste en determinar el número de soluciones *elite*, pues resulta evidente que se produce un *trade-off* entre el largo del conjunto y el tiempo empleado para ejecutar la fase de intensificación. Es por ello que se analizará el comportamiento del algoritmo para distintos largos y se escogerá finalmente aquel que presente la mejor relación costo-beneficio.

6.3.8. Consideraciones Adicionales

Tal y como ha sido expresada en las secciones precedentes, la heurística del tipo *Tabu Search* propuesta está desarrollada para resolver completamente el problema específico de la Planta N°1.

Sin embargo, es necesario recordar que el caso de la Planta N°2 presenta restricciones adicionales que deben ser incorporadas por el modelo. De esta forma -y a fin de permitir la reutilización de la heurística principal desarrollada para la Planta N°1- la estrategia a seguir para la Planta N°2 es, una vez obtenida una secuencia final mediante el algoritmo de Búsqueda Tabú descrito, incorporar las otras restricciones del problema mediante permutaciones definidas por movimientos de inserción, las que obliguen a la secuencia a satisfacer todo el conjunto de restricciones.

La estrategia anterior tiene ventajas y desventajas. La gran desventaja consiste en que actuar después de obtener los resultados de la Búsqueda Tabú para incorporar las restricciones adicionales, sin duda perjudicará la función objetivo en mayor medida que desarrollar una heurística completamente nueva que incorpore dichas restricciones desde el comienzo. Sin embargo, la ventaja de esta elección es que puede hacerse uso de los resultados y el trabajo realizado para solucionar el problema de la Planta N°1.

7. Calibración de la Heurística

La implementación y programación del algoritmo heurístico del tipo *Tabú Search* propuesto fue realizada en un entorno *Windows* bajo el lenguaje de programación *Visual Basic for Applications* (VBA).

Cabe destacar que, si bien este lenguaje es comparativamente inferior a otros disponibles (como Java o C++) en lo que respecta a rapidez de cómputo y manejo de memoria, existen una serie de restricciones de seguridad y privilegios de permiso en la organización que hacen imposible programar en otro tipo de plataforma. Es por ello que el desarrollo de la Búsqueda Tabú descrita en el capítulo 6.3 incorpora una serie de consideraciones especialmente dirigidas a mejorar la eficiencia del proceso y reducir el uso de recursos de memoria, tal y como se especificó en su momento.

Dicho lo anterior, sólo falta agregar que de obtener resultados positivos durante la presente investigación, el Departamento de Tecnologías se muestra dispuesto a comenzar una implementación formal del modelo. Sin embargo, este hecho escapa del alcance del presente proyecto.

7.1. Desempeño Búsqueda Tabú

Con la intención de mantener la consistencia de la estrategia de resolución del problema propuesta en el capítulo 6.3, es necesario comparar el desempeño de la Búsqueda Tabú bajo distintas combinaciones de métodos de obtención de soluciones iniciales y generación de vecindades, así como para distintos largos de la Lista Tabú.

Con este fin, se ha empleado como referencia para las distintas corridas de la heurística una instancia real para el problema. Dicha instancia es considerada como un “mes normal promedio” de acuerdo a la información histórica disponible y además bajo el juicio experto del encargado actual de la programación de actividades. Por ello, se considera adecuada para realizar la calibración de la Búsqueda Tabú. El valor de los parámetros empleados se muestra en la Tabla N°08. Cabe destacar que todos los cálculos han sido realizados considerando el problema de la Planta N°1.

Tabla N°08

Parámetros Empleados Búsqueda Tabú (1)	
α	0,5
β	0,5
Tiempo Resolución	20 min

Fuente: Elaboración Propia

Adicionalmente, se ha fijado arbitrariamente un tiempo de resolución equivalente a 20 minutos porque se busca asegurar la elección de aquellas combinaciones-estrategias que posibiliten una búsqueda más agresiva, es decir, que permitan evaluar la mayor cantidad de soluciones posibles en igual tiempo de procesamiento (lo que incrementa la probabilidad de encontrar mejores soluciones). Esto debido a las limitaciones del lenguaje de programación empleado. La Tabla N°09 presenta un resumen de los principales resultados, donde cada celda indica el valor obtenido para la función objetivo bajo la combinación de estrategia de generación de vecindad, tipo de solución inicial y largo de Lista Tabú correspondiente.

Tabla N°09

Desempeño Búsqueda Tabú Tiempo Procesamiento: 20 minutos

Vecindad	Solución Inicial	Largo Lista Tabú (% Tamaño Vecindad)				
		5%	10%	20%	30%	40%
API	Aleatoria*	0,12478624	0,08229455	0,07968957	0,07732620	0,10790217
	WSPT	0,08691558	0,08697558	0,07571749	0,07693308	0,07585907
	EDD	0,09158098	0,07991589	0,07774933	0,07677732	0,07800185
	ATCS	0,11134409	0,09430983	0,07532008	0,07614431	0,07704332
GPI	Aleatoria*	0,09989118	0,09501488	0,09993265	0,09880600	0,09814109
	WSPT	0,07885000	0,08133840	0,08133840	0,08133840	0,08133840
	EDD	0,08055492	0,08204130	0,08236544	0,08236544	0,08236544
	ATCS	0,09655333	0,08619825	0,09811091	0,09811091	0,09811091
TPF	Aleatoria*	0,15010883	0,14422567	0,12399123	0,14319328	0,13008787
	WSPT	0,10031939	0,09947286	0,09350993	0,08943833	0,09168534
	EDD	0,11368502	0,10734593	0,10597652	0,09928394	0,10417503
	ATCS	0,13970863	0,13758119	0,13574760	0,12667323	1,13963180
GTPF	Aleatoria*	0,10768059	0,09989588	0,09919620	0,09790975	1,00860825
	WSPT	0,08483092	0,08483092	0,08483092	0,08483092	0,84830922
	EDD	0,09418321	0,09418321	0,09418321	0,09418321	0,94183207
	ATCS	0,09793927	0,09263408	0,09793927	0,09793927	0,97939266
GIM	Aleatoria*	0,08920915	0,09583126	0,89715144	0,10470332	1,05068570
	WSPT	0,07949628	0,08108772	0,08108772	0,08108772	0,81087721
	EDD	0,08401126	0,08696384	0,86963838	0,08696384	0,86963838
	ATCS	0,08885076	0,09026828	0,09091903	0,09091903	0,90919028

* : Promedio cinco corridas

Fuente: Elaboración Propia

De la Tabla N°09 es posible constatar que el menor valor para la función objetivo se alcanza para la combinación que involucra la solución inicial ATCS, la creación de vecindades por el método API y un largo para la Lista Tabú equivalente al 20% del tamaño de la vecindad. Sin embargo, el desempeño de esta combinación empeora considerablemente al variar ligeramente alguno de los parámetros involucrados. Adicionalmente se debe considerar que, puesto que se está haciendo uso de una instancia particular (aún cuando ésta sea considerada “promedio” o “normal”), sus características específicas pueden llevar a que los resultados expuestos en la Tabla N°09 varíen levemente al aplicar el algoritmo a otras instancias del problema.

Como conclusión entonces, en lugar de poner énfasis en la mejor combinación para la instancia específica estudiada (resultados puntuales), el análisis de resultados se debe basar más bien en las tendencias de comportamiento observadas en la Tabla N°09, tomando como supuesto el que éstas son similares para otras instancias del problema.

7.2. Análisis de Tendencias de Comportamiento

Tomando como resultado la conclusión anterior, se pone ahora énfasis en estudiar las tendencias observadas. En este sentido, de la Tabla N°09 se desprenden varios elementos relevantes para el análisis. En primer lugar -en lo que respecta al largo de la Lista Tabú- debe mencionarse que, en términos generales, parece moverse en torno al 20% para obtener los mejores resultados bajo la estrategia de generación de vecindades *Adjacent Pairwise Interchange* (API), en el 5% para el *General Pairwise Interchange* (GPI), un 30% para el *Top Priority to the Front* (TPF) y un 5% para la estrategia *General Insertion Method* (GIM). Los resultados también sugieren que la solución parece ser insensible al largo de la Lista Tabú bajo la estrategia *General Top Priority to the Front* (GTPF).

Es interesante observar en este sentido que aquellas vecindades de gran escala, como GPI y GIM tienden a generar mejores soluciones para largos de la Lista Tabú más pequeños. Por el contrario, las vecindades de menor tamaño (API y TPF) se desenvuelven mejor para largos mayores.

Por otra parte, la estrategia de vecindades API tiende -en general- a entregar mejores resultados que todas las otras consideradas, independiente del método de solución inicial y el largo de la Lista Tabú. Esto último puede deberse a que, al ser una vecindad de pequeña escala, es capaz de realizar un mayor número de iteraciones que una de gran escala (pese a las reducciones incorporadas) para un mismo tiempo de resolución.

Finalmente, falta considerar el efecto producido en el resultado final por el método de solución inicial. De la Tabla N°09 se desprende que, en general, una solución inicial aleatoria entrega peores resultados que un método estructurado y más inteligente para construir la secuencia de inicio, independientemente de la estrategia de generación de vecindad y el largo de la Lista Tabú empleado. De los métodos restantes, se observa que el *Weighted Shortest Processing Time* (WSPT) frecuentemente conduce a resultados superiores o similares que el *Earliest Due Date* (EDD) y el *Apparent Tardiness Cost with Setups* (ATCS).

Resumiendo entonces, se concluye que las mejores soluciones tienden a ser alcanzadas, en general, para una combinación compuesta por el método de solución inicial WSPT, una estrategia de generación de vecindades del tipo API y un largo de la Lista Tabú del 20% de la cantidad de vecinos.

7.3. Calibración Tiempo de Ejecución

Ahora bien, fijando los parámetros determinados en el capítulo anterior como la mejor combinación encontrada dentro de todas las analizadas, falta calibrar únicamente el tiempo de ejecución que se asignará a la heurística. Para ello, se ha ejecutado nuevamente la Búsqueda Tabú para los parámetros descritos en la Tabla N°10.

Tabla N°10

Parámetros Empleados Búsqueda Tabú (2)	
α	0,5
β	0,5
Largo Lista Tabú	20% Vecindad
Tiempo Ejecución	8 horas
Solución Inicial	WSPT
Generación Vecindad	API

Fuente: Elaboración Propia

Donde se ha considerado un tiempo de resolución de 8 horas, que corresponde al intervalo empleado actualmente en el proceso de programación manual de operaciones en ambas plantas. Los resultados obtenidos se resumen en la Tabla N°11. Cabe destacar que el porcentaje de mejora representa la variación del valor de la función objetivo con respecto a la solución anterior.

Tabla N°11

Desempeño Búsqueda Tabú Como Función Tiempo Resolución		
Tiempo Ejecución (min)	Función Objetivo (F.O.)	Mejora (%)
1	0,09850034	-
5	0,08767015	10,995
10	0,08138867	7,165
20	0,07571749	6,968
30	0,07547355	0,322
120	0,07535665	0,155
180	0,07517372	0,243
240	0,07491438	0,345
480	0,07454453	0,494

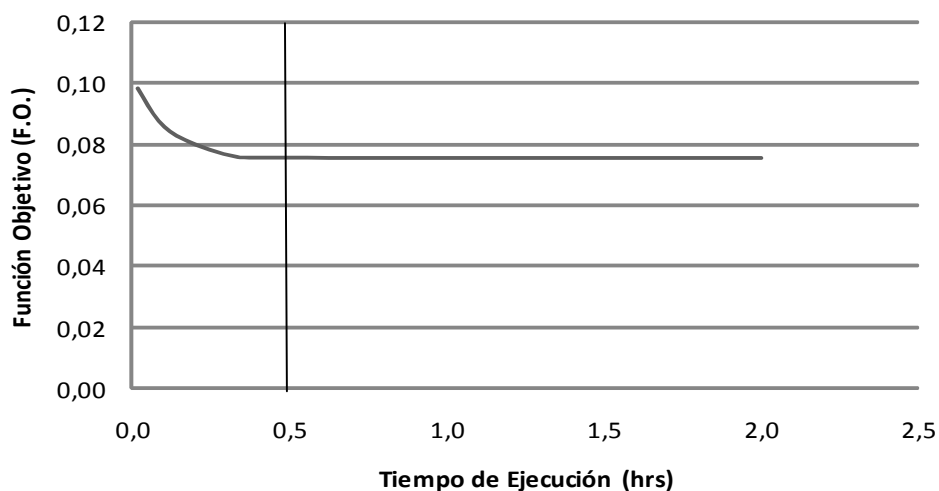
Fuente: Elaboración Propia

De la tabla se desprende en primer lugar que, como era de esperarse, el proceso de búsqueda permite alcanzar buenos resultados rápidamente. Sin embargo, a medida que el tiempo de resolución crece, se vuelve cada vez más difícil encontrar mejores soluciones a la vez que las ganancias en la función objetivo van disminuyendo.

Queda claro entonces que es perfectamente factible fijar un tiempo de ejecución relativamente bajo sin comprometer de manera importante el valor de la función objetivo (calidad de la solución). Para determinar dicho intervalo de tiempo, en la Figura N°06 se han graficado los valores de la Tabla N°11.

Figura N°06

Comportamiento Función Objetivo en Relación al Tiempo de Ejecución



Fuente: Elaboración Propia

Luego, aplicando la “regla del codo” (Pla, 1986), según la cual se debe seleccionar aquel punto en torno al cual se observa la inflexión de la curva en la Figura N°06, se fija el tiempo de ejecución de la Búsqueda Tabú en 30 minutos (0,5 horas). Esto último pues tiempos de resolución mayores no producen un aporte significativo en términos de la mejora en la función objetivo (análisis costo-beneficio).

7.4. Mejoras al Algoritmo Propuesto

Una vez encontrada la mejor combinación de solución inicial, estrategia de generación de vecindad y largo de Lista Tabú, cabe preguntarse si es posible mejorar los resultados obtenidos.

7.4.1. Estrategia de Diversificación

Según lo planificado en el capítulo 6.3.6, se ha implementado una técnica de diversificación que involucra la combinación de un número de iteraciones sin mejoras en la función objetivo para gatillar la fase de diversificación y un número de iteraciones para perturbar la solución con otro método de vecindad. Tomando en consideración lo anterior, la Tabla N°12 muestra los principales resultados. En ella, el porcentaje de mejora representa la ganancia de la estrategia con respecto a la función objetivo que se obtendría si es que no se aplicara ningún tipo de diversificación. Finalmente, los parámetros considerados son los mismos descritos en la Tabla N°10, pero con un tiempo de ejecución de 30 minutos.

Tabla N°12

Función Objetivo Bajo Estrategia Diversificación: Distintas Combinaciones

Método	Iteraciones sin Cambio - Iteraciones Diversificando					
	50 - 20		30 - 10		20 - 5	
Perturbación		Mejora (%)		Mejora (%)		Mejora (%)
GPI	0,07534545	0,491	0,07524992	0,618	0,07569323	0,032
TPF	0,07653170	-1,075	0,07583945	-0,161	0,07549879	0,289
GTPF	0,07638628	-0,883	0,07672334	-1,328	0,07648222	-1,010
GIM	0,07649542	-1,027	0,07435231	1,803	0,07721945	-1,984

Fuente: Elaboración Propia

Las pruebas experimentales permiten extraer varias conclusiones. En primer lugar, las mejoras observadas (si es que se producen) son pequeñas, lo que habla de una contribución más bien marginal de la técnica.

Por otra parte, pareciera ser que existen varias combinaciones que finalmente llevan a peores resultados en relación a escenarios en que no se utiliza la técnica de diversificación. Esto último puede deberse a que, o bien las técnicas de perturbación dirigen la búsqueda hacia regiones de peor calidad o se abandona muy pronto la región que se está explorando (y que puede contener mejores soluciones) para diversificar.

Ahora bien, si se tratara del último caso entonces debería observarse una mejora en las soluciones a medida que aumenta el número de iteraciones sin mejorar la función objetivo antes de pasar a la etapa de diversificación. Sin embargo, la Tabla N°11 no muestra dicha tendencia.

Por lo tanto, se debe concluir que varios de los métodos de perturbación tienden a dirigir la búsqueda hacia regiones inferiores. La única excepción pareciera ser la aplicación del método *General Pairwise Interchange* (GPI), pues independientemente de la combinación entre el número de iteraciones sin cambio antes de gatillar la diversificación y el número de iteraciones diversificando, es posible observar mejoras en la función objetivo (por pequeñas que éstas sean).

Finalmente, aunque el mejor resultado se alcanza con el método de perturbación GIM, esto no parece constituir una tendencia clara y más bien puede responder a características puntuales de la instancia considerada, pues las variaciones del resultado en función del número de iteraciones sin cambio e iteraciones diversificando son grandes. Es por ello que se descarta esta alternativa como aplicable en términos promedios a distintas instancias del problema y se escoge como técnica de diversificación aquella que involucra la perturbación de la solución por GPI. Adicionalmente, en función de los datos de la Tabla N°12, se fija el número de iteraciones sin mejora antes de comenzar la fase de diversificación en 30 iteraciones y en 10 el número de iteraciones durante las cuales se perturba la solución.

7.4.2. Estrategia de Intensificación

La Tabla N°13 muestra el desempeño de la estrategia de intensificación propuesta en el capítulo 6.3.7. Cabe destacar que los resultados han sido calculados incorporando la estrategia de diversificación, con los parámetros calibrados en el capítulo anterior. De esta forma se busca estudiar la contribución final efectiva de la intensificación a la heurística. El resto de los parámetros de la Búsqueda Tabú son idénticos a los mencionados en la Tabla N°10.

Tabla N°13

Función Objetivo Bajo Estrategia Intensificación

Número de Soluciones Elite	Función Objetivo (F.O.)	Mejora (%)	Tiempo Resolución (seg)	Incremento (%)
10	0,07524849	0,00189767	1901,867	5,659
25	0,07493348	0,42051807	2477,227	37,624
50	0,07493348	0,42051807	4450,191	147,233
100	0,07493348	0,42051807	12285,558	582,531

Fuente: Elaboración Propia

Ahora bien, debido a que la intensificación ha sido incorporada como un método de post-optimización (posterior al proceso de Búsqueda Tabú propiamente tal), se muestra también en la tabla el incremento en el tiempo de ejecución del algoritmo producto de la fase de intensificación.

Los resultados anteriores permiten confirmar que, al igual que en el caso de la diversificación, la contribución de esta estrategia sobre la calidad de la solución final es más bien marginal. Por otra parte, aún cuando es posible observar una tendencia a mejorar la calidad de la solución a medida que se incrementa el número de soluciones *elite*, estas mejoras son cada vez menos importantes y más difíciles de obtener desde el punto de vista del tiempo de ejecución. Es más, para la instancia específica analizada no fue posible obtener mejoras en la función objetivo después de considerar más de 25 soluciones *elite*.

Luego, tomando en cuenta el *trade-off* existente entre el valor de la función objetivo y el tiempo de resolución, se concluye que la mejor relación costo-beneficio se da considerando un conjunto de soluciones *elite* de un tamaño igual a veinticinco (25).

7.5. Resumen Parámetros Finales

A modo de resumen, se presentan en la Tabla N°14 los parámetros definitivos que han sido calibrados para la Búsqueda Tabú. Cabe destacar que todo lo anterior ha sido considerando el caso de la Planta N°1.

Tabla N°14

Parámetros Finales Calibrados Búsqueda Tabú

Largo Lista Tabú	20% Tamaño Vecindad
Tiempo Ejecución*	30 min
Solución Inicial	<i>Weighted Shortest Processing Time</i>
Generación Vecindad	<i>Adjacent Pairwise Interchange</i>
Método Diversificación	<i>General Pairwise Interchange</i>
Número Soluciones <i>Elite</i>	25

* : Sin incluir fase de Intensificación

Fuente: Elaboración Propia

Para el caso de la Planta N°2, se recuerda que el algoritmo de Búsqueda Tabú es idéntico al planteado para la Planta N°1. La única diferencia es que las restricciones adicionales que considera este problema particular, son incorporadas una vez finalizado el algoritmo propiamente tal, forzando la secuencia a satisfacerlas post-optimización.

Luego, dado que el proceso de Búsqueda Tabú en ambos casos está diseñado para el mismo problema (mismas restricciones) se toma como supuesto de aquí en adelante que los parámetros calibrados para la Planta N°1 son igualmente válidos para el caso de la Planta N°2, y por lo tanto se encuentran resumidos en la Tabla N°14.

8. Resultados

Una vez calibrados los parámetros finales a considerar en el algoritmo de Búsqueda Tabú propuesto, es posible analizar el aporte real de esta heurística para mejorar la calidad de servicio y la eficiencia productiva en comparación a la situación actual.

8.1. Caso Planta N°1

8.1.1. Desempeño para Pequeñas Instancias

Previo a estudiar el impacto de la heurística propuesta sobre instancias reales del problema, parece interesante observar qué sucede para instancias reducidas del mismo, donde los valores de la solución óptima pueden ser calculados mediante el modelo lineal propuesto en el capítulo 5.1. La Tabla N°15 muestra una comparación entre las soluciones encontradas por el modelamiento matemático y aquellas calculadas mediante la Búsqueda Tabú. Cabe destacar que con la intención de mantener cierta “proporcionalidad” entre el tamaño de la instancia y el tiempo de ejecución del algoritmo, éste último se ha fijado en 1 minuto (sin considerar la fase de intensificación).

Tabla N°15

Búsqueda Tabú en Instancias Pequeñas: Planta N°1

Número de Trabajos	$\alpha - \beta$	Modelo Exacto	Búsqueda Tabú	GAP (% al Óptimo)
6	0,00 - 1,00	0,00024971	0,00024971	0,00
	0,25 - 0,75	0,00594904	0,00594904	0,00
	0,50 - 0,50	0,01066762	0,01066762	0,00
	0,75 - 0,25	0,01414080	0,01414080	0,00
	1,00 - 0,00	0,01734133	0,01734133	0,00
8	0,00 - 1,00	0,00024971	0,00024971	0,00
	0,25 - 0,75	0,00697275	0,00711404	2,03
	0,50 - 0,50	0,01213073	0,01240338	2,25
	0,75 - 0,25	0,01613876	0,01615704	0,11
	1,00 - 0,00	0,01988632	0,01991070	0,12
10	0,00 - 1,00	0,00024971	0,00024971	0,00
	0,25 - 0,75	0,00844441	0,00855272	1,28
	0,50 - 0,50	0,01489940	0,01523933	2,28
	0,75 - 0,25	0,01985890	0,01999062	0,66
	1,00 - 0,00	0,02456628	0,02474191	0,71
12	0,00 - 1,00	0,00024971	0,00024971	0,00
	0,25 - 0,75	0,01011744	0,01025620	1,37
	0,50 - 0,50	0,01836396	0,01883858	2,58
	0,75 - 0,25	0,02410659	0,02423831	0,55
	1,00 - 0,00	0,02984435	0,03001998	0,59
13	0,00 - 1,00	0,00024971	0,00024971	0,00
	0,25 - 0,75	0,01125535	0,01128345	0,25
	0,50 - 0,50	0,01905293	0,01970348	3,41
	0,75 - 0,25	0,02652186	0,02739005	3,27
	1,00 - 0,00	0,03286678	0,03287915	0,04

Fuente: Elaboración Propia

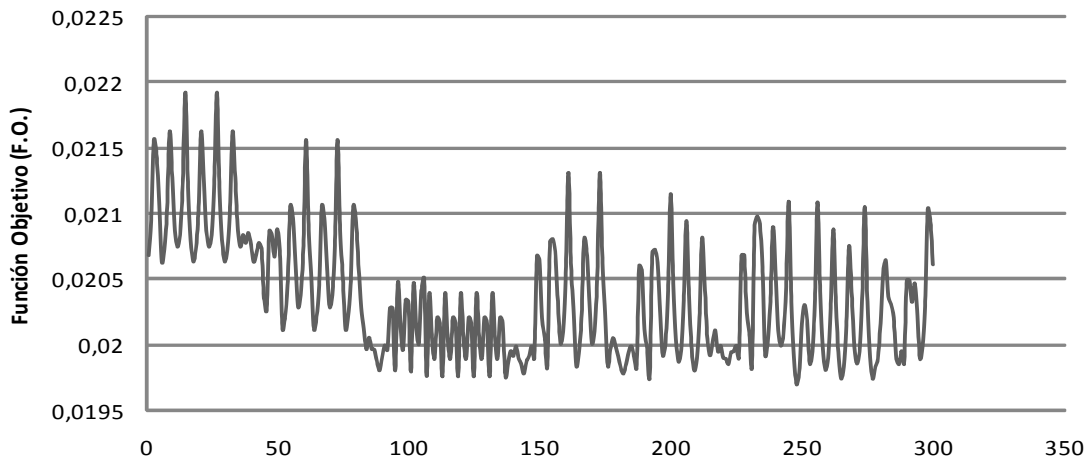
La tabla muestra que, probablemente, el algoritmo desarrollado permite alcanzar los valores óptimos para instancias menores o iguales a seis trabajos mientras que para instancias mayores se producen desviaciones.

Por otra parte, los GAP's más grandes se producen cuando la relación α sobre β se hace más cercana a uno. Esto último no es de extrañar, pues ya se había hecho notar - en el capítulo 5.4.5- que el problema se hace más difícil y complejo cuando lo anterior sucede. Y esta fue la principal razón para realizar la calibración del algoritmo tomando $\alpha = 0,5$ y $\beta = 0,5$. En caso contrario, probablemente se habrían observado desviaciones aún mayores.

En conclusión, de los resultados anteriores se desprende que si bien las desviaciones del valor óptimo no son tan grandes, tampoco son despreciables. Y más importante aún, éstas tienden a crecer a medida que el número de trabajos a secuenciar aumenta.

Luego, a fin de entender el por qué de esta situación, sólo queda concentrarse en el análisis mismo del proceso de búsqueda. Para tales efectos, se presenta en la Figura N°07 un gráfico que describe la evolución de la heurística en función del número de iteraciones, como una forma de visualizar la elección de soluciones realizadas por *Tabu Search*.

Figura N°07
Comportamiento Búsqueda Tabú: Planta N°1, 13 Trabajos



Fuente: Elaboración Propia

Cabe destacar que la confección del gráfico fue realizada en base a los resultados obtenidos para el caso que presenta una mayor desviación del óptimo en la Tabla N°15: 13 trabajos a secuenciar bajo una configuración $\alpha = 0,5$ y $\beta = 0,5$.

De la figura se desprende que, como ya se había hecho notar, las mayores ganancias se obtienen durante las primeras iteraciones. Y a medida que pasa el tiempo, las reducciones en la función objetivo son cada vez menores.

Por otra parte, no es posible observar la ocurrencia de ciclos, lo cual da cuenta de una buena elección para el tamaño de la Lista Tabú durante la fase de calibración. Se hace notar aquí también que otras pruebas conducidas permiten llegar -salvo unas pocas excepciones- a la misma conclusión, independientemente del tamaño del problema, el tiempo de ejecución y la elección de los parámetros α y β .

Adicionalmente, es posible constatar que el algoritmo parece ser capaz de escapar rápidamente de regiones de inferior calidad, lo que da cuenta de una estrategia efectiva de diversificación.

En conclusión, la elección de parámetros realizada durante la calibración del algoritmo (capítulo 7) parece ser razonable a la luz de los resultados empíricos. Luego, las desviaciones del óptimo observadas tal vez pueden responder a otras fuentes externas de error, que no han sido consideradas en el presente proyecto. Además, debe recordarse que por tratarse de una solución heurística, no es posible asegurar que efectivamente se alcanzarán los valores óptimos.

De esta forma, es posible suponer que para instancias de mayor tamaño (200 trabajos aproximadamente) pueden producirse diferencias importantes entre la solución óptima y aquella entregada por la heurística.

Pese a lo anterior, las pruebas para instancias reales permitieron obtener ganancias importantes, como se verá en el siguiente capítulo. Luego, se puede suponer que aún cuando las soluciones generadas por la Búsqueda Tabú parecen distar de las óptimas, los resultados obtenidos son suficientemente buenos -especialmente como primera aproximación- para aplicaciones prácticas.

8.1.2. Situación Actual versus Solución Propuesta

La Tabla N°16 presenta una comparación de los indicadores de desempeño para el caso de la Planta N°1. En ella no sólo se han incluido los datos bajo la situación actual, sino que también se han agregado los resultados del algoritmo de Búsqueda Tabú calibrado en el capítulo 7 bajo distintos escenarios para los valores de α y β . Los cálculos han sido realizados tomando un mes cualquiera (distinto de aquel utilizado para la calibración) como instancia de validación.

Tabla N°16

Situación Actual vs Solución Propuesta: Planta N°1

Indicador Desempeño	Situación Actual	$(\alpha - \beta)$				
		0,00 - 1,00	0,25 - 0,75	0,50 - 0,50	0,75-0,25	1,00 - 0,00
Determinación Programación	4,5 hrs.	41,37 min.	40,61 min.	41,20 min.	41,35 min.	41,42 min.
Tiempo Total Producción	24 días	31,41 días	23,03 días	23,10 días	22,50 días	21,72 días
Trabajos con Retraso	15,00%	7,69%	5,13%	11,54%	10,26%	10,26%
Trabajos con Retraso (Ponderado)	20,00%	14,55%	6,84%	19,98%	15,57%	15,19%
Retraso Promedio por Trabajo	1 día	0,09 días	0,24 días	0,49 días	0,72 días	0,94 días
Retraso Promedio por Trabajo (Ponderado)	1 día	0,15 días	0,31 días	0,59 días	1,25 días	1,37 días

Fuente: Elaboración Propia

El primer elemento que destaca de la Tabla N°16 es la significativa disminución en el tiempo empleado para determinar la programación, que representa una baja del 85% aproximadamente. Esto último se traduce principalmente en que la persona actual encargada de realizar la programación de operaciones -que es un recurso valioso- queda libre para dedicarse a otras actividades del Departamento de Logística.

Por otra parte, se observa que mientras mayor sea la importancia relativa que se le asigna a la eficiencia productiva (medida por el parámetro α), más pequeño es el tiempo de ciclo o el “tiempo total de producción”. Y esto es consistente con la formulación y el desarrollo planteado en el presente proyecto, lo que ayuda a validar los resultados. Sin embargo, se observa que es cada vez más difícil disminuir el tiempo total de producción, aún cuando aumente el valor de α a uno. Esto último debe responder necesariamente al hecho de que existe una capacidad productiva fija para la planta, que obviamente no es infinita.

Cabe destacar solamente que los valores obtenidos para el “tiempo total de producción” son sistemáticamente mejores que para la situación actual salvo en el caso $\alpha = 0$ y $\beta = 1$. Sin embargo, esto último no es de extrañar pues al darle completa importancia a la calidad de servicio, no se toma ninguna consideración en la reducción de *setup* ni en los tiempos de producción.

Ahora bien, en lo que respecta al porcentaje de trabajos que presentan retraso (tanto en su versión común como ponderada), no es posible ver ninguna tendencia clara respecto de su comportamiento, aún cuando los resultados son de mejor calidad a los de la situación actual para todos los escenarios analizados. Esto puede parecer algo inconsistente en principio, sin embargo, la explicación parece radicar en los dos últimos indicadores de la tabla.

En efecto, aún cuando la evolución del porcentaje de “trabajos con retraso” es poco predecible, tanto la medida de “retraso promedio por trabajo” como “retraso promedio por trabajo ponderado” tienden a disminuir a medida que se le da mayor importancia relativa a la calidad de servicio (medida por el parámetro β). En otras palabras, si bien pueden producirse mayor cantidad de trabajos atrasados, el foco del algoritmo está en reducir el “monto” de esos retrasos. Y esto último es consistente con el tratamiento dado al problema, lo que permite reafirmar el buen comportamiento de la heurística propuesta bajo los distintos escenarios analizados.

En conclusión, los resultados obtenidos al aplicar el algoritmo de Búsqueda Tabú desarrollado son sistemáticamente mejores a los que presenta la situación actual, independientemente del escenario considerado. Con ello entonces, se cumple el objetivo general del presente proyecto de título.

8.2. Caso Planta N°2

8.2.1. Desempeño para Pequeñas Instancias

La Tabla N°17 muestra una comparación entre los resultados del algoritmo de Búsqueda Tabú y las soluciones óptimas alcanzadas mediante el modelamiento matemático del capítulo 5.3, para pequeñas instancias del problema. Al igual que en el caso de la Planta N°1, el tiempo de resolución ha sido fijado en un minuto (sin considerar la fase de intensificación).

Tabla N°17

Búsqueda Tabú en Instancias Pequeñas: Planta N°2

Número de Trabajos	$\alpha - \beta$	Modelo Exacto	Búsqueda Tabú	GAP (% al Óptimo)
6	0,00 - 1,00	0,00327250	0,00336540	2,84
	0,25 - 0,75	0,01143900	0,01188727	3,92
	0,50 - 0,50	0,01960550	0,01980913	1,04
	0,75 - 0,25	0,02643707	0,02652798	0,34
	1,00 - 0,00	0,03229469	0,03240546	0,34
8	0,00 - 1,00	0,00771957	0,00773612	0,21
	0,25 - 0,75	0,02044839	0,02094807	2,44
	0,50 - 0,50	0,03317721	0,03386003	2,06
	0,75 - 0,25	0,04423146	0,04434504	0,26
	1,00 - 0,00	0,05360756	0,05372654	0,22
9	0,00 - 1,00	0,00771957	0,00773612	0,21
	0,25 - 0,75	0,02237578	0,02276935	1,76
	0,50 - 0,50	0,03703199	0,03748487	1,22
	0,75 - 0,25	0,05020279	0,05026153	0,12
	1,00 - 0,00	0,06067215	0,06078705	0,19
10	0,00 - 1,00	0,00771957	0,00773612	0,21
	0,25 - 0,75	0,02403010	0,02470510	2,81
	0,50 - 0,50	0,04034063	0,04090576	1,40
	0,75 - 0,25	0,05529438	0,05545851	0,30
	1,00 - 0,00	0,06694269	0,06736651	0,63

Fuente: Elaboración Propia

En este caso, es posible observar desviaciones del valor óptimo de similar magnitud que en el caso de la Planta N°1, pero con mayor regularidad. Sin embargo, este hecho ya podía ser anticipado debido al diseño de la heurística, tal y como se indicó en el capítulo 6.3.8.

Por otra parte, las mayores desviaciones tienden a observarse nuevamente cuando la relación α sobre β se hace más cercana a la unidad. Y la explicación es exactamente igual al caso anterior: el problema parece ser más difícil cuando la condición anterior se satisface.

Otro elemento importante que se debe mencionar es que los GAP's se mantienen relativamente estables para todas las instancias consideradas. Sin embargo, los tamaños del problema analizados son muy pequeños como para suponer que esta situación se mantiene para una instancia real (100 trabajos aproximadamente).

Pese a lo anterior, las ganancias observadas en instancias reales -como se verá en siguiente capítulo- parecen ser suficientemente buenas como para cumplir con los objetivos del presente trabajo de memoria.

8.2.2. Situación Actual versus Solución Propuesta

Para el caso de la Planta N°2 se ha procedido de igual manera que en el caso de la Planta N°1. La Tabla N°18 presenta el resumen de indicadores de desempeño bajo distintos escenarios para los parámetros α y β . Aquí también se ha utilizado un mes cualquiera (agosto 2008) como instancia de validación.

Tabla N°18

Situación Actual vs Solución Propuesta: Planta N°2

Indicador Desempeño	Situación Actual	$(\alpha - \beta)$				
		0,00 - 1,00	0,25 - 0,75	0,50 - 0,50	0,75-0,25	1,00 - 0,00
Determinación Programación	3 hrs.	40,02 min.	40,03 min.	39,76 min.	39,96 min.	39,91 min.
Tiempo Total Producción	28,62 días	27,06 días	25,63 días	25,13 días	25,12 días	25,09 días
Trabajos con Retraso	43,84%	47,95%	46,58%	39,73%	39,73%	39,73%
Trabajos con Retraso (Ponderado)	57,97%	69,66%	68,66%	62,35%	51,51%	52,91%
Retraso Promedio por Trabajo	5,38 días	2,93 días	3,41 días	3,60 días	4,48 días	4,16 días
Retraso Promedio por Trabajo (Ponderado)	6,69 días	2,57 días	2,93 días	3,64 días	4,79 días	4,93 días

Fuente: Elaboración Propia

En este caso también se observa una dramática disminución en el “tiempo de determinación de la programación” (78% en promedio). La conclusión por tanto es idéntica a la de la Planta N°1: se libera un recurso valioso, como es el encargado de la programación.

En lo que respecta al “tiempo total de producción” es posible constatar que los valores obtenidos son sistemáticamente inferiores a los de la situación actual, independientemente del escenario considerado. Adicionalmente, a medida que crece la importancia de la eficiencia productiva (aumenta α), el tiempo de ciclo se reduce, lo que es un resultado deseable y consistente con la formulación planteada.

Por otra parte, los indicadores de “trabajos con retraso” y “trabajos con retraso ponderado” disminuyen en la medida que crece la importancia relativa de la eficiencia productiva. Esto último parece ser un resultado contradictorio, pero como se mencionó a lo largo del análisis de de la Planta N°1, el énfasis de la Búsqueda Tabú está en reducir el tiempo de retraso por sobre la cantidad de trabajos con retraso. Y los dos últimos indicadores dan cuenta de lo anterior pues, a medida que se incrementa la importancia relativa de la calidad de servicio (β crece), disminuye tanto el “retraso promedio” como el “retraso promedio ponderado por trabajo”. Y esto es un resultado consistente con el desarrollo del algoritmo, lo que sumado a la disminución del tiempo de ciclo en función de α , permite validar el buen comportamiento de la Búsqueda Tabú para el sistema de laminación de la Planta N°2.

8.3. Valoración de las Mejoras Obtenidas

De la Tabla N°16 y la Tabla N°18 se puede calcular, considerando los valores promedio por fila, las ganancias reales asociadas directamente al algoritmo heurístico que ha sido desarrollado a lo largo del presente proyecto.

En este sentido, se pueden observar disminuciones en los tiempos de producción del 5% y 10% en promedio para cada planta respectivamente. Del mismo modo, los tiempos de retraso (ponderados) se ven reducidos en un 26,6% para la Planta N°1 y un 43,62% en la Planta N°2.

Sin embargo, si se analiza el mejor escenario en cada caso, es posible constatar ganancias en los tiempos de producción del 9,5% para la Planta N°1 y 12,33% para la Planta N°2. Por otra parte, en lo que respecta a la disminución de los retrasos (ponderados), es posible alcanzar reducciones del 85% y del 61,58% para cada planta respectivamente.

Por otro lado, el algoritmo se comporta adecuadamente en función de los parámetros α y β . Luego, se ha podido desarrollar una herramienta que posibilita la incorporación de un análisis de escenarios; por lo que la programación de operaciones se puede realizar de forma mucho más flexible, de acuerdo a las necesidades de producción que predominen en cada momento.

Otro elemento que debe mencionarse es que los resultados obtenidos parecen mostrar que la aplicación del algoritmo al caso de la Planta N°2 genera mayores beneficios que para el caso de la Planta N°1. Esto último se debe posiblemente al hecho de que la Planta N°1 tiene un *mix* de productos equivalente a casi el doble del de la Planta N°2 y, como ya ha sido señalado, la calidad de las soluciones entregada por la heurística disminuye a medida que se incrementa el número de productos.

Por otra parte, si bien a estas alturas es posible poner fin al trabajo de título -pues se consigue cumplir con todos los objetivos propuestos al inicio de la investigación-, resulta interesante estudiar el impacto económico sobre la empresa al hacer uso de la solución aquí desarrollada.

Pues bien, se debe señalar en primer lugar que existen dos fuentes importantes de mejoras: la disminución en los tiempos de producción y el aumento en la calidad de servicio.

Respecto de la reducción en el tiempo de ciclo es posible hablar de un ahorro de aproximadamente 105 mil toneladas anuales de inventario en proceso⁸ (palanquillas), lo que a la larga se traduce en un valor creado del orden de los \$1.500 millones al año. El razonamiento empleado y detalle del cálculo pueden ser encontrados en el Anexo E.

Finalmente, el Anexo F explica en detalle una estimación simple que permite cuantificar las mejoras obtenidas en la calidad de servicio. Para ello se toma como supuesto que una mejor calidad de servicio se traduce en un ingreso anticipado de los flujos económicos. Las estimaciones permiten hablar de un valor creado por un monto equivalente al 0,24% del Valor Actual Neto (VAN) de los ingresos anuales por ventas generados por la compañía, que por razones de confidencialidad no es posible revelar en el presente informe.

⁸ Inventario que ya ha sido parcialmente procesado y está esperando otra operación.

9. Conclusiones

El problema de programación de operaciones para las actividades de laminación en una compañía siderúrgica ha sido analizado en el presente trabajo de título y una solución heurística del tipo *Tabu Search* ha sido propuesta.

Dicha solución ha permitido dar cumplimiento a los objetivos planteados al inicio de la investigación: desarrollar una solución que permita decidir la secuencia en que se ejecutarán los trabajos, minimizando los tiempos de producción y los retrasos. En este sentido, se incorporado también la flexibilidad que permita analizar las soluciones bajo distintos escenarios (necesidades de planta).

Es así como es posible observar reducciones en los tiempos de producción del 5% en promedio para la Planta N°1 y del 10% en promedio para la Planta N°2. Del mismo modo, los tiempos de retraso (ponderados) han sido disminuidos en promedio en un 26,6% en el caso de la Planta N°1 y un 43,62% en el caso de la Planta N°2.

Por otra parte, ha sido posible también disminuir de forma drástica el tiempo empleado para determinar la programación (82% en promedio para ambas plantas), con lo que se consigue la liberación de la persona encargada para dedicarse a otras actividades.

Las pruebas experimentales han demostrado que las soluciones encontradas -si bien se traducen en mejoras importantes- pueden distar del óptimo. Ahora bien, en lo que al presente proyecto se refiere, los resultados son considerados suficientemente buenos para aplicaciones prácticas, especialmente como una primera aproximación de solución al problema.

Sin embargo, resulta evidente que se generan posibilidades de mejoras futuras de los resultados, por lo que se recomienda dar continuidad a la investigación. Algunos puntos que podrían abordarse en el futuro son: considerar otras alternativas de diversificación, como por ejemplo la penalización de los movimientos a partir de una matriz de frecuencias; y evaluar la incorporación de la estrategia de intensificación en el algoritmo mismo y no sólo como un procedimiento de post-optimización.

Adicionalmente, se ha conseguido incorporar una metodología formal que permita justificar las decisiones tomadas en torno a la programación de operaciones en las actividades de laminación. Y aún cuando la solución prototipo ha sido desarrollada en un lenguaje de baja eficiencia, el diseño del algoritmo y sus características son igualmente válidos. Sin embargo, se recomienda considerar la implementación en un lenguaje de programación con mejores características de memoria (como C++ por ejemplo), pues la cantidad de soluciones visitadas durante el mismo tiempo de ejecución podría ser incrementada por un factor de 20, de acuerdo a las estimaciones de Bettinger *et al.* (2005).

Por otra parte, la solución aquí desarrollada genera como resultado la fecha exacta para la cual cada trabajo es finalizado (obviamente buscando reducir los retrasos respecto de la fecha sugerida que es entregada como *input*). Luego, es perfectamente posible permitir que el algoritmo entregue la fecha para aquellos productos del tipo *Make-to-Order* (MTO), simplemente entregando como *input* una fecha suficientemente grande como para dejar inactiva la restricción. De esta forma, se sugiere emplear la solución propuesta para fijar las fechas de entrega de este tipo de productos, en lugar de dejarlo arbitrariamente en manos del Departamento de Ventas, pues la heurística cuenta con la ventaja de considerar la carga de trabajo mensual y las actividades de planta.

Finalmente, el algoritmo presentado puede ser igualmente aplicable a cualquier otro tipo de problema combinatorial que involucre, o pueda ser visto como, la programación de n –pedidos en una máquina (por ejemplo, determinar el orden en que deben ser colocadas las capas aislantes de un material compuesto a fin de maximizar su valor de aislamiento total). En particular, es posible utilizar la heurística aquí propuesta para determinar la programación de operaciones en otras de las actividades de planta de la misma compañía, por ejemplo, a nivel de la Acería. La única salvedad es considerar el valor adecuado para los distintos parámetros. En este sentido, la metodología aquí propuesta permite el diseño, calibración y evaluación de cualquier heurística del tipo *Tabu Search*.

10. Referencias Bibliográficas

- [1] ARMENTANO, V., MAZZINI, R. 2000. A Genetic Algorithm for Scheduling on a Single Machine with Setup Times and Due Dates. *Production Planning and Control* 11(8): 713-720.
- [2] BEAUSOLEIL, R. 2000. Intensification and Diversification Strategies with Tabu Search: One-Machine Problem with Weighted Tardiness Objective. *Springer* 1973:52-62.
- [3] BETTINGER, P., BOSTON, K., KIM, Y., ZHU, J. 2007. Landscape-level Optimization using Tabu Search and Stand Density-related Forest Management Prescriptions. *European Journal of Operational Research* 176(2):1265-1282.
- [4] BUTCHERS, E., RYAN, D. 2001. Optimized Crew Scheduling at Air New Zealand. *Institute for Operations Research and the Management Sciences* 31(1): 30-56.
- [5] DIRECMIN. Crisis Financiera Obliga a Vale a Reducir Producción y Gastos. 2008. [en línea] <http://www.direcmin.com/Noticias/Whats-Up-Empresas/Crisis-Financiera-Obliga-a-Vale-a-Reducir-Produccion-y-Gastos.html> [consulta: 27 noviembre 2008].
- [6] EREN, T., GÜNER, E. 2006. A bicriteria scheduling with sequence-dependent setup times. *Applied Mathematics and Computation* 179(1): 378-385.
- [7] GLOVER, F. 1986. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research* 13: 533-549.
- [8] GLOVER, F. 1989. Tabu Search-Part I. *ORSA Journal of Computing* 1:190-206.
- [9] GLOVER, F. 1990. Tabu Search- Part II. *ORSA Journal on Computing* 2: 4-32.
- [10] GLOVER, F. 1993. A user's guide to tabu search. *Annals of Operations Research*.41:3-28.
- [11] GLOVER, F., LAGUNA, M. 1997. *Tabu Search*. Boston, Kluwer Academic Publishers. 382p.
- [12] GLOVER, F., MELIÁN, B. 2003. Búsqueda Tabú. *Revista Iberoamericana de Inteligencia Artificial* 19:29-48.
- [13] GRAVEL, M., PRICE, W. L., GAGNE, C. 2000. Scheduling in a Alcan Aluminium Factory. *International Journal of Production Research* 38:3031-3041.
- [14] GUPTA, A. K., SIVAKUMAR, A. I. 2005. Single Machine Scheduling with Multiple Objectives in Semiconductor Manufacturing. *The International Journal of Advanced Manufacturing Technology* 26(9-10):950-958.

- [15] GUPTA, S., SMITH, J. 2006. Algorithms for single machine total tardiness scheduling with sequence dependent setups. *European Journal of Operational Research* 175(2): 722-739.
- [16] LAGUNA, M., MARTÍ, R. 1997. A Greedy Randomized Adaptive Search Procedure for the 2-Partition Problem. *Operations research* 42(4):677-687.
- [17] LAGUNA, M., MARTÍ, R. 1999. GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. *Institute for Operations Research and the Management Sciences* 11:44-52
- [18] LEE, Y. H., BHASKARAN, K., PINEDO, M. 1997. A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions* 29:45-52.
- [19] LIN, S., YING, K. 2006. Solving single machine total weighted tardiness problems with sequence dependent setup times. *International Journal of Advanced Manufacturing Technology* 34(11-12):1183-1190.
- [20] MORTON, T. E., PENTICO D. W. 1993. *Heuristic Scheduling Systems: With Applications to Production Systems and Project Management*. John Wiley & Sons, Inc., New York, 720p.
- [21] ORELLANA, G. 2008. Crisis financiera provoca la suspensión de proyectos por US\$ 2362 millones en Chile. [en línea] *El Mercurio Online*. 20 de noviembre, 2008. http://www.mercurio.cl/2008/11/20/economia_y_negocios/economia_y_negocios/noticias/E59844CB-1029-4D32-9E92-797B835D6F9A.htm?id={E59844CB-1029-4D32-9E92-797B835D6F9A} [consulta: 24 de noviembre, 2008]
- [22] PINEDO, M. 1995. *Scheduling: Theory, Algorithms and Systems*. Prentice-Hall.
- [23] PLA, L. 1986. *Análisis Multivariado: Método de Componentes Principales*. Monografía N°27, Serie Matemática. O.E.A., Washington D.C.
- [24] REEVES, C.R. (Ed.) 1993. *Modern heuristic techniques for combinatorial problems*. Ed 1. John Wiley & Sons, Inc., New York, 320p.
- [25] SUPERINTENDENCIA DE BANCOS E INSTITUCIONES FINANCIERAS DE CHILE. 2008. Tasa de Interés Promedio del Sistema Bancario [en línea] Santiago, Chile. <http://www.sbif.cl> [consulta: 03 diciembre 2008].
- [26] VALENTE J., ALVES, R. 2008. Beam search algorithms for the single machine total weighted tardiness scheduling problem with sequence-dependent setups. *Computers & Operations Research* 35(7): 2388-2405.

11. Anexos

Anexo A. Normalización de Variables para la Función Objetivo

Debido a que las dos medidas de desempeño propuestas para el modelo de programación lineal son de magnitudes distintas, se hace necesario normalizarlas a un mismo rango de valores a fin de mantener la consistencia entre los pesos que se asignan a cada una.

Para cumplir con el objetivo antes propuesto, se aplicó la técnica que se describe a continuación. En primer lugar, sea una variable x cualquiera con valores definidos en un intervalo:

$$x \in [x_{min}, x_{max}] \quad (1)$$

Se busca entonces normalizar dicha variable, es decir, convertir sus valores a un nuevo intervalo definido por:

$$x^* \in [x_{min}^*, x_{max}^*] \quad (2)$$

Para resolver este problema, se puede definir el siguiente sistema de ecuaciones lineales:

$$k * x_{min} + p = x_{min}^* \quad (3)$$

$$k * x_{max} + p = x_{max}^* \quad (4)$$

Donde k y p se denominan factores de escala. La solución al sistema de ecuaciones planteado es:

$$k = \frac{x_{max}^* - x_{min}^*}{x_{max} - x_{min}} \quad (5)$$

$$p = \frac{x_{min}^*(x_{min} - x_{max}^*)}{x_{max} - x_{min}} \quad (6)$$

Por lo que la variable x normalizada se puede expresar como:

$$x^* = k * x + p = \frac{x_{max}^*(x - x_{min}) + x_{min}^*(x_{min} - x)}{x_{max} - x_{min}} \quad (7)$$

La forma más común de normalizar -y que es la se empleará aquí- consiste en definir $x^* \in [0,1]$. Reemplazando entonces en (5) y (6) se obtiene:

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (8)$$

Finalmente, es necesario considerar que el tiempo de producción asociado a un trabajo (C_j) es una variable que puede tomar valores entre el menor de los tiempos de procesamiento y un mes. Por otro lado, la variable de tiempo de retraso de un trabajo cualquiera (T_j) se mueve en un intervalo entre cero y un mes. En otras palabras, se tiene:

$$C_j \in [\min\{p_j\}, 43200] \quad (9)$$

$$T_j \in [0, 43200] \quad (10)$$

Donde los valores han sido expresados, en una elección arbitraria, en minutos. Luego, reemplazando en (8) se obtiene entonces:

$$C_j^* = \frac{C_j - \min\{p_j\}}{43200 - \min\{p_j\}} \quad (11)$$

$$T_j^* = \frac{T_j}{43200} \quad (12)$$

De esta forma, las medidas de desempeño definidas para el modelo de programación lineal propuesto serán normalizadas antes de ser evaluadas en la función objetivo. De esta forma se asegura mantener la proporcionalidad definida para cada medida de desempeño mediante los coeficientes α y β .

Anexo B. Breve Análisis de Sensibilidad Planta N°1

Rendimiento Promedio

En el presente proyecto de título se ha asumido que el rendimiento promedio por producto puede ser estimado razonablemente mediante el rendimiento promedio de la familia a la que éste pertenece. Debido a que se trata de una aproximación, resulta relevante estudiar el comportamiento de la solución respecto de pequeños cambios que puedan producirse en los rendimientos promedio.

En base a lo anterior, se propone un análisis del peor escenario. En otras palabras, se estudiarán las variaciones que pudieran producirse en la función objetivo suponiendo que todos los rendimientos promedio considerados tienen una desviación del 5%. Luego, se repetirá el procedimiento pero considerando que todos los rendimientos promedio tienen una variación del -5%. Los modelos han sido compilados y ejecutados en un entorno Windows, utilizando el software OPL Studio 3.7 con el solver Cplex en un PC Pentium 4 de 2.4GHz con 512 MB RAM. La Tabla N°19 resume los resultados encontrados.

Tabla N°19

Función Objetivo frente a Cambios en Rendimiento Promedio por Familia: Planta N°1

$(\alpha - \beta)$	Estimación por Familia	Cambio Rendimiento por Familia			
		-5%	Diferencia(%)	5%	Diferencia(%)
0,00-1,00	0,0002	0,0003	25,00%	0,0002	-20,83%
0,25-0,75	0,0112	0,0118	5,36%	0,0110	-1,79%
0,50-0,50	0,0191	0,0207	8,38%	0,0187	-2,09%
0,75-0,25	0,0265	0,0278	4,91%	0,0257	-3,02%
1,00-0,00	0,0328	0,0344	4,88%	0,0314	-4,27%

Fuente: Elaboración Propia

Cabe destacar que se ha utilizado como instancia de prueba aquella que considera un total de 13 trabajos a secuenciar. Esta elección se debe a que dicha instancia es la más grande para el problema que se puede resolver de manera exacta.

Los resultados muestran que las variaciones en la función objetivo son generalmente menores en relación a los cambios en el rendimiento promedio. La única excepción la constituye el caso $\alpha = 0$ y $\beta = 1$. Sin embargo, esto último podría encontrar explicación en el pequeño valor de la solución óptima.

Por otra parte, se observa que la función objetivo crece a medida que disminuye el rendimiento promedio y disminuye en el caso contrario, independientemente del valor que se asigne a los pesos relativos α y β . Este resultado no es de extrañar pues si el rendimiento baja, los tiempos de procesamiento suben y, en consecuencia, los tiempos de producción también aumentan repercutiendo directamente en la función objetivo.

Finalmente, si se consideran los bajos cambios en el valor de la función objetivo y además se recuerda que se ha considerado un análisis del peor caso, entonces se debe concluir que la función objetivo es poco sensible frente a cambios en el rendimiento promedio.

Es así como, a falta de información más precisa, es posible aproximar el rendimiento promedio por producto por el rendimiento de la familia a la que éste pertenece sin que esto genere grandes errores en la función objetivo.

Tiempos de Cambio

El análisis de sensibilidad respecto de variaciones en los tiempos de cambio (*setup*) entre un producto y otro se muestra en la Tabla N°20.

Tabla N°20

Función Objetivo frente a Cambios en Tiempos de Setup por Familia: Planta N°1

$(\alpha - \beta)$	Estimación por Familia	Cambio Tiempo Setup por Familia			
		-5%	Diferencia(%)	5%	Diferencia(%)
0,00-1,00	0,0002	0,0002	4,76%	0,0002	-4,76%
0,25-0,75	0,0112	0,0119	6,25%	0,0113	0,89%
0,50-0,50	0,0191	0,0204	6,81%	0,0197	3,14%
0,75-0,25	0,0265	0,0272	2,64%	0,027	1,89%
1,00-0,00	0,0328	0,0327	-0,30%	0,033	0,61%

Fuente: Elaboración Propia

En este caso también se ha considerado un análisis del peor escenario. Es decir, se han hecho los cálculos considerando primero que todos los tiempos de *setup* aumentan un 5% y luego todos disminuyen en la misma proporción. Del mismo modo, y para conservar la integridad del análisis, se ha utilizado la misma instancia que en el caso anterior y el mismo *software* de resolución.

Los resultados de la Tabla N°20 muestran que las desviaciones en la función objetivo tienden a ser menores al cambio en los tiempos de *setup*. Sin embargo, es posible observar que las mayores diferencias se dan cuando los tiempos de cambio son reducidos en un 5%. Esto significa que las variaciones en la función objetivo tienden a ser mayores cuando los tiempos de *setup* disminuyen.

Luego, se puede concluir que la función objetivo es algo sensible frente a variaciones en los tiempos de cambio, especialmente si éstas son negativas. Pese a lo anterior, debe hacerse hincapié nuevamente en que se ha considerado un análisis del peor caso. De esta forma, considerando la salvedad anterior, es posible aproximar los tiempos de *setup* entre un producto y otro como el tiempo de cambio entre las familias a las que éstos pertenecen.

Anexo C. Breve Análisis de Sensibilidad Planta N°2

Tiempos de Cambio

En el presente proyecto se ha asumido que el tiempo de cambio entre un producto y otro, para el caso de la Planta N°2, puede ser estimado razonablemente como el tiempo de cambio entre las semifamilias a las que estos productos pertenecen. Se requiere por tanto un análisis de sensibilidad para analizar la validez de este supuesto.

La Tabla N°21 muestra los resultados bajo un análisis del peor escenario. Es decir, se ha considerado que todos los tiempos de *setup* aumentan un 5% y luego todos disminuyen en el mismo 5%. Los modelos han sido compilados y ejecutados en un entorno Windows, utilizando el software OPL Studio 3.7 con el solver Cplex en un PC Pentium 4 de 2.4GHz con 512 MB RAM. Adicionalmente, se ha utilizado como instancia de prueba aquella que considera un total de 10 trabajos debido a que dicha instancia es la más grande para el problema que se puede resolver de manera exacta.

Tabla N°21

Función Objetivo frente a Cambios en Tiempos de Setup por Familia: Planta N°2

$(\alpha - \beta)$	Estimación por Familia	Cambio Tiempo Setup por Familia			
		-5%	Diferencia(%)	5%	Diferencia(%)
0,00-1,00	0,0077	0,0077	0,13%	0,0077	0,26%
0,25-0,75	0,0240	0,0242	0,83%	0,0240	0,00%
0,50-0,50	0,0403	0,0404	0,17%	0,0404	0,25%
0,75-0,25	0,0552	0,0554	0,36%	0,0554	0,36%
1,00-0,00	0,0669	0,0669	0,00%	0,0671	0,30%

Fuente: Elaboración Propia

De los resultados se desprende que las variaciones en la función objetivo son considerablemente inferiores en relación a los cambios en los tiempos de cambio. Y esta condición es independiente de la importancia relativa asignada a α y β .

Es más, las diferencias son mucho menores a aquellas observadas para el caso de la Planta N°1. Esto último puede deberse a que en la Planta N°1 se trabaja con información a nivel agregado, mientras que en la Planta N°2 a nivel semiagregado, lo que permite tener una mayor precisión en los resultados.

Finalmente, si se consideran los cambios marginales en el valor de la función objetivo, se concluye entonces que la solución es prácticamente invariante frente a variaciones en los tiempos de *setup*, por lo que es posible aproximar el tiempo de cambio entre un producto y otro como aquel existente entre las semifamilias a las que estos productos pertenecen.

Anexo D. Calibración Parámetros Apparent Tardiness Cost with Setups

La heurística desarrollada por Lee, Bhaskaran y Pinedo (1997) requiere necesariamente la calibración previa de las constantes k_1 y k_2 . Afortunadamente, los mismos autores proponen una táctica sencilla para poder estimarlas rápidamente y que es la que se empleará en el presente proyecto.

Según Pinedo, los dos parámetros de escala pueden ser estimados como función de otros tres factores:

- a) Factor de Estrechez en Fecha de Entrega (*due-date tightness factor*). Indica el grado de holgura en las fechas de entrega y es definido como:

$$\tau = 1 - \frac{\bar{d}}{C_{max}} \quad (1)$$

Donde

\bar{d} = Promedio de las fechas de entrega.

C_{max} = Tiempo de Ciclo.

- b) Factor de Rango en Fecha de Entrega (*due-date range factor*). Indica la amplitud en el rango de fechas de entrega y está definido como:

$$R = \frac{d_{max} - d_{min}}{C_{max}} \quad (2)$$

Donde

d_{max} = Fecha de entrega más lejana.

d_{min} = Fecha de entrega más cercana.

C_{max} = Tiempo de Ciclo.

- c) Factor de Severidad en Tiempos de Cambio (*setup-time severity factor*). Éste es definido como:

$$\eta = \frac{\bar{s}}{\bar{p}} \quad (3)$$

Donde

\bar{s} = Tiempo promedio de setup.

\bar{p} = Tiempo promedio de procesamiento.

Debido a que el valor del tiempo de ciclo depende de la secuencia en que se procesan los distintos trabajos, éste debe necesariamente ser estimado. Una forma simple y rápida de hacerlo es utilizando la siguiente expresión:

$$\hat{C}_{max} = \sum_{j=1}^n p_j + n\bar{s} \quad (4)$$

Con ello, es posible calcular tanto el factor de estrechez en fecha de entrega como el factor de rango en fecha de entrega. Finalmente, los parámetros k_1 y k_2 pueden ser estimados de acuerdo a la siguiente regla empírica propuesta por Pinedo:

$$k_1 = 4.5 + R \quad \text{si } R \leq 0.5 \quad (5.1)$$

$$k_1 = 6 - 2R \quad \text{si } R > 0.5 \quad (5.2)$$

Y para la selección de la constante k_2 se puede utilizar la siguiente regla:

$$k_2 = \tau / (2\sqrt{\eta}) \quad (6)$$

Anexo E. Estimación Beneficios Económicos por Reducción Tiempo Ciclo

Las reducciones alcanzadas por la solución de tipo heurístico propuesta sobre los tiempos de ciclo de producción, pueden ser cuantificadas en términos monetarios. Para ello, se requiere introducir previamente algunos conceptos importantes.

En términos generales, y como se detalló en el capítulo 2.1, el Sistema de Laminación estudiado puede ser entendido como sigue. Los productos semiterminados -denominados palanquillas- ingresan al sistema a una determinada tasa, mientras que el laminador transforma esos semiproductos en productos finales a una tasa que tiende a ser menor a la de ingreso; razón por la cual se forma un “cuello de botella”. Por lo tanto, debido a que la tasa de entrada es mayor a la de salida, se produce una cola de productos semiterminados (WIP) esperando ser procesados por el laminador.

Pues bien, la situación antes descrita puede ser analizada desde la perspectiva de la Teoría de Colas. En este sentido, el área de laminación puede ser entendida como un sistema M/M/1 de acuerdo a la notación de Kendall.

Como consecuencia de lo anterior, se satisface la Ley de Little, que relaciona la tasa de llegada con la cantidad promedio de “entidades” y el tiempo promedio de espera en la cola del sistema:

$$L_q = \lambda * w_q \quad (1)$$

Donde

L_q = Número promedio de unidades en la cola (*Work in Process*)

λ = Tasa de entrada al sistema

w_q = Tiempo promedio de espera en la cola

Lo interesante aquí es que el tiempo promedio de espera puede ser estimado como el tiempo de ciclo de producción (26,31 días como promedio de la Planta N°1 y Planta N°2). Adicionalmente, medidas en terreno y el juicio experto del encargado de operaciones de la compañía, han determinado que el inventario en proceso (WIP) en promedio es de 3500 toneladas de palanquilla al día. Luego, de la ecuación (1), se obtiene que la tasa de llegada al sistema es $\lambda = 133,03$ [ton/día].

Ahora bien, suponiendo que la tasa de llegada al sistema permanece constante una vez incorporada la solución heurística desarrolla en el presente trabajo de título, entonces es posible calcular el inventario en proceso que se generaría producto de su implementación.

Bajo este nuevo escenario, el tiempo promedio de espera en la cola puede estimarse como 24,12 días. Este valor ha sido calculado como el promedio del tiempo de ciclo entre la Planta N°1 y la Planta N°2, considerando además una situación de equilibrio con $\alpha = 0,5$ y $\beta = 0,5$.

De esta forma, el WIP bajo este nuevo escenario queda como sigue:

$$L_q^* = \lambda * w_q^* \quad (2)$$

$$L_q^* = 133,03 \text{ [ton/día]} * 24,12 \text{ [día]} \quad (3)$$

$$L_q^* = 3208,67 \text{ [ton]} \quad (4)$$

Como consecuencia de lo anterior, se produce una disminución del inventario en proceso igual a:

$$\Delta L = L_q - L_q^* = 3500 \text{ [ton]} - 3208,67 \text{ [ton]} = 291,33 \text{ [ton]} \quad (4)$$

En otras palabras, en un año es posible obtener un ahorro de 105.000 toneladas aproximadamente. Y si se considera que una tonelada de palanquilla cuesta aproximadamente \$100.000, entonces es posible hablar de un ahorro del orden de 10.500 millones de pesos al año en material que ya no estará detenido en las actividades de planta.

Finalmente, a una tasa de interés real del 15,1%⁹ (tasa de interés promedio del sistema bancario para octubre del 2008 de acuerdo a cifras publicadas por la Superintendencia de Bancos e Instituciones Financieras), se puede hablar de un valor creado por la solución propuesta al reducir los tiempos de ciclo de aproximadamente 1.500 millones de pesos al año.

⁹ SUPERINTENDENCIA DE BANCOS E INSTITUCIONES FINANCIERAS DE CHILE. 2008. Tasa de Interés Promedio del Sistema Bancario [en línea] Santiago, Chile. <http://www.sbif.cl> [consulta: 03 diciembre 2008].

Anexo F. Estimación Beneficios por Reducción Retrasos

La reducción de los retrasos respecto de las fechas de entrega se traduce directamente en un aumento de la credibilidad y confianza por parte del cliente. La estimación del valor creado como consecuencia de lo anterior es realmente compleja pues se deben considerar una gran cantidad de elementos.

Sin embargo, y dicho lo anterior, es posible calcular una estimación ingenua pero sencilla. Se toma como supuesto que en caso de aumentar los retrasos en las fechas de entrega, el cliente tiende a postergar la siguiente compra, lo que a su vez significa que el ingreso de los flujos también se posterga.

Ahora bien, como consecuencia del incremento en el cumplimiento, resulta lógico pensar que la postergación en la compra se reducirá. Sea una reducción pequeña, adelantando en un día la compra con respecto a la situación actual. Si además se supone una compra de carácter mensual, los flujos de ingresos por venta a lo largo de un año representan un Valor Actual Neto (VAN) que puede ser calculado como sigue:

$$VAN' = \frac{C}{(1+r)^{0,97}} + \frac{C}{(1+r)^{1,94}} + \frac{C}{(1+r)^{2,91}} + \frac{C}{(1+r)^{3,97}} + \dots + \frac{C}{(1+r)^{11,64}} \quad (1)$$

Mientras que en la situación actual, el VAN corresponde a:

$$VAN = \frac{C}{(1+r)^1} + \frac{C}{(1+r)^2} + \frac{C}{(1+r)^3} + \frac{C}{(1+r)^4} + \dots + \frac{C}{(1+r)^{12}} \quad (2)$$

Donde

C = Monto de la Orden Total (Considerando todo el conjunto de clientes).

r = Tasa de interés (expresada en términos mensuales).

Cabe destacar que tanto el pedido del cliente como la tasa de interés se han supuesto invariantes para todo el horizonte de evaluación considerado.

Si adicionalmente se considera una tasa de interés mensual real del 1,26% ¹⁰(tasa de interés promedio del sistema bancario para octubre del 2008 de acuerdo a cifras publicadas por la Superintendencia de Bancos e Instituciones Financieras), entonces se puede calcular cada valor:

$$VAN = 11,07236 * C \quad (3)$$

$$VAN' = 11,09882 * C \quad (4)$$

En consecuencia es posible observar un aumento del valor creado por la compañía al reducir los retrasos, producto de la aplicación de la solución desarrollada, equivalente a:

$$\Delta\% = \frac{VAN' - VAN}{VAN} = \frac{11,09882C - 11,07236C}{11,07236C} = 0,239\% \quad (5)$$

¹⁰ SUPERINTENDENCIA DE BANCOS E INSTITUCIONES FINANCIERAS DE CHILE. 2008. Tasa de Interés Promedio del Sistema Bancario [en línea] Santiago, Chile. <http://www.sbif.cl> [consulta: 03 diciembre 2008].

Ahora bien, por razones de confidencialidad no es posible contar con información respecto del monto del VAN generado por los ingresos de la compañía. Es por ello, que sólo queda propuesto el valor de la mejora en términos porcentuales.

