

***Tesis de Licenciatura en Ciencias de la  
Computación***

# ***Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search***

***(A Real Case of the Job Shop Scheduling  
Problem with Tabu Search)***



***Departamento de Computación***

*Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

**Facultad de Ciencias Exactas y Naturales**

**Universidad de Buenos Aires**

**Luciano Fiorini (760/98)  
2007**

**Julio de**

**Directora: Irene Loiseau**

## **Agradecimientos**

En lo personal quisiera agradecer a toda mi familia por el apoyo incondicional en todos estos años y en especial a mis padres por la oportunidad que me dieron de estudiar y ser un profesional. Quiero mencionar especialmente a mi esposa Andrea por su apoyo, paciencia y ayuda en todo este tiempo juntos. No quiero olvidarme de los amigos y conocidos que de una u otra manera siempre me expresaron su apoyo y me brindaron su ayuda.

En lo profesional quisiera agradecer, en primer lugar, a mi directora de tesis Irene, quien me ayudo a comprender la importancia y el alcance de esta última etapa de la carrera, su apoyo fue muy necesario para poder completarla. Agradezco a mis compañeros de trabajo, en especial a Miguel, Pablo y Federico, que me brindaron su ayuda desinteresadamente desde su conocimiento y experiencia así como su apoyo personal para culminar esta etapa. No quiero olvidarme de mencionar a mis compañeros de carrera, en especial a aquellos que se transformaron en amigos, por la ayuda, compañerismo y conocimientos compartidos en tantos años de estudio.

Quiero mencionar la ayuda aportada por la empresa siderúrgica a modo de ejemplo que utilice en este trabajo que me brindo información y herramientas fundamentales para poder desarrollar este trabajo.

## Índice

<u>TESIS DE LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN.....</u>	<u>1</u>
<u>UN CASO REAL DE SECUENCIACIÓN DE TAREAS DEPENDIENTES CON TABU SEARCH.....</u>	<u>1</u>
<u>(A REAL CASE OF THE JOB SHOP SCHEDULING PROBLEM WITH TABU SEARCH).....</u>	<u>1</u>
<u>DEPARTAMENTO DE COMPUTACIÓN.....</u>	<u>1</u>
<u>FACULTAD DE CIENCIAS EXACTAS Y NATURALES.....</u>	<u>2</u>
<u>UNIVERSIDAD DE BUENOS AIRES.....</u>	<u>2</u>
<u>LUCIANO FIORINI (760/98).....</u>	<u>JULIO DE 2007.....</u>
<u>DIRECTORA: IRENE LOISEAU.....</u>	<u>2</u>
<u>AGRADECIMIENTOS.....</u>	<u>3</u>
<u>ÍNDICE.....</u>	<u>4</u>
<u>I - RESUMEN.....</u>	<u>5</u>
<u>II - INTRODUCCIÓN.....</u>	<u>6</u>
<u>III - NOMENCLATURA.....</u>	<u>8</u>
<u>IV - DESCRIPCIÓN DEL PROBLEMA.....</u>	<u>10</u>
<u>V - MODELO PROPUESTO.....</u>	<u>13</u>
<u>VI - TABU SEARCH.....</u>	<u>20</u>
<u>VII - EXPERIMENTOS.....</u>	<u>25</u>

# Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

## I - Resumen

En este trabajo se plantea una solución algorítmica a un caso particular del problema de la secuenciación de tareas dependientes, en inglés *Job Shop Scheduling Problem (JSSP)*, que consiste en planificar un conjunto de tareas en un grupo de máquinas capaces de procesarlas. Las restricciones básicas del problema son que cada máquina no puede procesar más de una tarea a la vez y que cada tarea tiene una secuencia ordenada de máquinas por las que tiene que ser procesada. El objetivo del problema es secuenciar todas las tareas de forma tal que el tiempo necesario para completarlas a todas sea mínimo. En este trabajo se estudio en detalle el problema particular de secuenciar la producción de una empresa que realiza un proceso productivo de transformación, es decir, un proceso productivo que mediante transformaciones sucesivas de una materia prima dada, obtiene el producto final esperado.

Este trabajo comienza con un análisis detallado de la problemática de una empresa particular. Se utilizó como ejemplo una empresa siderúrgica cuyo producto principal son los tubos de acero con y sin costura. El problema fue desglosado para poder plantear soluciones parciales, las cuales se integraron alcanzando la solución general. Fue necesario determinar un algoritmo sobre el cual basar la solución y Tabu Search fue la heurística elegida luego de una búsqueda y análisis de trabajos existentes sobre el tema. Se implementó una pequeña parte de esta solución y se probó sobre datos concretos de la empresa siderúrgica mencionada. Los resultados obtenidos fueron buenos y alentadores para continuar con el estudio y desarrollo de este trabajo.

*This work establishes an algorithmic solution to a particular case of the Job Shop Scheduling Problem (JSSP), which consists in scheduling a set of jobs in a group of machines capable of processing them. The basic restrictions of the problem are that each machine cannot process more than one job at the time and each job has a ordered sequence of machines in which it has to be processed. The objective of the problem is to schedule all the jobs so that the necessary time to complete all of them is minimized thoroughly the particular matter of scheduling the production of a company which has a productive process of transformation. That is to say, a productive process, which by means of successive transformations of a given raw material obtains the expected final product.*

*This work begins with a detailed analysis of the issues in a particular company. As example an iron and steel company was taken, whose main product is steel seamless and welded pipes. The problem was broken down to propose partial solutions, which, integrated, provided a general solution. It was necessary to determine an algorithm on which to base the solution and Tabu Search was the heuristic chosen after a search and analysis of existing efforts on the topic. A small part of this solution was implemented and tested on concrete data of the iron and steel company mentioned before. The obtained results were good, and encouraging to follow through the study and development of this labor.*

## II - Introducción

Toda empresa competitiva necesita una planificación óptima de su producción para cumplir con los compromisos acordados con los clientes y para poder elegir los nuevos negocios a futuro. Este trabajo se basa en la búsqueda de una solución algorítmica para dicho problema en el caso particular de las empresas que realizan un proceso productivo de transformación. Se trata de un proceso productivo que mediante transformaciones sucesivas de una materia prima dada, obtiene el producto final esperado. Esta aclaración es importante para que el lector entienda el alcance de este trabajo y sus limitaciones. No es tema de este trabajo el análisis de la aplicabilidad de la solución planteada a otros tipos de procesos productivos.

El JSSP (*Job Shop Scheduling Problem*) consiste en planificar un conjunto de tareas en un grupo de máquinas capaces de procesarlas. En la mayoría de los casos el JSSP considera como restricciones que cada tarea tiene una lista ordenada de máquinas por las que tiene que ser procesada y cada tarea solo puede ser procesada por una máquina a la vez y ninguna máquina puede tener asignada más de una tarea al mismo tiempo. El JSSP tiene como objetivo final planificar todas las tareas de forma tal que el tiempo comprendido entre el inicio de la primera tarea y el fin de la última sea el mínimo. Es un claro problema de optimización combinatoria. Algunos consideran que una persona puede hacer el esfuerzo de secuenciar cuatro tareas en tres máquinas pero intentar con diez tareas alcanza niveles de experto. Existe mucha bibliografía sobre el JSSP que en su forma general es un problema NP-Hard (Garey y Johnson [5]). Algunos autores mencionan que es “notoriously hard”.

Este trabajo considera como ejemplo de un proceso productivo de transformación el problema de planificar la producción de una empresa siderúrgica. Este problema particular se puede modelar como un caso general de JSSP donde las restricciones son las siguientes:

1. Una tarea tiene un grafo dirigido que representa las máquinas por las que tiene que ser procesada. Este grafo considera las distintas alternativas de listas ordenadas de máquinas por las que puede ser procesada la tarea.
2. Una tarea puede estar asignada a más de una máquina a la vez. Podría haber una máquina procesando la salida de otra (*máquinas trabajando en línea*) o incluso podría haber dos máquinas procesando la misma tarea en paralelo.
3. Una máquina no puede procesar más de una tarea a la vez.

Si consideramos como JSSP general a aquel donde las tareas tienen una sola lista ordenada de máquinas y donde una tarea no puede ser asignada a más de una máquina a la vez, es claro observar que este problema tiene un espectro mayor de soluciones posibles.

En la búsqueda y análisis de trabajos existentes sobre JSSP nos encontramos con casos muy interesantes como el de Dai y Weiss [4], donde consideran los trabajos de una máquina como flujos homogéneos en lugar de tareas discretas con distintos tiempos de procesamiento. Asumen que las tareas fluyen a un valor proporcional al “esfuerzo” de la máquina, si este es máximo entonces es igual a  $\frac{1}{p}$  donde  $p$  es el promedio de los tiempos de procesamiento de todas las tareas en esta máquina. Según indican, de esta forma logran que la máquina “cuello de botella” trabaje a máxima capacidad y las demás a una fracción de sus capacidades.

## *Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

Sería imposible recorrer todos los trabajos existentes sobre JSSP, pero no puede faltar una mención al respecto de algoritmos genéticos. Uno de los trabajos es el de Lin, Goodman y Punch [8] donde abordan esta técnica sobre el caso particular de la secuenciación de tareas que arriban continuamente a las máquinas. Un modelo determinista y uno estocástico fueron planteados para el problema de la secuenciación dinámica de producción y los resultados obtenidos indicaron una buena representación y robustez.

El trabajo de Adelson-Velsky y Levner [1] representa las dependencias entre las tareas con un grafo dirigido de peso en los arcos y nodos de condición “and/or”. Los arcos representan las operación mientras que los nodos representan el inicio y fin de de las operaciones. El peso en cada arco representa la duración de la operación y la condición de “and/or” en cada nodo determina las restricciones para cada operación. Existen trabajos basados en el algoritmo de Dijkstra pero tienen la limitación de no contemplar los casos en que existan ciclos de longitud cero. Este trabajo contempla estos casos pero como bien indican sus autores el algoritmo puede no encontrar una solución para ciertos casos.

A partir de estos y muchos otros trabajos consideramos que la técnica Tabu Search podía ser adecuada para ser aplicada en el modelo propuesto. Sus características de sencillez, eficacia y adaptabilidad la posicionan como una de las mejores en el área de los problemas de optimización combinatoria y la gran cantidad de trabajos y resultados exitosos así lo demuestran. Es así que se decidió adoptarla como la técnica sobre la cual basar toda la solución algorítmica propuesta en este trabajo.

A continuación se encuentra una nomenclatura detallada sobre la planificación de la producción para poder establecer un lenguaje común con el lector y así proseguir a la descripción minuciosa de la problemática en la empresa ejemplo. El desglosamiento del problema y las definiciones de las soluciones parciales en cada etapa muestran el desarrollo de la solución propuesta. El trabajo continúa con la integración de Tabu Search a todo el modelo, con los experimentos realizados al respecto y finaliza con las conclusiones correspondientes.

## III - Nomenclatura

A continuación se listan una serie de términos utilizados en el trabajo que son propios del lenguaje empleado en las empresas. A cada término se le adjunta una breve explicación para eliminar todo tipo de ambigüedad respecto a su comprensión. En la figura 1 se muestra un diagrama de entidad relación sobre todas las entidades definidas.

**Producto**, entidad física, real y concreta que permite representar cada uno de los elementos que una planta puede producir. Esta entidad tiene una serie de características y/o propiedades que lo describen unívocamente, determinando el proceso de producción requerido y pudiendo o no influir en las condiciones del mismo.

**Orden**, entidad que identifica un contrato de producción y venta, entre la empresa productora y un cliente, para producir una cantidad de piezas de un material específico dado (producto). Esta tiene entre otras las siguientes características:

1. *Cliente*, persona física o jurídica que contrata.
2. *Producto*, producto comprometido a entregar.
3. *Piezas*, cantidad del material a entregar.
4. *Fecha de Entrega*, fecha acordada para entregar las piezas de la orden.
5. *Fecha de Liberación*, fecha a partir de cuando se puede comenzar a procesar la orden. Siempre menor a Fecha de Entrega.
6. *Fecha de Inicialización*, fecha a partir de la cual se comenzó a procesar la orden.
7. *Fecha de Finalización*, fecha en la cual se termino de procesar toda la orden.
8. *Lead Time*, diferencia entre Fecha de Finalización y Fecha de Inicialización.
9. *Condiciones de entrega*, pago, precio, etc., entre muchas otras.

**Stock**, conjunto homogéneo de piezas, asociado siempre a una orden y que cuenta con las siguientes características:

1. *Orden*, orden a la que pertenece el stock.
2. *Estado de Elaboración*, estado de transformación alcanzado en el proceso.
3. *Ubicación*, lugar físico donde se encuentra en la planta.
1. *Piezas*, cantidad de piezas que lo conforman.

**Centro**, es un conjunto no vacío de máquinas. Las mismas están físicamente juntas o cercanas, y el conjunto puede ser visualizado como una única máquina que realiza una dada operación. Los centros pueden o no tener reglas de procesamiento. Se utilizará indistintamente el termino maquina y centro en este trabajo.

**Reglas**, conjunto de restricciones que condicionan la actividad de un centro. Estas restricciones hacen a impedimentos mecánicos, físicos o económicos de las máquinas que lo conforman, influyendo sobre la secuenciación de órdenes en el mismo.

**Setup**, tiempo de preparación de un centro para poder procesar un producto. Es el tiempo utilizado para calibrar las máquinas y ponerlas a punto antes de comenzar la producción de un nuevo producto. Su valor es función no sólo del producto por procesar, sino también del último procesado.



# Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

**Fábrica Nave (FN)**, área física dentro de la planta donde se puede almacenar material (stock). Las FN poseen una cota máxima de capacidad hasta donde poder llenarlas. Cada centro tiene un conjunto de FN de donde poder tomar material para procesar, y otro conjunto de FN de donde puede dejar material procesado. Los centros no pueden procesar material (deben parar) en caso de que al hacerlo, el stock luego debiera ser almacenado en una FN saturada.

**Estado**, código que identifica el estado de elaboración de un stock e indirectamente determina cual es el próximo centro/operación que debe procesar dicho material dependiendo de su CC (ciclo completo).

**Hoja de Ruta (HdR)**, secuencia de centros por los que tiene que pasar un material para obtener el estado de terminado para un producto determinado. El orden en la secuencia implica que el material procesado por un centro es el que va luego a ser procesado en el centro siguiente. La conexión entre dos centros de la HdR esta dado en que el primero produce un material en el mismo estado en que el siguiente centro lo procesará, existiendo una intersección entre el conjunto de posibles FN de salida del primer centro y el conjunto de FN de entrada del segundo centro. El primer centro de una HdR es el que procesa el producto en su estado inicial, y el último centro es el que produce el producto en estado de terminado.

**Delay**, tiempo necesario para que un stock producido por un centro pueda ser procesado por el centro siguiente en la HdR. Incluye tiempos de movimiento, de ensayos de laboratorio y algunos otros tecnológicos tales como el enfriamiento del material, por ejemplo.

**Productividad**, cantidad de piezas que puede procesar un centro por unidad de tiempo al realizar una operación determinada, en una Hdr y para un cierto estado de elaboración de entrada.

**Descarte**, porcentaje de piezas no recuperables (chatarra) que genera un centro en una Hdr y para un cierto estado de elaboración de entrada, al procesar una orden de un producto dado.

**Descarte diferencial**, porcentaje de piezas recuperables que genera un centro en una Hdr y para un cierto estado de elaboración de entrada, al procesar una orden de un producto dado. Estas piezas siguen por otra Hdr (Hdr de recuperación) que en algún punto se vuelve a unir a la Hdr original.

**Ciclo Completo (CC)**, conjunto de HdR para un producto determinado de una orden dada. Al CC también se lo conoce como el conjunto de alternativas de HdR para producir un cierto producto. Cada orden tiene su propio CC.

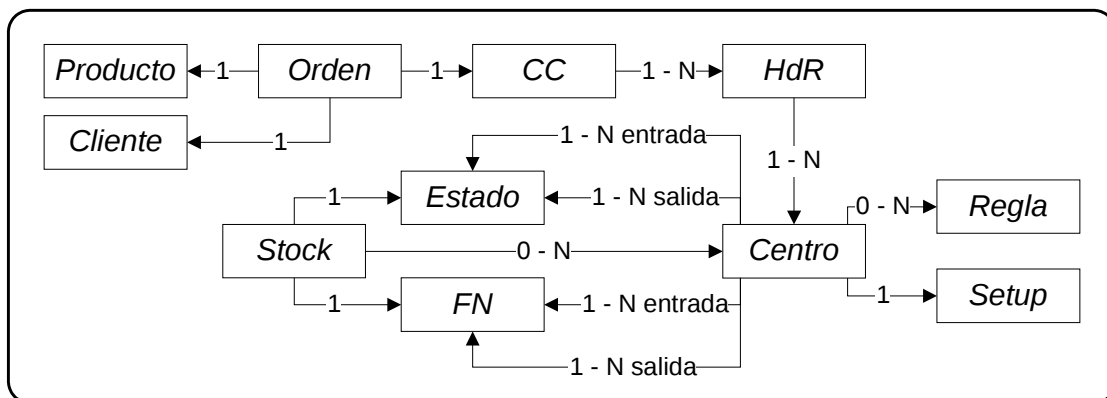


Figura 1

## IV - Descripción del problema

La planta trabaja mediante una metodología bien definida que sufre periódicos cambios como resultado del estudio permanente de posibles mejoras. A este estudio permanente se lo conoce como “mejora continua” y se basa en el constante análisis de la metodología actual y en el avance de la tecnología utilizada para definir los cambios hacia una mejor metodología de trabajo. Si a este desarrollo continuo le sumamos los cambios inesperados y excepciones, se hace muy difícil tener un modelo completo y detallado de toda la operatoria de la planta. Lo que sí podemos tener es un cierto modelo que en parámetros generales se ajuste muy bien a la realidad y que periódicamente vayamos adaptando a los cambios que surjan.

Una empresa tiene órdenes de producción que cumplir y éstas pueden clasificarse en dos grandes grupos, las firmes y las previsiones. Las firmes son negocios acordados con un cliente y las previsiones son negocios que están a punto de acordarse con un cliente o pueden ser una intención de venta de la empresa que ni si quiera tiene un cliente definido. Las previsiones tienen la característica de sufrir cambios (el tipo de producto, la cantidad, las fechas, etc.) durante su existencia. Una orden firme, si bien puede sufrir un cambio, es poco probable que lo haga porque requiere del aval del cliente para hacerlo y esto es mediante una nueva negociación. El conjunto total de firmes y previsiones es lo que se conoce como la *cartera* de la empresa. Los diferentes productos y cantidades de las órdenes se los considera el *mix* de la cartera. Es importante aclarar que en su gran mayoría las previsiones desaparecen de la cartera de la empresa para transformarse en firmes y el resto de las previsiones desaparecen porque dejaron de ser un posible negocio o dejaron de ser intenciones de la empresa. Las firmes desaparecen de la cartera solo cuando se las termina de procesar y el cliente las considera terminadas.

Una empresa no puede comenzar a producir una previsión porque esto implica riesgo económico y de calidad. Debido a los cambios que puede sufrir una previsión, podría suceder que parte de las tareas realizadas a una previsión fuera necesario realizarlas nuevamente. Incluso todas las tareas realizadas podrían no ser salvables y fuese necesario volver a comenzar. Por otro lado los clientes no concretan un negocio con la empresa hasta no estar seguros del producto y cantidades que quieren. Debido a estas características se da una cierta distribución de las previsiones y las firmes en la línea de tiempo que podemos visualizar en la figura 2. En la distribución mencionada es importante la cantidad de firmes en el corto plazo y las previsiones en el mediano y largo plazo. Es acá donde podemos observar claramente la diferencia entre programar y planificar la producción. Si bien en ambos casos el objetivo es determinar la mejor secuenciación de la cartera en cada centro de la empresa, la diferencia fundamental radica en el horizonte de tiempo contemplado y en el tipo de cartera utilizada.

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

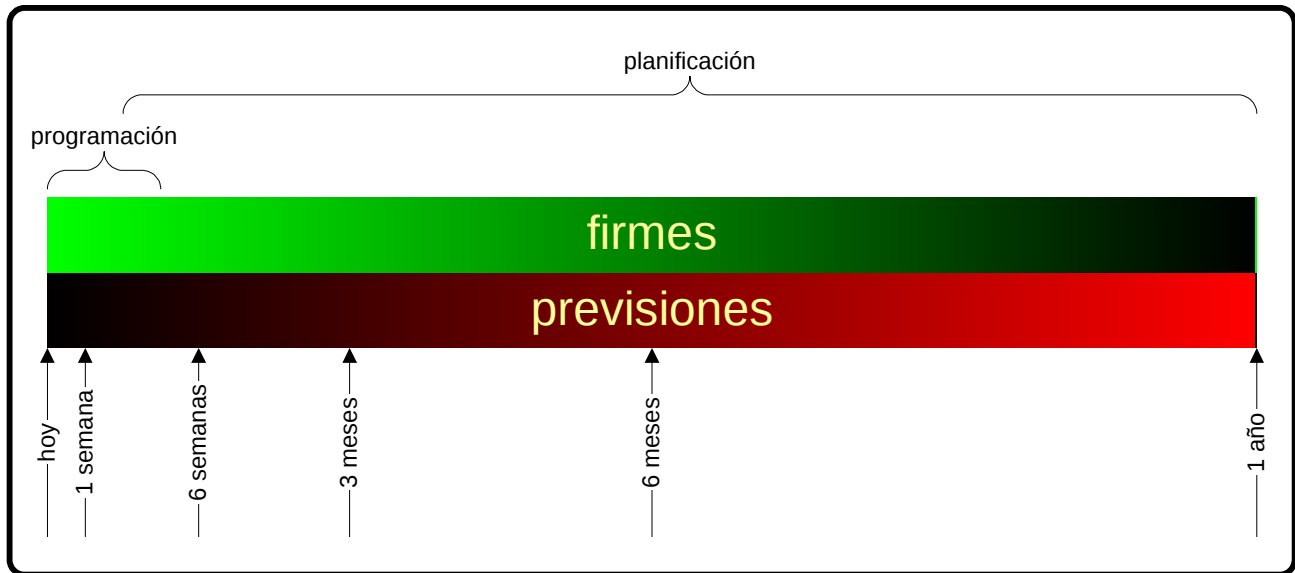


Figura 2

Esta diferencia hace que la programación y la planificación trabajen sobre distintas carteras, es decir, que trabajen con distinta variabilidad en la información que manipulan y entonces es distinto el grado de detalle utilizado en la secuenciación. Supongamos como ejemplo que en el día de hoy programamos la secuencia de un centro para toda la semana a partir de mañana. Entonces, necesitamos que la secuencia tenga un detalle de segundos porque estamos armando el programa de producción que la planta deberá ejecutar. En cambio si planificamos la secuencia de una semana para dentro de 6 meses en el mismo centro, este nivel de detalle no es necesario. Es posible que dentro de un mes la cartera de esa semana cambie, sean necesarias otras consideraciones y entonces la secuencia resulte absolutamente distinta. Podemos resumir este concepto de la siguiente manera, mientras la planificación tiene una visión “macro” y mensual de la secuenciación, la programación tiene una visión “micro” y diaria.

Las empresas generalmente congelan toda la información de sus plantas en un momento dado y trabajan sobre ella para generar una secuenciación completa de toda su cartera. La información congelada contiene el estado completo de la planta, lo sucedido desde el último congelamiento de información y las órdenes existentes. Desde el punto de vista de la programación se busca en la secuenciación de la cartera el poder indicarle a la planta el plan de producción para los próximos días para que ésta pueda prepararse y el poder controlar cualquier imprevisto que suceda. En cambio desde la planificación se busca visualizar a futuro la secuenciación para poder detectar problemas con anticipación, gestionar lo necesario para un mejor aprovechamiento del negocio y evaluar los posibles nuevos negocios.

En este escenario una empresa debe optimizar la secuenciación de su producción en cada uno de sus centros para todo el horizonte de tiempo de la mejor manera posible. Las premisas a tener en cuenta en esta labor son las siguientes:

1. **Cumplimiento**, debe cumplir con las fechas de todas las órdenes.
2. **Procesamiento**, debe respetar la secuencia de procesamiento de las Hdr de cada orden.
3. **Reglas**, no debe romper ninguna de las reglas de producción de los centros.
4. **Setups**, debe minimizar los setups de cambio de medida en cada uno de los centros.
5. **Stocks intermedios**, debe minimizar los stocks intermedios.
6. **Paradas**, no debe generar paradas por falta de material innecesarias en los centros.

## *Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

Las empresas desean tener una herramienta capaz de secuenciar en forma óptima su cartera para todo el horizonte de tiempo que desean visualizar. Esta herramienta debe respetar las premisas antes mencionadas, debe tener en cuenta el estado de la planta y considerar toda la cartera. Siempre existen imprevistos, cambios de último momento en la cartera o en el estado de la planta que deben ser reflejados en la herramienta. Estos cambios pueden ser órdenes que cambian condiciones, órdenes que desaparecen, órdenes nuevas, centros que se rompen, problemas gremiales, etc. Si estos cambios existen en el largo plazo, hay tiempo suficiente para controlarlos pero si se producen en el corto plazo este tiempo es casi nulo. Es por eso que esta herramienta debe tener un tiempo de respuesta breve para reflejar estos cambios, visualizar la nueva situación del negocio y así poder tomar las decisiones necesarias.

La empresa siderúrgica tomada como ejemplo cuenta con una cartera promedio de 6700 órdenes (70% firmes), unos 150 centros distintos y un horizonte de tiempo de alrededor de 1 año. Cada orden tiene un grafo dirigido que representa los centros por los que tiene que ser procesada que puede variar desde 1 hasta 74 nodos (13,25 nodos promedio). Considerando una distribución homogénea de las órdenes en los centros podemos considerar que existen unas 45 órdenes por centro, es decir, unas  $45!$  ( $1 \times 10^{56}$ ) soluciones posibles por centro.

## V - Modelo propuesto

En la figura 3 se muestra esquemáticamente como puede ser la HdR (hoja de ruta) de una orden con centros alternativos en alguna tarea. La elección de la HdR para una orden es cuestión del departamento de “Ingeniería de Producto” de la empresa que mediante ciertos criterios (calidad, tecnología, disponibilidad, etc.) determina la mejor HdR para cada orden, pero esto no es tema de este trabajo. Podemos destacar algunos detalles de este tipo de HdR como ser que en la tarea 3 existen 3 centros alternativos que pueden hacer la misma tarea. Esto no quiere decir que la empresa solo tenga 3 centros para esta tarea si no que la empresa decide utilizar solo estos 3 centros para hacer esta tarea para esta orden y puede ser que para otras ordenes utilice más o incluso menos centros. También podemos observar que a veces, la elección de una alternativa (por ejemplo el centro  $C_1$ ) implica la elección de la alternativa en la tarea siguiente (centro D) ya que según esta HdR solo existe un centro posible en la siguiente tarea. También podría darse el caso de que un centro tenga la tecnología suficiente para realizar la tarea que le corresponde y la que sigue como si fuera una sola tarea (centro  $C_2$  y  $C_3$ ). Es simple imaginar la gran cantidad de casos que se pueden dar pero esto no complica el problema ya que se desarrolló un modelo que se aísla de estos detalles.

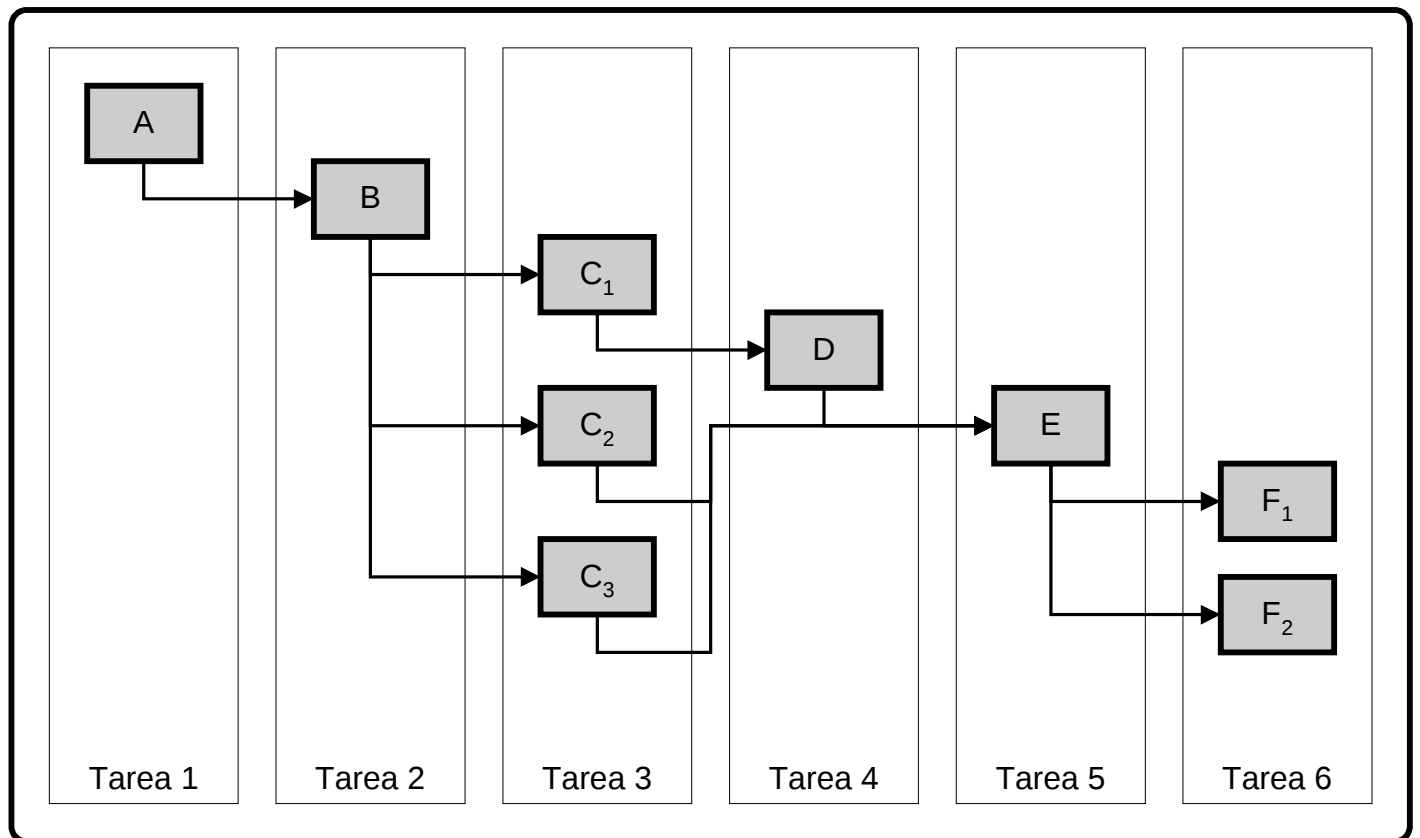


Figura 3

Supongamos que una orden se debe entregar en una fecha que corresponde al mes  $n$ , esto quiere decir que todas las piezas de la orden deben estar terminadas antes de dicha fecha. Considerando el ejemplo de la figura 3, decimos que la tarea 6 debe terminarse para la fecha de entrega pactada. Entonces, ¿cuándo deberíamos terminar las tareas anteriores para poder cumplir con toda la orden?

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

En la figura 4 se gráfica un diagrama de PERT para visualizar como se calcula la “fecha de entrega” para una tarea a partir de la fecha de entrega de la tarea posterior. Dada la fecha de entrega de una orden en un centro, se calcula la “fecha tardía” en este mismo centro al restarle el tiempo de producción de la orden en el centro. Al restarle el tiempo de espera (stock promedio) que en promedio espera la orden al pie del centro para ser procesada, obtenemos la “fecha de liberación” en este centro. Esta última fecha es cuando la orden deberá estar disponible en el centro para que en promedio se pueda cumplir con la fecha de entrega del centro. Si a esta fecha le restamos el tiempo necesario para trasladar la orden del centro anterior a este obtenemos la fecha de entrega de la orden en el centro anterior. Este calculo lo podemos aplicar de forma recursiva desde el último centro hasta el primero para obtener la fecha de entrega en cada centro de la HdR, partiendo de la consideración de que la fecha de entrega del último centro es igual a la fecha de entrega de la orden. Esto último es “proyectar” la fecha de entrega de la orden en toda su HdR.

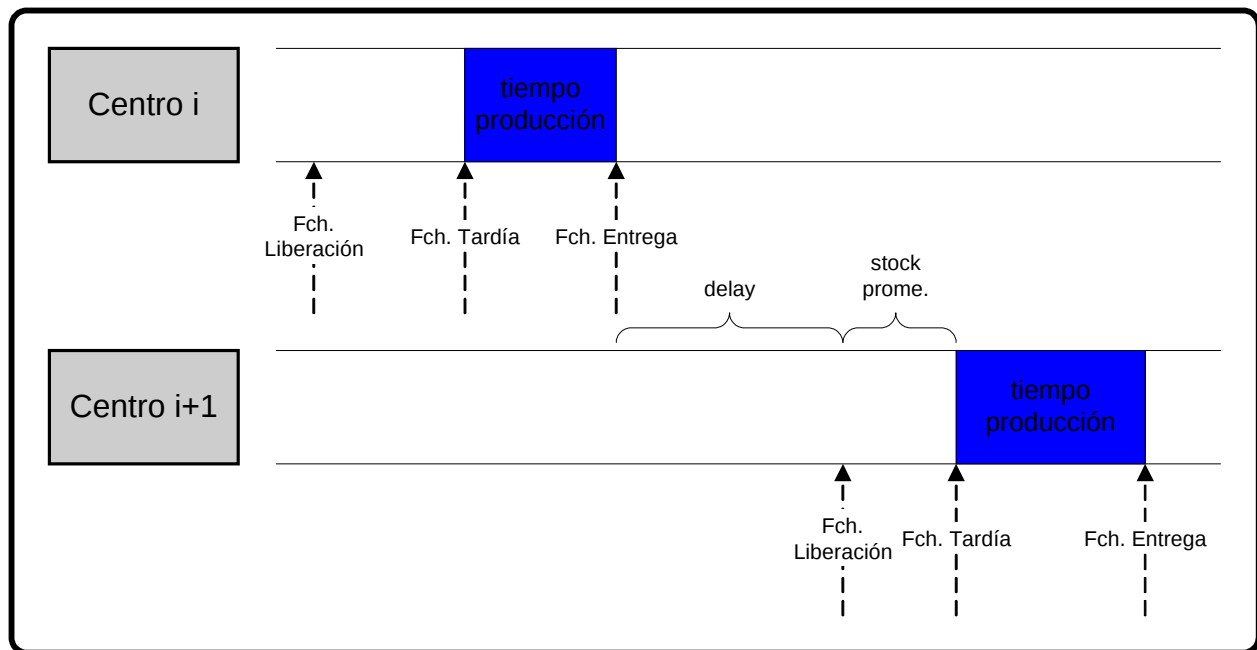


Figura 4

Cuando existe más de un centro en una tarea se debe tomar un criterio para determinar respecto a que fecha de liberación de los centros se calcula la fecha de entrega de la tarea anterior. Un posible criterio es considerar como fecha de liberación de una tarea a la más comprometida de todos sus centros. A partir de esta se calcula la fecha de entrega para cada uno de los centros de la tarea anterior aplicando el delay correspondiente entre los mismo. Este es el criterio que vamos a considerar en este trabajo. De esta forma podemos obtener para cada orden la fecha de entrega en cada uno de los centros de su HdR. Luego de calcular las fechas de cada HdR podemos determinar para cada centro de la empresa las órdenes que tiene que procesar en algún momento dado. Si a éstas órdenes las agrupamos por mes según la fecha de entrega obtenemos lo que llamamos “cartera mensual” o simplemente “cartera” de un centro.

El tiempo de producción en el cálculo de fechas de una orden está relacionado con la productividad en el centro y una cantidad de piezas determinada. La productividad es la cantidad de piezas que un centro puede procesar en un turno y cuando proyectamos la fecha de entrega de la orden desde el último centro lo hacemos junto al “saldo” de la orden. El saldo es la cantidad de piezas que faltan terminar de la orden.

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

Existen órdenes que en algún centro pueden tener la fecha de entrega en un mes y la fecha tardía en el mes anterior, es decir que están a “caballo” del mes. Estos casos se resuelven partiendo proporcionalmente el tiempo de producción de la orden para que esta tenga fechas de entrega en cada mes. Estas órdenes estarán en la cartera de más de un mes. Siguiendo con el ejemplo, la orden estará en la cartera del mes anterior con una fecha de entrega igual a la fecha cuando finaliza el mes y también estará en el mes siguiente con la fecha de entrega calculada en su momento. Las dos fechas de entrega se proyectarán sobre el resto de la HdR, aunque podríamos volver a juntarlas en caso de encontrarse nuevamente en un mismo mes. Existen más detalles como este en el cálculo de la cartera mensual de cada centro pero no vamos a detenernos en ellos ya que no hacen al desarrollo de este trabajo. Lo importante es que una vez salvados estos detalles en el cálculo, según el criterio de cada empresa, obtenemos las fechas de todos los centros de cada HdR.

En la tabla 1 resumimos las carteras mensuales de todos los centros (20 en este ejemplo) de una empresa como cantidad de turnos (turno = 8 horas) de producción en cada mes. Los valores en rojos indican que superan la cantidad de turnos libres del mes para ese centro, es decir que tenemos pensado para ese mes mayor producción que la capacidad del centro y eso se conoce como “overbooking”. Los turnos libres, indicados debajo de cada mes, se consideran como el tiempo calendario del mes menos los días feriados, menos los días que el centro no trabaja y menos el tiempo de setup promedio del centro para ese mes. Por simplificación, consideramos que todas los centros trabajan sin régimen de descanso y con 10% de setup promedio. A continuación se muestra el cálculo del tiempo libre para el mes de Mayo para un centro cualquiera:

$$\begin{aligned} \text{Tiempo Libre Mayo} &= ( 31 \text{ días calendario} - 2 \text{ días feriados} - 0 \text{ días de descanso} ) * ( -10\% \text{ setup} ) \\ &= ( 93t - 6t - 0t ) * 0.9 = 78.3t \end{aligned}$$

Tarea	Centro	Mes					
		Febrero (75.6)	Marzo (83.7)	Abril (70.2)	Mayo (78.3)	Junio (78.3)	Julio (81)
A	1	76.9	78.9	69.1	79.0	82.7	77.0
	6	76.5	83.9	69.8	76.8	77.0	83.2
	12	81.6	81.6	73.1	75.9	79.3	78.3
B	2	78.1	80.5	70.8	79.8	82.0	80.0
	18	77.1	85.7	69.1	76.7	75.2	77.3
C	3	83.3	84.6	67.2	76.4	80.0	83.9
	8	76.4	88.0	68.3	79.4	75.0	80.0
	11	74.9	82.2	73.0	73.6	74.6	78.8
	13	77.3	82.9	74.2	82.3	77.8	77.1
	16	78.8	87.3	69.8	78.1	81.8	80.1
D	4	79.7	84.0	65.9	81.6	81.5	76.4
	7	76.0	85.9	72.6	81.1	77.0	84.0
	9	79.8	87.0	72.1	75.1	79.9	82.4
E	5	79.0	83.0	69.7	76.5	73.7	81.7
	10	73.0	87.0	72.4	74.1	79.4	80.5
	15	76.8	88.4	70.0	76.7	78.3	80.9
	17	79.4	83.0	68.0	80.9	78.6	85.5
F	14	76.6	88.0	67.0	80.7	81.9	80.1
	19	74.0	80.2	75.0	75.8	77.1	80.6
	20	78.1	85.6	69.9	76.0	81.7	77.2

Tabla 1

## *Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

Cada uno de los números de la tabla anterior representa en turnos la cartera del centro para cada mes, es decir, una bolsa de órdenes para procesar en ese centro en cada mes sin un orden establecido. En este punto logramos dividir toda la cartera de la empresa en carteras mensuales por centro y pretendemos resolver cada una para integrarlas en búsqueda de la solución general. Siendo este el camino a seguir, supongamos que tenemos un algoritmo capaz de secuenciar una cartera mensual de forma óptima, considerando las premisas de cumplimiento, procesamiento, reglas, setups, stocks intermedios y paradas que ya se explicaron con anterioridad. Llamaremos “J” a este algoritmo que secuenciar en forma óptima una cartera mensual. Este algoritmo por sí solo no es suficiente para alcanzar la solución general por dos razones a saber:

1. Si la cartera de un centro/mes no tiene overbooking podría ser que al secuenciarla si tenga. Esto puede suceder si el centro tiene reglas que obligan a mantener una campaña mínima de producto y en la cartera no existan órdenes de ese producto suficientes como para completarla. Así generamos paradas por falta de material en la secuenciación que no son deseables.
2. Si la cartera de un centro/mes tiene overbooking entonces no existe una secuenciación tal que solucione el overbooking.

En ambos casos el problema es el mismo, obtenemos una secuenciación óptima pero tenemos un overbooking. La única forma de solucionar un overbooking es quitando ordenes de la cartera hasta eliminarlo. La dificultad que ahora se presenta se resume en saber qué órdenes quitamos y dónde las colocamos. La segunda cuestión es más fácil de resolver porque tiene solo dos posibles soluciones. Las ordenes que quitamos las podemos pasar a centros alternativos de la tarea que no tienen overbooking en el mismo mes (ejemplo: desde el centro 6 en el mes de Marzo al centro 1 o 12, según tabla 1) o las pasamos a alguno de los meses siguientes del mismo centro (ejemplo: desde el mes de Mayo a Junio en el centro 8, según tabla 1). La primera cuestión, en cambio, no tiene una solución definida y en estos casos, como siempre, se resuelven mediante un cierto criterio.

Es importante entender que si movemos ordenes de un centro a otro alternativo en el mismo mes, es una decisión que (en la mayoría de las veces) solo atañe a la empresa pero si movemos ordenes de un mes al siguiente podríamos llegar a no cumplir con la fecha de entrega de la orden y esto involucra, en ciertos casos, acordarlo con el cliente. Considerado esto, podríamos pensar en un algoritmo capaz de secuenciar con el poder de mover órdenes entre centros alternativos en búsqueda de una mejor secuenciación de la cartera de la tarea y no de un centro en particular solamente. Este algoritmo lo vamos a considerar como una evolución del algoritmo “J” que habíamos considerado y lo vamos a llamar “Jo”. Sin embargo, aún no eliminamos el problema de overbooking en su totalidad ya que podríamos tener a todos los centros alternativos con overbooking. Debemos buscar un algoritmo capaz de resolver también los movimientos de un mes a otro.

Considerar movimientos de órdenes entre meses es influir en las carteras posteriores y es en este punto donde podemos diferenciar el objetivo entre programación y planificación de la producción. En su horizonte un programador solo quiere la mejor secuenciación de la cartera y considera que el overbooking serán órdenes atrasadas que deberán hacerse el próximo mes. Esta forma de considerar el problema es la realidad ya que no hay forma de producir más ordenes que la capacidad del centro lo permite, pero por otro lado es una solución a medias porque el próximo mes por considerar los atrasos de este vamos a volver a tener overbooking que será atraso para el siguiente y así sucesivamente. Es la gente de planificación que ve el problema de otra forma en su horizonte de trabajo y consideran que el overbooking debe resolverse con criterio. Es decir, tomar en cuenta información externa para considerar que orden puede atrasarse e incluso cuanto hacerlo. Por ejemplo, ciertas órdenes pueden atrasarse en un centro e igualmente llegar a cumplir con



## *Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

la orden completa porque los tiempos considerados en el cálculo de las fechas son promedios. Otras órdenes pueden atrasarse porque el costo de la penalidad no es significativo respecto a la rentabilidad. Y otras órdenes no pueden atrasarse de ninguna manera porque el cliente es demasiado importante. Recordemos que en el horizonte de planificación se trabaja mayormente con previsiones que son negocios no definitivos con el cliente y podría renegociarse la fecha de entrega. Lo cierto es que estos criterios no pueden sistematizarse y debemos tomar un criterio para resolver el overbooking.

En este trabajo vamos a considerar que una vez secuenciado de la mejor manera posible la cartera de una tarea vamos a quitar las órdenes de atrás para adelante en la secuenciación hasta eliminar el overbooking en cada centro. La justificación de este criterio es que se alinea con programación y es una muy buena sugerencia para planificación, aunque esto nos lleva a considerar la posibilidad de que el usuario lo haga mano con su propio criterio. Aplicando “Jo” desde la primera cartera de una tarea y considerando el overbooking como la secuencia inicial de la siguiente cartera, podemos secuenciar la tarea en todo el horizonte de tiempo. A este nuevo algoritmo lo vamos a llamar “Jos” y considerando como funciona vamos a poder obtener como resultado la tabla 2 que es una modificación de la tabla 1.

La diferencia fundamental entre la tabla 1 y la tabla 2 es que ahora indicamos en la cartera de cada mes los turnos que recibe de la cartera anterior (la columna de números en azul). De esta forma llegamos a una secuenciación de cada cartera y además el planificador tiene el poder de verificar que ordenes del overbooking (las ultimas de la secuenciación) se están atrasando de una cartera a la siguiente y tomar las decisiones que desee. Hasta este momento habíamos considerado un algoritmo que actúa sin interacción con usuarios pero ahora podemos modificar este algoritmo para que sea capaz de trabajar con modificaciones de un usuario. Este pequeño detalle nos lleva a un sin fin de criterios, definiciones y consideraciones de como debe ser el algoritmo que sin duda serán una notable mejora pero que se considera absolutamente fuera del alcance de este trabajo.

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

Tarea	Centro	Mes												
		Febrero (75.6)		Marzo (83.7)		Abril (70.2)		Mayo (78.3)		Junio (78.3)		Julio (81)		
A	1	2.0	74.9	1.3	78.9	0.0	69.1	0.0	79.0	0.7	82.7	5.0	77.0	1.1
	6	0.0	76.5	0.9	83.9	1.1	69.8	0.7	76.8	0.0	77.0	0.0	83.2	2.2
	12	1.0	80.6	6.0	81.6	3.8	73.1	6.7	75.9	4.3	79.3	5.3	78.3	2.6
B	2	2.5	75.6	2.5	80.5	0.0	70.8	0.6	79.8	2.1	82.0	5.8	80.0	4.8
	18	4.0	73.1	1.5	85.7	3.5	69.1	2.4	76.7	0.8	75.2	0.0	77.3	0.0
C	3	1.8	81.5	7.7	84.6	8.6	67.2	5.6	76.4	3.7	80.0	5.4	83.9	8.3
	8	0.0	76.4	0.8	88.0	5.1	68.3	3.2	79.4	4.3	75.0	1.0	80.0	0.0
	11	3.5	71.4	0.0	82.2	0.0	73.0	2.8	73.6	0.0	74.6	0.0	78.8	0.0
	13	2.9	74.4	1.7	82.9	0.9	74.2	4.8	82.3	8.9	77.8	8.4	77.1	4.4
	16	1.8	77.0	3.2	87.3	6.8	69.8	6.4	78.1	6.2	81.8	9.7	80.1	8.8
D	4	2.4	77.3	4.1	84.0	4.4	65.9	0.1	81.6	3.5	81.5	6.6	76.4	2.0
	7	3.5	72.5	0.4	85.9	2.6	72.6	5.0	81.1	7.8	77.0	6.5	84.0	9.5
	9	0.8	79.0	4.2	87.0	7.5	72.1	9.5	75.1	6.2	79.9	7.8	82.4	9.2
E	5	1.5	77.5	3.4	83.0	2.7	69.7	2.2	76.5	0.4	73.7	0.0	81.7	0.7
	10	3.5	69.5	0.0	87.0	3.3	72.4	5.5	74.1	1.3	79.4	2.4	80.5	1.9
	15	2.0	74.8	1.2	88.4	5.9	70.0	5.7	76.7	4.0	78.3	4.0	80.9	3.9
	17	0.0	79.4	3.8	83.0	3.1	68.0	0.9	80.9	3.5	78.6	3.8	85.5	8.3
F	14	2.7	73.9	1.0	88.0	5.3	67.0	2.1	80.7	4.5	81.9	8.1	80.1	7.2
	19	3.4	70.6	0.0	80.2	0.0	75.0	4.8	75.8	2.3	77.1	1.1	80.6	0.7
	20	4.5	73.6	2.5	85.6	4.3	69.9	4.0	76.0	1.7	81.7	5.1	77.2	1.3

Tabla 2

Sabemos que las tareas tienen centros alternativos que pueden realizar la misma tarea y que las tareas son dependientes entre si pero no son alternativas. Supongamos que las tareas de la tabla 2 están ordenadas en el sentido por el que tienen que pasar las órdenes para producirse y si bien, como hemos visto, alguna orden puede saltar una tarea, esta siempre respetará el orden de la tabla. No podemos, por definición de Hdr, intercambiar una orden de la tarea E a la tarea C porque existe un orden entre ellas. Entonces, la única forma de secuenciar todas las tareas es hacerlo de una por vez y en forma ordenada. Así llegamos al algoritmo final de este trabajo que llamaremos "Joss". Este algoritmo debe secuenciar una tarea por vez hasta secuenciar todas las tareas de la empresa, pero tiene que resolver la dificultad de saber qué orden debe seguir. Nuevamente vamos a resolver este dilema mediante la adopción de un criterio porque a decir verdad cada empresa puede determinar el orden que más le interese dependiendo de sus necesidades y limitación. Mientras que una empresa puede decidir comenzar por la tarea más costosa/rentable, otra empresa puede decidir comenzar por la tarea más lenta. También se evaluó la posibilidad de ir secuenciando la cartera de un mes de todas las tareas mediante un cierto orden antes de pasar al siguiente mes. Se descartó porque en realidad cada cartera influye considerablemente en la posterior y además, todo cambio en la cartera afecta a las demás tareas. Supongamos que en una tarea determinada cambiamos una orden de un mes al siguiente, esta cambiará todo su cálculo de fechas en la Hdr afectando las carteras de las demás tareas donde se encuentre la orden. Entonces, pareciera más prolijo secuenciar toda una tarea antes de pasar a otra.

Una secuenciación óptima de una cartera no se puede hacer sin violar alguna de las fechas calculadas. Supongamos que dos órdenes distintas tienen en una cartera la misma fecha de entrega o muy similar. Al ubicarlas en la secuencia es probable que se encuentren juntas y si, además, se ubican cerca de la fecha de entrega es muy posible que la segunda o ambas no cumplan con la fecha. Estos casos se hacen más frecuentes si consideramos los movimientos manuales de un usuario como hemos analizado. Si logramos la

## *Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

mejor secuenciación de una cartera y violamos alguna fecha, porque no utilizar esta información para futuras secuenciaciones. Esto nos lleva a la idea de que una vez secuenciada una tarea y antes se secuenciar la siguiente en el orden que corresponda, podríamos utilizar la información de las fechas de cada orden en la primera secuenciación como base para un recálculo de las fechas en la HdR en tareas anteriores como en posteriores. Esto ayudaría a una mejor secuenciación de todas las tareas ya que las carteras se verían afectadas por otras que fueron secuenciadas con anterioridad. Supongamos que secuenciamos primero la tarea de mayor rentabilidad para la empresa, entonces todas las demás tareas se verían afectadas por esta. Las tareas posteriores reflejarían mejor la situación de la planta donde optimizamos los productos más rentables y las anteriores serían afectadas visualizandopara intentar cumplir lo mejor posible con ésta. Es esta última definición lo que vamos a considerar realmente como el algoritmo “Joss”.

Podemos resumir el algoritmo “Joss” como la aplicación ordenada de un algoritmo que secuencia carteras mensuales de una tarea con máquinas alternativas modificando las carteras faltantes como resultado de las ya secuenciadas a medida que avanza en la secuenciación de todas las tareas. Lo que no mencionamos hasta ahora es que algoritmo será capaz de secuenciar de la mejor manera posible una cartera considerando las premisas ya mencionadas. Es “Tabu Search” la heurística elegida para tal fin. Finalmente, como nuestro algoritmo “J” esta basado en ésta heurística vamos a denominar como “Joss<sub>by ts</sub>” al algoritmo general de este trabajo que acabamos de describir en detalle. El nombre corresponde a las siglas de *Job Shop Scheduling Problem* y hace mención a *Tabu Search* como el motor de secuenciación óptima.

Podemos formalizar lo expuesto hasta este punto de la siguiente manera. Este trabajo busca un algoritmo capaz de secuenciar la cartera completa de una empresa teniendo en cuenta la siguiente información de la misma.

$\{O_{1i}, O_{2i}, \dots, O_{ni}\}$ : es el conjunto de tareas de la orden  $i$  que se tienen que ejecutar para considerarla terminada. Conforman un árbol de posibilidades.

$M(O_{ji})$ , es la función que determina el centro que puede procesar la tarea  $j$  de la orden  $i$ .

$T(i, t)$ , es la función que determina que tarea esta procesando el centro  $i$  en el momento  $t$ .

$S(i, j, m)$ , es la función que determina el tiempo de setup entre la orden  $i$  y la orden  $j$  en el centro  $m$ .

$TE(j, n, i)$ , es la función que determina el tiempo de espera obligatorio entre la tarea  $j$  y  $n$  de la orden  $i$ .

El algoritmo busca una secuenciación que podemos describir con la siguiente información.

$C_i$ , cantidad de días de cumplimiento de la orden  $i$ . Si  $C_i > 0$ , significa que la orden se termino  $C_i$  días antes de su fecha estipulada.

$S_i$ , cantidad de turnos de Setup del centro  $i$ .

$T_{ij}$ , tiempo de espera del stock  $i$  para procesarse en la tarea  $j$ .

La secuenciación buscada debe tener el máximo cumplimiento, el mínimo setup y el mínimo tiempo de espera del stock. Lo formalizamos en la siguiente expresión:

$\text{máx.}\{ \sum C_i \} \& \text{mín.}\{ \sum S_i \} \& \text{mín.}\{ \sum T_{ij} \}$

## VI - Tabu Search

Mencionamos anteriormente que existe una gran cantidad de trabajos que proponen distintos algoritmos para aproximar a la solución del JSSP. Entre todos estos trabajos existen muchos que proponen utilizar la metaheurística Tabu Search. Un trabajo importante al respecto es el de Barnes y Laguna [3] donde recorren algunas de las propuestas que contribuyeron al éxito de Tabu Search en el área de la planificación de la producción. El trabajo recorre los casos de *Single Machine Scheduling Problems*, *Multiple Machine Scheduling Problems* y describe la similitud del problema de la secuenciación de tareas con el *Vehicle Routing Problem*. Trabajo sumamente interesante que finaliza sugiriendo algunas direcciones a seguir en futuras investigaciones.

Otro trabajo para destacar sobre esta técnica es el de Widmer y Hertz [14] donde presentan una nueva heurística para resolver la secuenciación óptima de  $n$ -tareas en  $m$ -maquinas. Este método combina una búsqueda inicial de una secuenciación utilizando una analogía con problema del viajante de comercio y la optimización de esta solución inicial por medio de Tabu Search.

La metaheurística Tabu Search (TS), vio sus inicios hace tiempo y fue formalizada en 1986 por Fred Glover. Desde ese momento ha demostrado ser muy efectiva frente a otras técnicas en un vasto espectro de problemas de optimización y este éxito fue particularmente importante en el área de la secuenciación de producción. En esencia, TS es una técnica adaptable que dirige uno o más procesos de búsqueda local en persecución de un óptimo global partiendo de una solución inicial y utilizando funciones de memoria para evitar caer en óptimos locales. El marco de TS permite un alto grado de libertad para diseñar procedimientos de búsqueda y esta libertad es la que los investigadores han aprovechado desde sus inicios para explorar estrategias que resultaron en métodos de solución muy potentes. En la figura 5 se presenta un esquema de cómo funciona la metaheurística TS.

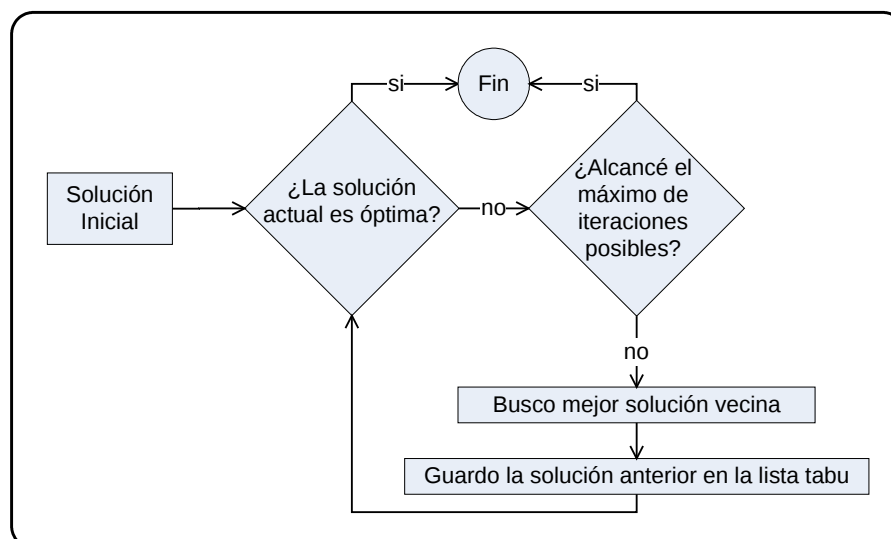


Figura 5

Lo primero que se debe definir para poder utilizar TS es una función que, considerando las restricciones definidas, sea capaz de evaluar una solución del problema. El poder evaluar una solución nos permite comparar soluciones entre si y determinar que solución es mejor. A continuación, se debe definir la estrategia de búsqueda que se va a utilizar para pasar de una solución a otra solución vecina. El poder pasar

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

de una solución a otra nos permite recorrer el campo de soluciones y considerando la función de evaluación vamos a determinar si la nueva solución es mejor que la anterior. El pasar de la solución “A” a la solución “B” significa que realizamos un cambio en la secuenciación de la solución “A” para poder obtener la solución “B”. Un cambio implica mover una orden de una posición a otra y es la posición original la que vamos a guardar en la lista tabú para evitar que próximos cambios vuelvan a dejar a esta Op en la posición original que tenía. Por último vamos a definir la longitud de la lista tabú que utilizaremos, esta tabla guardará la posición original de cada cambio realizado en los últimos  $n$  saltos entre soluciones vecinas. Estas posiciones originales no se podrán utilizar en los próximos cambios mientras estén en la lista tabú. Usamos una lista tabú de longitud fija  $n$  que se comporta como una lista FIFO (First In First Out), entonces un cambio no será posible repetirlo por los próximos  $n$  cambios una vez ingresado en la lista. A continuación hay un pseudo código que intenta clarificar como procede TS:

1.  $X \leftarrow$  solución inicial
2.  $L \leftarrow []$
3. mientras CANT  $\neq$  0
4.        $Y \leftarrow$  conj. de soluciones vecinas a X
5.        $X' \leftarrow$  mejor solución de Y
6.        $L \leftarrow \text{dif}( X, X' ) + L$
7.        $X \leftarrow X'$
8.       CANT  $\leftarrow$  CANT - 1
9. devolver X

donde X, X' son soluciones del problema, L es la lista tabú, CANT es una cantidad fija de iteraciones, Y es un conjunto de soluciones, dif(A, B) es el cambio que diferencia una solución de otra y “+” representa la operación de insertar en la primera posición de la lista tabú la posición original del cambio descartando la última porque L es de longitud fija.

En la figura 6 se visualiza la representación de la secuenciación de tareas en un centro dado al que aplicaremos TS para optimizar. Este grafo representa los órdenes en los nodos y los tiempos de setup entre las órdenes en los ejes, siendo los azules los que indican la secuenciación actual y en negro los posibles cambios que existen. Cada eje tiene un peso (no se visualiza en la figura) que es igual al tiempo de setup entre las ordenes que conecta y el sentido del eje indica el orden del cambio. Es necesario el sentido de los ejes porque no necesariamente el tiempo de setup al cambiar de la orden A a la B es el mismo que al cambiar de la orden B a la A, incluso podría ser que uno de los dos cambios no sea factible. Si pasar de una orden a otra no es factible debido a las reglas de producción del centro, esto se representa mediante la ausencia del eje correspondiente. Cada uno de los nodos se inserta en el grafo con la información que será utilizada en la estrategia de búsqueda, en este caso cada nodo contiene la fecha de liberación, la fecha tardía y el tiempo de duración. Es necesario ubicar la secuenciación en el tiempo para poder manejar las fechas como restricciones en la estrategia de búsqueda, por ello cada nodo tiene una fecha absoluta, fecha desde, donde debe ubicarse en el tiempo. La fecha desde del primer nodo de la secuenciación es configurada al momento de iniciar la optimización y las demás fechas se calculan a partir de esta con la ayuda de las duraciones y tiempos de setup entre cada nodo. La fecha desde no puede ser menor a la fecha de liberación (no puedo comenzar a producir si no tengo el material en el centro) en cada nodo y en los casos que lo fuese implica una parada por falta de material igual a la diferencia de tiempo entre las fechas.

# Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

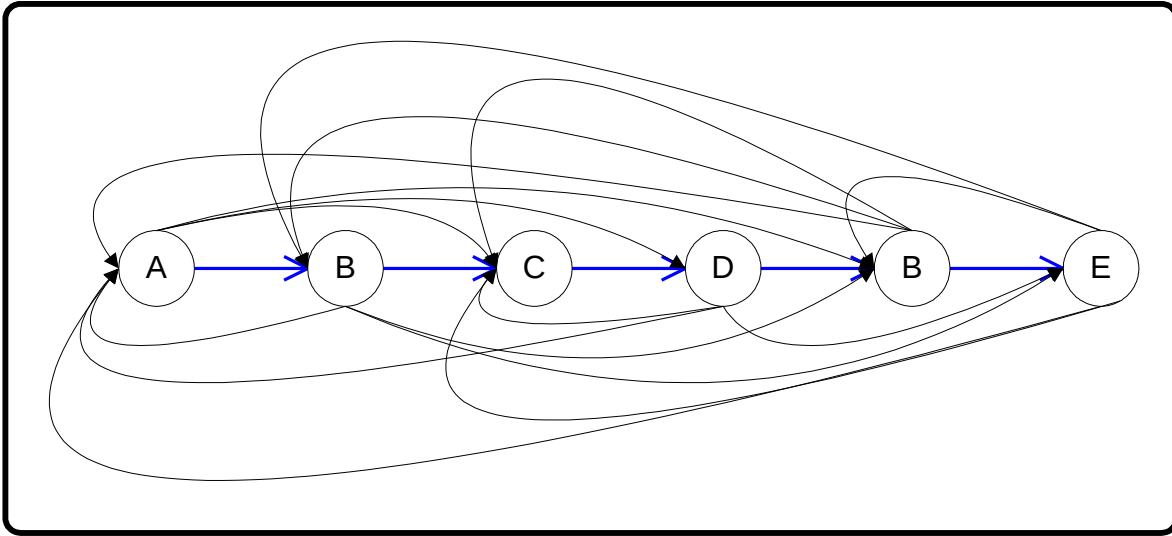


Figura 6

En nuestro caso la estrategia de búsqueda se basa en analizar para una secuenciación dada la posibilidad de mover una orden de lugar. Es decir que el conjunto de soluciones vecinas a una solución X son todas las secuenciaciones que se pueden obtener moviendo solamente una orden de la secuenciación X y la posición original de la orden antes del movimiento será lo que guardaremos en nuestra lista tabú como futuros movimientos inválidos. A continuación vamos a definir que significa mover una orden de lugar y cual será la función de evaluación de una secuenciación para poder completar el algoritmo de TS que aplicamos.

Mover una orden de lugar lo vamos a definir como un conjunto de restricciones necesarias que podemos visualizar en la figura 7. Supongamos que queremos mover el nodo D entonces son necesarios los ejes AD, DB y CE para poder realizarlo aunque para los casos en que D o A son los últimos nodos de la secuenciación no son necesarios los ejes CE o DB respectivamente. En caso de realizar este movimiento vamos a guardar en la lista tabú la tripla (C, D, E) que indica para futuros movimientos no realizar uno que deje a D entre C y E.

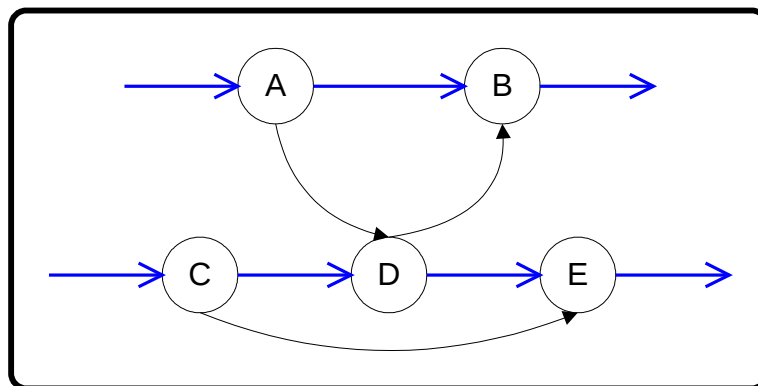


Figura 7

La función de evaluación nos va a permitir evaluar cual de todas las soluciones vecinas es la mejor que no necesariamente es mejor que la actual. Siguiendo el ejemplo de la figura 6, esta función la definimos como:

$$\Delta S + (T_{\max} * \text{Coef})$$

siendo

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

$$\Delta S = (S_{AB} + S_{CD} + S_{DE}) - (S_{AD} + S_{DB} + S_{CE})$$

$$\text{Coef} = \begin{cases} 6 & \text{si } F_B \ \& \ F_D \ \& \ \Delta S \geq 0 \\ 3 & \text{si } F_B \ | \ F_D \ \& \ \Delta S \geq 0 \\ 0 & \text{si } !F_B \ \& \ !F_D \ \& \ \Delta S \geq 0 \\ 0 & \text{si } F_B \ \& \ F_D \ \& \ \Delta S < 0 \\ -3 & \text{si } F_B \ | \ F_D \ \& \ \Delta S < 0 \\ -6 & \text{si } !F_B \ \& \ !F_D \ \& \ \Delta S < 0 \end{cases}$$

donde  $S_{ij}$  es el setup entre el nodo  $i$  y el nodo  $j$ ,  $T_{\max}$  el tiempo máximo de setup del centro y  $F_i$  es una función que determina si el nodo  $i$  cumple la propiedad de las fechas.

La función de evaluación determina cual es el efecto en el tiempo de setup para un movimiento ya que hace la diferencia entre los setups que había (AB, CD y DE) antes del movimiento con los setups que habrá (AD, DB y CE) después del movimiento. Si la diferencia es positiva significa que estoy obteniendo una disminución en el tiempo de setup (deseable) y si la diferencia es negativa significa que estoy obteniendo un incremento en el tiempo de setup (no deseable). Este efecto en el setup se ve agrupado por el efecto en las fechas de los nodos B y D que son los más impactados por el movimiento. La propiedad de las fechas se cumple si la fecha desde de un nodo se encuentra dentro de su respectiva ventana entre la fecha de liberación y la fecha tardía. En la figura 8 se visualiza como quedan ordenadas y agrupadas las soluciones vecinas con esta función de evaluación y aquella que se ubique más a la izquierda será la mejor vecina.

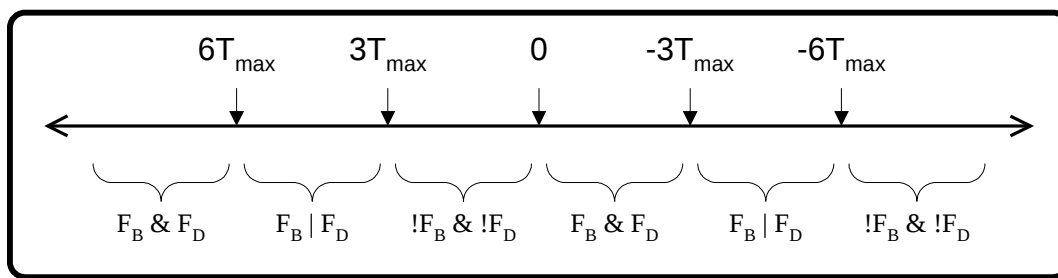


Figura 8

En un principio los coeficientes se pensaron como valores positivos solamente, pero estos no permitían el efecto que acabamos de describir. Necesitábamos de alguna forma lograr diferenciar las soluciones “deseables” de las “no deseables” y es por ello que se introdujeron los valores negativos para calcular los coeficientes. De esta forma continuamos manejando todo el conjunto de soluciones, priorizando las deseables y dejando como segunda alternativa en caso de no tener deseables las soluciones no deseables. Por otro lado, necesitábamos que las soluciones no se superpusieran para evitar conflictos. Si consideramos que  $\Delta S$  tiene un valor máximo de  $3 * T_{\max}$  cuando  $S_{AB} = S_{CD} = S_{DE} = T_{\max}$  y  $S_{AD} = S_{DB} = S_{CE} = 0$ , entonces los valores debían ser 0, 3 y 6 como coeficientes positivos y negativos.

La lista tabú se consideró de longitud  $n^{1/2}$ , donde  $n$  es la cantidad de nodos del grado, debido a que la bibliografía consultada sugiere este valor como el más apropiado. En la implementación de este trabajo (ver experimentos) se consiguieron los mejores resultados con este valor luego de utilizar varios distintos.

Lo explicado hasta este punto es un algoritmo que funciona para intentar optimizar mediante TS la secuenciación de un centro considerando las restricciones de setup, reglas y fechas. Este algoritmo fue

## *Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search*

implementado y dio resultados (ver experimentos) alentadores para seguir adelante. El siguiente paso será poder considerar la optimización de más de un centro al mismo tiempo y una vez alcanzado esto podemos utilizarlo como se explicó en “Modelo Propuesto”.



## VII - Experimentos

En esta sección vamos a presentar los resultados obtenidos al experimentar con la implementación del algoritmo “J” que describimos detalladamente con anterioridad. El algoritmo se implementó en Visual C++ 6.0 como una clase y esta se incorporó al sistema de programación y planificación de producción que tiene actualmente la empresa siderúrgica particular que tomamos como ejemplo de este trabajo. Este sistema es básicamente un algoritmo goloso que genera un gantt de la producción y una serie de módulos para la planificación, lo que facilitó el uso de la clase ya que este sistema tiene toda la información que necesitamos.

Las pruebas consistieron en secuenciar con el algoritmo “J” la cartera mensual de un centro y comparar el resultado obtenido con la secuencia original que genera el sistema de la empresa. Además de tomar varios escenarios distintos, también se aprovechó para realizar pruebas modificando los parámetros del algoritmo como la longitud de la lista tabú, la cantidad de iteraciones y la fecha de ubicación en el tiempo. A continuación se presentan los más representativos de ciertos casos relevantes para informar. Se tomó finalmente como valor óptimo de longitud de la lista tabú a  $n^{1/2}$  donde  $n$  es la cantidad de tareas involucradas como sugiere el trabajo de Schmidt [10].

A continuación se presenta una breve descripción de cada uno de los centros utilizados como ejemplo en este trabajo a modo informativo para el lector.

- BO39** Corta el extremo del tubo, lo inspecciona interna y externamente con equipos de control no destructivo y además le hace el bisel.
- DANO** Es un torno a control numérico y sirve para hacer roscas especiales (Premium principalmente).
- MORI** Idem DANO.
- PHTW** Es la prueba hidráulica para tubos Premium principalmente.
- ROTU** Es la roscadora a control numérico para roscas comunes (API).
- SWAG** Expande y distensiona extremos para hacer roscas especiales Premium.
- TELI** Es la Balanza de la línea ROTU, es decir pesa el tubo, lo mide y le hace la marcación.
- TTTU** Es el tratamiento térmico para darle el grado final, inspecciona los extremos y los endereza.
- USL1** Es el ultrasonido del tubo para ver defectos internos y externos de laminación.

El algoritmo “J” se implementó con una interfaz gráfica que permite visualizar en forma clara, rápida y prolija los resultados obtenidos. En la figura 8 se muestran las partes que conforman la interfaz gráfica. Lo primero que debemos destacar es la sección de “Configuración datos” donde podemos configurar la fecha donde se ubicará la secuenciación seleccionada, lo que nos permite la verificación de las fechas, y podemos determinar si el primer nodo es posible o no reubicarlo en la secuenciación, lo que permite la conexión entre una secuenciación y otro. En la sección de “Configuración Tabu Search” podemos configurar la cantidad de iteraciones que el algoritmo va a utilizar y la longitud de la lista tabú. Contamos con el botón “Optimizar” que utilizamos una vez configuradas las dos secciones antes mencionadas y una barra de progreso que nos

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

indica el avance de la optimización. El dialogo presenta 5 solapas para poder aplicar la optimización hasta con 5 configuraciones distintas sobre el mismo juego de datos.

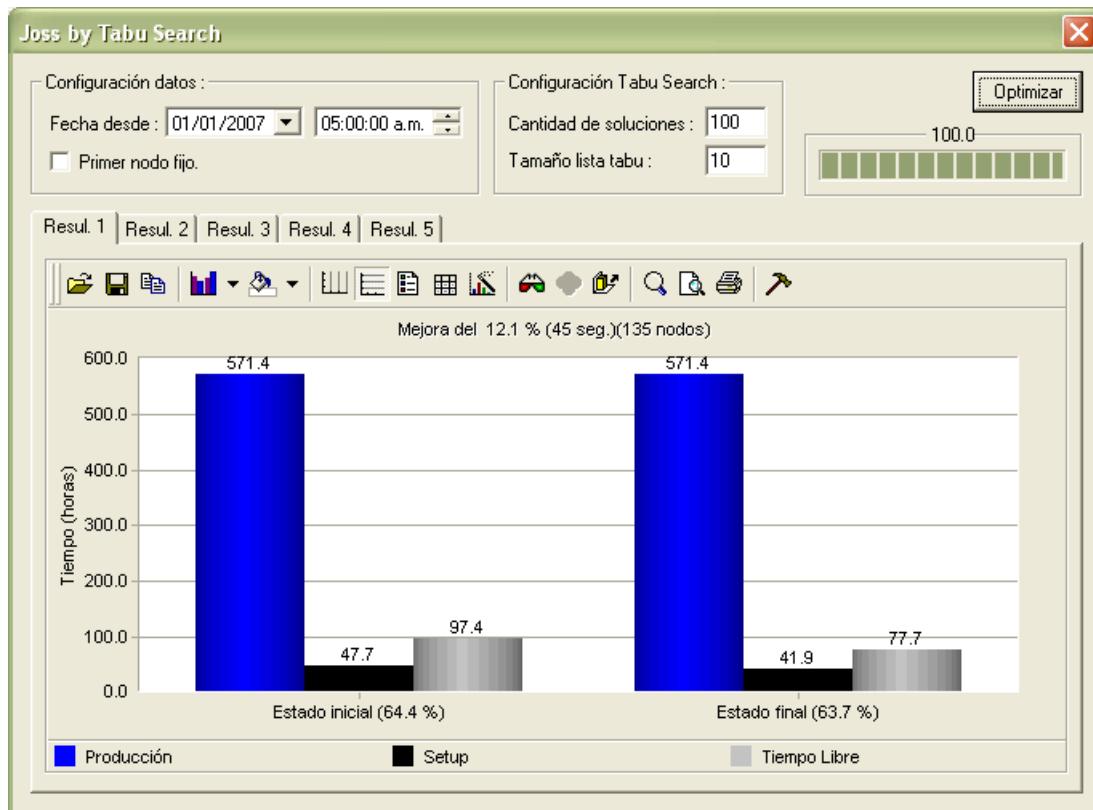


Figura 8

Una vez finalizada la optimización podemos visualizar en el gráfico central los resultados obtenidos considerando la siguiente información. El gráfico es una comparación entre los valores antes de optimizar (“Estado inicial”) y los valores después de optimizar (“Estado final”). El estado inicial corresponde a la secuenciación resultante del algoritmo de la empresa y se considera como la solución inicial que vamos a optimizar con nuestro modelo. Es el estado final corresponde a los resultados obtenidos luego de aplicar la parte de nuestro modelo programada al estado inicial. Los valores que se comparan son los más significativos para evaluar los resultados. Estos son el tiempo total de producción, setup y tiempo libre medidos en horas. El primero de ellos debe ser igual en ambos casos, ya que por definición no variamos la cartera, y los otros dos se espera reducirlos. En el título vamos a visualizar el porcentaje de mejora del tiempo de setup, junto al tiempo que se tardó en optimizar y la cantidad de nodos involucrados. Sobre cada columna de datos se indica el valor de horas que representa la misma y al lado de los títulos de cada estado se indica el porcentaje de nodos que cumplen con la propiedad de las fechas.

A continuación presentamos una serie de casos que resumimos en tablas para una mejor visualización de toda la información en conjunto.

### **Caso 1:**

Un caso muy representativo de la problemática de esta empresa. El centro BO39 tiene siete tipos de setup distintos pero de valores muy similares como se puede ver en la tabla 3. Al ser tiempos tan similares la secuenciación óptima pasa a ser una meseta máxima, es decir que más de una solución vecina son secuenciaciones óptimas y una vez alcanzada una de ellas es difícil que el algoritmo se salga de esta meseta.

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

Cambios de Setup de BO39		
N°	Descripción	Tiempo(min)
1	Cambio de diámetro	25
2	Cambio de producto	20
3	Cambio de espesor	20
4	Cambio de diámetro y CND	25
5	Cambio sin ultrasonido	20
6	Cambio de ultrasonido longitudinal	23
7	Cambio de ultrasonido transversal	20
8	Sin cambio	0

Tabla 3

Este efecto lo podemos ver en la siguiente secuencia de optimizaciones que se realizaron sobre la cartera de Enero de la BO39. Obsérvese que los valores alcanzados en el primer escenario se repiten sistemáticamente en los demás por más que variamos la longitud de la lista tabú. Por otro lado también observamos en el escenario 2 que el anclaje del primer nodo orienta al algoritmo hacia otra solución que incluso aumenta el tiempo libre de forma desproporcionada. En general es muy significativa la mejora del tiempo de setup (promedio de 12.1%) y no es significativo el empeoramiento de la cantidad de nodos que cumplen con al propiedad de las fechas (promedio de 0.7%  $\approx$  1 nodo).

	BO39														
	Escenario 1			Escenario 2			Escenario 3			Escenario 4			Escenario 5		
	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.
Producción	571.4	571.4	0.0%	571.4	571.4	0.0%	571.4	571.4	0.0%	571.4	571.4	0.0%	571.4	571.4	0.0%
Setup	47.7	41.9	12.2%	47.7	42.0	11.9%	47.7	41.9	12.2%	47.7	41.9	12.2%	47.7	41.9	12.2%
Tiempo libre	97.4	77.7	20.2%	97.4	135.4	-39.0%	97.4	77.7	20.2%	97.4	77.7	20.2%	97.4	77.7	20.2%
Nodos totales	135.0			135.0			135.0			135.0			135.0		
Nodos cumplen PF	64.4%	63.7%	0.7%	64.4%	63.7%	0.7%	64.4%	63.7%	0.7%	64.4%	63.7%	0.7%	64.4%	63.7%	0.7%
Tiempo computo (seg)	45			43			44			46			47		
Cant. soluciones	100			100			100			100			100		
Long. lista tabu	10			10			5			50			100		
Fecha desde	01/01/2007 05:00			01/01/2007 05:00			01/01/2007 05:00			01/01/2007 05:00			01/01/2007 05:00		
Primer nodo fijo	no			si			no			no			no		

Tabla 4

### Caso 2:

Otro caso muy representativo de la problemática de esta empresa es el de la cartera de Febrero del centro TTTU. Este centro tiene 14 tipos de setup distintos, combinables y de valores muy dispersos entre si como se puede ver en la tabla 5. La diferencia importante es que los tiempos de setup son muy dispersos y por ende no se forman mesetas de soluciones óptimas como sucedía en el centro antes expuesto. Incluso podemos observar que hasta el cambio “Sin Cambio” implica un tiempo mínimo de 10’, lo que representa el tiempo necesario para efectuar el cambio de orden en los sistemas de control.

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

Cambios de Setup de TTTU		
N°	Descripción	Tiempo(min)
1	Cambio de producto no definido	58
2	Cambio de CND	30
3	Ajuste de CND	20
4	Apertura de módulos	20
5	Cierre de módulos	45
6	Ajuste pequeños módulos	10
7	Ajuste grandes módulos	17
8	Ajuste pequeños enderezadora	10
9	Ajuste grandes enderezadora	17
10	Cambio de tina	20
11	Montado de calibración	60
12	Desmontado de calibración	40
13	Cambio de calibración	80
14	Sin cambio	10

Tabla 5

En la siguiente secuencia de optimizaciones podemos visualizar que si tiene efecto en el resultado utilizar distintas longitudes de lista tabú. En el escenario 1 visualizamos una mejora del 12.7% al utilizar la longitud que consideramos óptima. En cambio, en el escenario 2, observamos que al utilizar una longitud menor obtenemos solo una mejora del 5.6%. En los dos últimos escenarios configuramos longitudes mayores de lista tabú y obtenemos los mismos resultados que en el primero. El escenario 3 muestra un caso especial donde se encontró una solución considerablemente superior pero no es un resultado para utilizar porque empeora demasiado el tiempo libre (> 700%)

	TTTU														
	Escenario 1			Escenario 2			Escenario 3			Escenario 4			Escenario 5		
	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.
Producción	574.7	574.7	0.0%	574.7	574.7	0.0%	574.7	574.7	0.0%	574.7	574.7	0.0%	574.7	574.7	0.0%
Setup	60.5	52.8	12.7%	60.5	57.1	5.6%	60.5	46.6	23.0%	60.5	52.8	12.7%	60.5	52.8	12.7%
Tiempo libre	17.0	20.3	-19.4%	17.0	16.0	5.9%	17.0	150.3	-784.1%	17.0	20.3	-19.4%	17.0	20.3	-19.4%
Nodos totales	101			101			101			101			101		
Nodos cumplen PF	80.9%	81.9%	-1.0%	80.9%	84.9%	-4.0%	80.9%	76.7%	4.2%	80.9%	81.9%	-1.0%	80.9%	81.9%	-1.0%
Tiempo computo (seg)	17			17			17			17			17		
Cart. soluciones	100			100			100			100			100		
Long. lista tabu	10			5			20			50			75		
Fecha desde	01/02/2007 05:00			01/02/2007 05:00			01/02/2007 05:00			01/02/2007 05:00			01/02/2007 05:00		
Primer nodo fijo	no			no			no			no			no		

Tabla 6

### Caso 3

Considerados en detalle los casos anteriores que son los más representativos de la problemática de la empresa vamos a presentar más resultados para otros centros a modo de seguir ejemplificando el uso del algoritmo. En la tabla 7 se resumen los resultados obtenidos con la cartera de Abril para el centro DANO y MORI y la cartera de Febrero para el centro SWAG y USL1. En todos los escenarios tenemos una mejora en el tiempo de setup, pero se repiten los casos en que se dispara el tiempo libre como en el DANO y el USL1. Se observa, como en resultados anteriores, una mejora en la cantidad de nodos que cumplen con la propiedad de las fechas en la mayoría de los escenarios. En algunos casos también se observa un empeoramiento pero tanto en la mejora como en el empeoramiento se da en un valor muy pequeño. Este fenómeno se puede traducir en que la secuenciación original no es óptima y que con algunos cambios se llega a una mejor

## Un Caso Real de Secuenciación de Tareas Dependientes con Tabu Search

solución sin violar más fechas que la original. También es interesante observar como en el escenario del USL1 se da una mejora de prácticamente el 50% pero a costa de una 300% de empeoramiento del tiempo libre.

	DAHO			MORI			SWAG			USL1		
	Escenario 1			Escenario 1			Escenario 1			Escenario 1		
	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.
Producción	520.6	520.6	0.0%	559.5	559.5	0.0%	497.0	497.0	0.0%	487.4	487.4	0.0%
Setup	79.8	66.2	17.0%	15.5	14.5	6.5%	43.6	35.6	18.3%	40.3	21.0	47.9%
Tiempo libre	119.6	267.4	-123.6%	55.7	29.3	47.4%	92.8	100.8	-8.6%	70.4	340.9	-384.2%
Nodos totales	26			22			45			52		
Nodos cumplen PF	69.2%	65.4%	3.8%	69.1%	73.6%	-4.5%	75.6%	75.6%	0.0%	71.5%	67.7%	3.8%
Tiempo computo (seg)	0			1			3			3		
Cant. soluciones	100			100			100			100		
Long. lista tabu	10			10			10			10		
Fecha desde	01/04/2007 05:00			01/04/2007 05:00			01/02/2007 05:00			01/02/2007 05:00		
Primer nodo fijo	no			no			no			no		

Tabla 7

### Caso 4:

A continuación en la tabla 8 se resumen los escenarios especiales de dos carteras juntas para distintos centros. En particular trabajamos con las carteras de Febrero y Marzo juntas para los centros PHTW, ROTU y TELI. En este caso podemos observar que el escenario de la PHTW no presenta ningún cambio respecto a la solución original. Los restantes escenarios no presentan diferencias respecto a lo que hemos observado hasta el momento.

	PHTW			ROTU			TELI		
	Escenario 1			Escenario 1			Escenario 1		
	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.	Estado Inicial	Estado Final	Dif.
Producción	1201.1	1201.1	0.0%	793.8	793.8	0.0%	955.2	955.2	0.0%
Setup	37.3	37.3	0.0%	138.6	96.7	30.2%	45.2	34.8	23.0%
Tiempo libre	101.3	101.3	0.0%	487.2	529.2	-8.6%	418.1	428.4	-2.5%
Nodos totales	253			173			255		
Nodos cumplen PF	18.6%	18.6%	0.0%	47.4%	46.8%	0.6%	53.7%	53.7%	0.0%
Tiempo computo (seg)	7			52			15		
Cant. soluciones	100			100			100		
Long. lista tabu	10			10			10		
Fecha desde	01/02/2007 05:00			01/02/2007 05:00			01/02/2007 05:00		
Primer nodo fijo	no			no			no		

Tabla 8

Se realizaron muchas más pruebas pero solo se presentan las más representativas. Se puede observar que los resultados son muy variados pero en líneas generales siempre se presenta una mejora importante en el tiempo de setup (>10%) junto con alguna pequeña o nula mejora en el tiempo libre. Hay que tener en cuenta que estas pruebas cumplen con el objetivo de observar el buen funcionamiento de TS como herramienta para optimizar la secuenciación de producción. Es decir que es solamente una pequeña parte respecto a todo el modelo presentado en el trabajo.

## VIII - Conclusiones

Los resultados obtenidos nos indican, una vez más, lo exitoso de la técnica Tabu Search para resolver problemas de optimización. Podemos observar en la mayoría de los casos expuestos y de los no expuestos, que se obtienen resultados con una mejora significativa frente a la solución del sistema actual que tiene la empresa estudiada. Hemos observado que la longitud de la lista tabú juega un papel importante y que el valor  $n^{1/2}$  dio buenos resultados en la mayoría de los escenarios. Al modificar el valor de ésta variable se obtuvieron los mismos o peores resultados. Es por eso que sugerimos utilizar este valor como el más adecuado en futuros trabajos. Destacamos que estos experimentos son solamente un pequeño paso en el estudio de toda la solución integral desarrollada en este trabajo. Los resultados obtenidos alientan a seguir con el modelo propuesto en este trabajo.

En el siguiente paso se debe resolver la secuenciación de la cartera de órdenes de una tarea en lugar de solamente un centro. En esta cuestión se debe sumar la posibilidad de que mover una orden de un centro a otro, es decir de la cartera de un centro a otra. En la figura 9 se visualiza la secuenciación de una tarea que deberá enfrentar nuestro nuevo algoritmo teniendo en cuenta la cartera de tres centros de una misma tarea y representamos con ejes en línea punteada las nuevas alternativas.

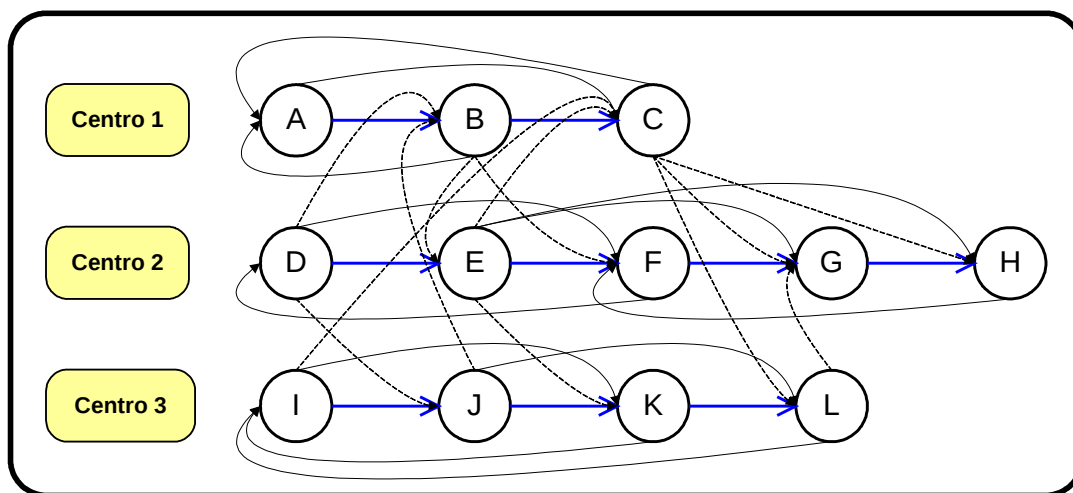


Figura 9

Este siguiente paso está fuera del alcance de este trabajo e implica no solo la capacidad de resolverlo sino también de hacerlo en un tiempo de respuesta acorde. Recordemos que la realidad de la planta está en constante movimiento (rotura de máquinas, atrasos inesperados, cambios de condiciones, etc.) y este algoritmo debe responder a esos cambios en forma rápida. Pensamos que el algoritmo deberá dar una optimización inicial y luego sobre esta los usuarios podrán ir contemplando los cambios inesperados que la planta informe. El tiempo de cómputo para la optimización inicial no debe ser una cuestión importante porque en general el usuario estará operando el algoritmo con los cambios inesperados, los ingresos y egresos. Si el tiempo de procesamiento de esta optimización inicial fuera extremadamente grande, se la puede pensar como un proceso “batch” anterior a la operatoria del usuario. En cambio, el tiempo de respuesta del procesamiento de los cambios inesperados debe ser mínimo.

## **IX - Bibliografía**

- [1] G. Adelson-Velsky and E. Levner. “*Project Scheduling in and-or graphs: a Generalization of Dijkstra’s Algorithm*”. Mathematics of Operations Research, Vol. 27, N° 3, pp 504-517, 2002.
- [2] D. Applegate and W. Cook. “*A Computational Study of the Job-Shop Scheduling Problem*”, ORSA Journal on Computing, Vol. 3, N° 2, 1991.
- [3] J. W. Barnes and M. Laguna. “*A Tabu Search Experience in Production Scheduling*”, Annals of Operations Research, Vol. 41, pp 141-156, 1993.
- [4] J. G. Dai and G. Weiss. “*A Fluid Heuristic for Minimizing Makespan in Job Shops*”, Operational Research INFORMS, Vol. 50, N° 4, pp. 692-707, 2001.
- [5] M. R. Garey and D. S. Johnson. “*Computers and Intractability: A Guide to the Theory of NP-Completeness*”, Kluwer Academic Publishers, 1979.
- [6] A. Garrido, M. A. Salido, F. Barber and M. A. López. “*Heuristic Methods for Solving Job Shop Scheduling Problem*”, Universidad politécnica de Valencia, España, 2000.
- [7] J. Käschel, T. Teich, G. Köbernik and B. Meier. “*Algorithms for de Job Shop Scheduling Problem – a comparison of different methods*”, Proceedings of the European Symposium on Intelligent Techniques. Crete (Greece), S. 74, 1999.
- [8] S. Lin, E. D. Goodman and W. F. Punch. “*A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems*”, Proc. 7<sup>th</sup> Internat. Conf. on Genetic Alg., East Lansing, MI, Morgan Kaufmann, San Francisco, pp. 481-488, 1997.
- [9] N. Sadeh, K. Sycara and Y. Xiong. “*Backtracking Techniques for the Job Shop Scheduling Constraint Satisfaction Problem*”, tech. report CMU-RI-TR-94-31, Robotics Institute, Carnegie Mellon University, October, 1994.
- [10] K. Schmidt. “*Using Tabu Search to Solve the Job Shop Scheduling Problem with Sequence Dependent Setup Times*”, ScM Thesis, Brown University, 2001.
- [11] J. P. Watson, L. D. Whitley and A. Howe. “*A Dynamic Model of Tabu Search for the Job Shop Scheduling Problem*”, Multidisciplinary International Conference on Scheduling, 2003.
- [12] M. Widmer. “*Job Shop Scheduling with Tooling Constraints: a Tabu Search Approach*”, Journal of Operational Research Society, Vol. 42, N° 1, pp 75-82, 1991.
- [13] M. Widmer and A. Hertz. “*A New Heuristic Method for the Flow Shop Sequencing Problem*”, European Journal of Operational Research, Vol. 41, pp. 186-193, 1989.