

# Proyecto Fin de Master en Organización Industrial y Gestión de Empresas

Una metodología para la planificación y  
programación de la producción en la industria de  
componentes plásticos usando modelos de  
optimización

Autor: Rafael Márquez Bueno

Tutor: David Canca Ortiz

**Dpto. Organización Industrial y Gestión de  
Empresas I**

**Escuela Técnica Superior de Ingeniería**

Sevilla, 2020





Proyecto Fin de Master  
Organización Industrial y Gestión de Empresas

# **Una metodología para la planificación y programación de la producción en la industria de componentes plásticos usando modelos de optimización**

Autor:

Rafael Márquez Bueno

Tutor:

David Canca Ortiz

Catedrático de Universidad

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020



Proyecto Fin de Master: Una metodología para la planificación y programación de la producción en la industria de componentes plásticos usando modelos de optimización

Autor: Rafael Márquez Bueno

Tutor: David Canca Ortiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

*A mi familia*

*A mis maestros*



# Agradecimientos

---

A mi familia, y más concretamente a mi hermano, por confiar en mí y guiarme en todo momento desde el día que decidí afrontar el camino de la ingeniería. A lo largo de estos años no ha dejado de transmitirme su motivación y dedicación en este campo.

A mis profesores del Master, en especial a mi tutor del proyecto. Gracias a su atención, su siempre disposición y ganas de enseñar y, sobre todo, paciencia a lo largo de todo el recorrido de la elaboración de este trabajo que pone fin a mi etapa como estudiante de la US.

A la empresa, por darme la oportunidad de conocer su entorno, permitir que pudiera realizar el proyecto que nos concierne y por facilitar los datos y la información relevante para poder llevarlo a cabo.

Rafael Márquez Bueno

Sevilla, 2020



# Resumen

---

El presente trabajo fin de master se enmarca en el campo de la programación y secuenciación de la producción aplicado a un contexto industrial real. Se trabaja con una organización dedicada a la fabricación de envases de plástico para el sector alimentario. El producto en el que nos vamos a centrar es una pieza de plástico que se procesa en unas máquinas, en adelante líneas, en las que entra el plástico fundido y, a través de su inyección en un molde y de un ensamblaje final se obtiene el producto terminado.

Estas líneas están automatizadas, pero cuando hay que cambiar entre los diferentes colores del producto a fabricar se requiere de un proceso de preparación de las líneas o setup que supone importantes costes para la empresa.

Se conocen las referencias a producir a través de pedidos de aprovisionamiento del stock del almacén o pedidos que se reciben directamente de clientes. En ambos casos se dispone de un plazo de al menos 15 días para la entrega a clientes.

A todo esto, hay que sumarle las necesarias paradas para el mantenimiento preventivo de las líneas, que deben ser contempladas en la programación de la producción.

El sistema de producción de la compañía se compone de un conjunto de líneas de fabricación, dispuestas en paralelo, con distintas velocidades y con puntuales paradas para un mantenimiento preventivo. Estas líneas deben procesar unos pedidos sujetos a unas fechas de entrega. Los cambios entre las referencias de cada pedido en una línea provocan la aparición de unos tiempos de setup que se desean minimizar.

Para resolver este problema se propone un método de resolución exacta mediante la aplicación sucesiva de dos modelos de optimización. En una primera fase se pretende simplificar el tamaño del problema mediante la preasignación de los pedidos a subconjuntos de líneas de fabricación en función del color de los pedidos. En la segunda fase se resuelve el modelo de programación de la producción teniendo en cuenta las restricciones y condicionantes mencionados.



# Abstract

---

This final master's work is part of the field of production scheduling and sequencing applied to a real industrial context. We work with an organization dedicated to the manufacture of plastic containers for the food sector. The product we are going to focus on is a piece of plastic that is processed in machines, hereinafter lines, in which the molten plastic enters and, through its injection into a mold and a final assembly, the finished product.

These lines are automated, but when it is necessary to change between the different colors of the product to be manufactured, a line preparation or setup process is required, which entails significant costs for the company.

The references to be produced are known through supply orders for warehouse stock or orders received directly from customers. In both cases there is a period of at least 15 days for delivery to customers.

To all this, we must add the necessary stops for the preventive maintenance of the lines, which must be considered in the production scheduling.

The company's production system is made up of a set of manufacturing lines, arranged in parallel, with different speeds and with occasional stops for preventive maintenance. These lines must process orders subject to delivery dates. Changes between the references of each order in a line cause the appearance of setup times that you want to minimize.

To solve this problem, an exact resolution method is proposed through the successive application of two optimization models. In a first phase, the aim is to simplify the size of the problem by pre-assigning orders to subsets of production lines based on the color of the orders. In the second phase, the production scheduling model is solved taking into account the aforementioned restrictions and conditions.



# Índice

---

<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xv</b>
<b>Índice de Tablas</b>	<b>xvii</b>
<b>Índice de Figuras</b>	<b>xix</b>
<b>1 Objetivos</b>	<b>1</b>
<b>2 Descripción del problema</b>	<b>3</b>
2.1. <i>Definición del producto y el entorno productivo</i>	3
2.2. <i>Demanda y planificación de la producción</i>	4
2.3. <i>Descripción técnica de las líneas de fabricación</i>	5
2.4. <i>Planteamiento del problema</i>	7
<b>3 Estado del arte</b>	<b>9</b>
<b>4 Modelos matemáticos</b>	<b>15</b>
4.1 <i>Modelo Preliminar. Equilibrado y pre-asignación de colores a líneas</i>	18
4.1.1 <i>Formulación matemática del modelo preliminar</i>	19
4.2 <i>Modelo Principal. Programación de la producción</i>	22
4.2.1 <i>Entornos</i>	22
4.2.2 <i>Restricciones</i>	23
4.2.3 <i>Objetivos</i>	25
4.2.4 <i>Formulación matemática del modelo principal</i>	28
<b>5 Resolución del problema</b>	<b>33</b>
5.1. <i>Estructuración y obtención de los datos iniciales</i>	33
5.1.1 <i>Datos de los pedidos entrantes</i>	33
5.1.2 <i>Datos de la configuración del entorno productivo</i>	35
5.2. <i>Desarrollo e implementación</i>	37
5.3. <i>Análisis de resultados</i>	39
<b>6 Conclusiones y posibles mejoras</b>	<b>51</b>
<b>Referencias</b>	<b>53</b>



# ÍNDICE DE TABLAS

---

<i>Tabla 5.1 Listado de colores y tonalidades</i>	34
<i>Tabla 5.2 Ejemplo de tiempos de setup</i>	34
<i>Tabla 5.3 Parámetros de configuración inicial</i>	42
<i>Tabla 5.4 Tasas de salida (velocidades) de las líneas</i>	42
<i>Tabla 5.5 Pedidos en curso de las líneas</i>	43
<i>Tabla 5.6 Paradas por mantenimiento preventivo planificado en cada línea</i>	43
<i>Tabla 5.7 Listado de pedidos a programar</i>	44
<i>Tabla 5.8 Resumen de solución obtenida para el modelo preliminar</i>	46
<i>Tabla 5.9 Resumen de solución obtenida para el modelo principal</i>	46



# ÍNDICE DE FIGURAS

---

<i>Figura 2.1 Esquema de una línea de fabricación</i>	6
<i>Figura 4.1 Clasificación de los procesos del sistema productivo (Hartmut Stadler, 2005)</i>	16
<i>Figura 5.1 Flujoograma del proceso de programación por el planificador</i>	41
<i>Figura 5.2 Asignaciones de colores en líneas</i>	46
<i>Figura 5.3 Visualización Gantt de resultados de la programación de la producción</i>	47
<i>Figura 5.4 Programación de la producción detallada en la hoja de cálculo</i>	48



# 1 OBJETIVOS

---

Este trabajo fin de master aborda un problema real, por lo tanto, el principal objetivo es proporcionar una herramienta de uso práctico para resolver el problema en cuestión. Se trata de la programación y secuenciación de la producción en un entorno de máquinas uniformes en paralelo con el objetivo de minimizar los tiempos de setup entre los distintos pedidos que se procesan. Lo que podría parecer un problema ampliamente estudiado y de baja complejidad como es el de máquinas paralelas con tiempos de setup, se complica al estudiar un sistema real y los factores influyen en la programación de la producción. Estos son, por ejemplo, el tamaño de las instancias a resolver o las paradas de mantenimiento para las líneas que deben integrarse en las programaciones.

Es por esto que un primer objetivo es contextualizar el problema realizando una revisión de la literatura recopilando estudios que afrontan aplicaciones prácticas para problemas similares al que se trata en este trabajo.

El siguiente objetivo es lograr desarrollar un método de resolución exacto y que sea ejecutable tantas veces como reprogramaciones sean necesarias por las variaciones que puedan surgir en el día a día. Esto sin renunciar a unos tiempos de computación asumibles que permita la obtención de soluciones en un corto periodo de tiempo.

Se debe abarcar un horizonte temporal de 15 días, contemplando el número de pedidos que puedan llegar a programarse para ese periodo, el estado de las líneas en el momento de la resolución y las paradas de mantenimiento planificadas.



## 2 DESCRIPCIÓN DEL PROBLEMA

---

**E**n el presente proyecto se afronta un caso real. Para conservar la privacidad de la empresa que nos atañe, se generalizará la definición del producto y el entorno productivo. La empresa en cuestión se dedica a la producción de envases de plástico en el ámbito del sector alimentario. El producto final es el resultado del ensamblaje de los distintos elementos que son fabricados en los centros que posee la empresa. El caso de estudio se va a enfocar en una de las unidades productivas, dedicada a la fabricación de uno de estos subcomponentes. En adelante, para simplificar, se considerará este como el producto final a tener en cuenta.

### 2.1. Definición del producto y el entorno productivo

La unidad productiva que origina el caso que se estudia en este TFM produce piezas de plástico que se fabrican mediante un proceso de moldeo y ensamblaje. Este proceso tiene lugar en líneas de fabricación y montaje automáticas con un sistema integral y estanco donde entra la materia prima y sale el producto ya finalizado. Estas líneas poseen además otros sistemas de gestión de la producción como controles de calidad para los defectos y buffer interno para garantizar una tasa de salida constante del producto. En un apartado posterior se describirá en más detalle el funcionamiento de las líneas.

Podemos distinguir dos familias de productos sustancialmente diferenciadas según el tipo de envase al que va dirigido, los llamaremos productos A y B. Además, se fabrican en dos tipos de líneas completamente diferentes, lo que no permite permutaciones ni balanceos en las cargas de trabajo.

Las piezas que se producen pueden tener distintas propiedades, fundamentalmente se diferencian en cuanto al color y el grabado. Esto da lugar a una variedad de hasta 700 configuraciones o referencias distintas. No obstante, el grueso de la facturación corresponde a unas 34 versiones del tipo A y otras 20 del tipo B.

La fábrica dispone de 12 líneas de fabricación paralelas e idénticas para el producto A y 5 para el producto B. Como se ha comentado anteriormente, estas líneas son automáticas y disponen de un sistema automático de control de calidad que desecha las unidades con defectos, por lo que, para el caso de estudio, se considerarán como una caja negra. Cada línea se caracteriza por su tasa de salida, variable en función de ciertos parámetros,

que se detallarán más adelante.

Cada línea procesa pedidos por lotes de 4375 unidades de A y 3000 de B, cuya unidad mínima productiva es un pallet de 24 cajas de estos productos. Estos pallets se empaican y se almacenan en stock, para luego ser despachados al cliente interno.

En resumen, podemos decir que estamos ante un modelo de fabricación en masa de un producto de bajo coste, pero cuyo volumen de fabricación es bastante alto y la rotación del material, por lo general, no supera el año. Además, hay que tener en cuenta que el tiempo de vida del producto, desde su fabricación hasta que se desecha por el usuario, es aproximadamente de un año y medio.

Para poder abastecer toda la demanda, las líneas se mantienen en funcionamiento durante las 24 horas del día, en 5 turnos de trabajo. Para supervisar las líneas, controlar y gestionar las averías, revisar los defectos y atender a posibles atascos de la forma más rápida posible, se dispone de personal cualificado capaz de atender simultáneamente hasta 2 líneas. No obstante, se puede recurrir a personal extra para asumir picos de trabajo, por lo que este punto no va a ser considerado como un factor limitante en el problema.

## 2.2. Demanda y planificación de la producción

La compañía maneja una cartera de clientes muy amplia, que realizan pedidos de gran volumen a lo largo de todo el año. Para abastecer la demanda y optimizar los costes de inventario, la compañía ha establecido dos estrategias:

- **Make to Stock**, para las referencias de mayor demanda. Plantea un modelo dinámico de inventarios con políticas de reaprovisionamiento, stocks de seguridad y previsión de la demanda. Trata de garantizar la satisfacción de la demanda habitual del cliente, actuando ante posibles picos de demanda o la aparición de pedidos urgentes. De este modo, se tiene destinado un almacén para este tipo de referencias, y mientras se produce para reponer a los niveles de stock establecidos, el departamento de shipping se encarga de realizar los despachos y satisfacer de forma autónoma la demanda de las referencias más solicitadas.
- **Make to Order**, para configuraciones de productos específicas o de poca significación. Modelo de fabricación bajo pedido. De esta forma, se elimina el inventario de referencias especiales o de baja rotación y se realiza la expedición directa.

Para poder establecer un balance entre ambas estrategias, la empresa previamente realizó un estudio ABC de segmentación y clasificación de productos, así como una previsión de la demanda de varios periodos. Como conclusión, se determinó que aproximadamente un 60% de las referencias suministradas están bajo la estrategia MTS frente al 40% que siguen la estrategia MTO.

En cuanto a la demanda, se realiza una previsión anual para la que se planifica en base a datos históricos y a las estimaciones de los volúmenes de las referencias básicas y de las especiales (menos comunes) según cliente, realizándose el pedido en firme en cualquier momento del año en curso.

Cada trimestre, se actualiza la previsión considerando el estado de los acuerdos anuales con el cliente. Esta previsión puede tener una variabilidad del 10% aproximadamente y se tienen en cuenta las desviaciones y novedades surgidas a lo largo del periodo.

Respecto a la entrega, por acuerdo con el cliente, los pedidos deben ser solicitados con un plazo de al menos 15 días de antelación a la fecha deseada, que es el tiempo de suministro medio necesario para garantizar la correcta entrega. No obstante, si se dispone del stock necesario, las solicitudes de pedidos se pueden despachar de forma inmediata.

Es posible negociar el tiempo de entrega en caso de pedidos sobrevenidos, si hay cambios de última hora o la cantidad total solicitada supera los márgenes anuales acordados, ya que no se debe perjudicar a clientes de demanda estable en favor de otros con demanda irregular. Sin embargo, la política global de la compañía es la de evitar en la medida de lo posible cualquier tipo de retraso en las entregas, debido al posible impacto que esto tendría en el resto de la cadena de suministro.

A pesar de las diferencias de volumen existentes entre diferentes clientes en lo que se refiere a la demanda, no se considera ningún tipo de prioridad entre ellos.

### 2.3. Descripción técnica de las líneas de fabricación

Como se ha comentado anteriormente, las líneas son sistemas integrales de fabricación donde entra la materia prima y sale el producto finalizado. A continuación, se describen las etapas o proceso productivos de los que constan las líneas:

1. **Entrada de la materia prima.** La fábrica posee una serie de silos que almacenan el plástico granulado que será dosificado a las líneas. El plástico granulado pasa por una tolva y cae en un depósito donde se funde. Ahí se introduce el colorante, dispensado por botes, y se mezcla con el plástico.
2. **Moldeado.** El plástico fundido se inyecta por el molde hasta llegar a las 3 matrices que lo distribuye en la línea. Estas son las que formarán los 3 componentes con los que se forma la pieza final. Cada matriz tiene hasta 48 cavidades disponibles, es decir, cada proceso de moldeado produce de golpe hasta 48 unidades. La realidad es que, por defectos de calidad o deterioro, pueden no estar disponibles las 48 cavidades. Los 3 componentes resultado del proceso de moldeado son:
  - a. Cutter
  - b. Frame
  - c. Lid
3. **Control de calidad.** Las piezas resultantes circulan por la línea y pasan por una cámara donde se monitoriza y controla que cumplen con el estándar de calidad y propiedades necesarias para continuar el proceso de fabricación. Las piezas defectuosas se desechan del sistema y caen a un depósito donde posteriormente se retiran. Estas piezas se tratarán como un desperdicio, ya que, de acuerdo a la normativa de productos alimentarios, las piezas defectuosas no pueden reciclarse reincorporándolas de nuevo al proceso de fundición.
4. **Buffer intermedio.** Para minimizar el posible impacto en la tasa de salida de piezas que puedan surgir debido a las cavidades no disponibles y piezas defectuosas, se usa un buffer de salida intermedio, con unos niveles mínimo y máximo establecidos de unas 1500 unidades.
5. **Ensamblador.** En este módulo se reciben los tres componentes (una pieza de cutter, una de frame y una de lid), se realiza el ensamblado, se sellan y se realiza el grabado, obteniéndose la pieza final. El ensamblador procesa las piezas de diez en diez. Al finalizar, estas piezas pasan por otro control de calidad y de ahí caen a un depósito donde se almacenan hasta completar la producción de la referencia programada en la línea.

A continuación, se muestra un esquema que ilustra la composición de las líneas.

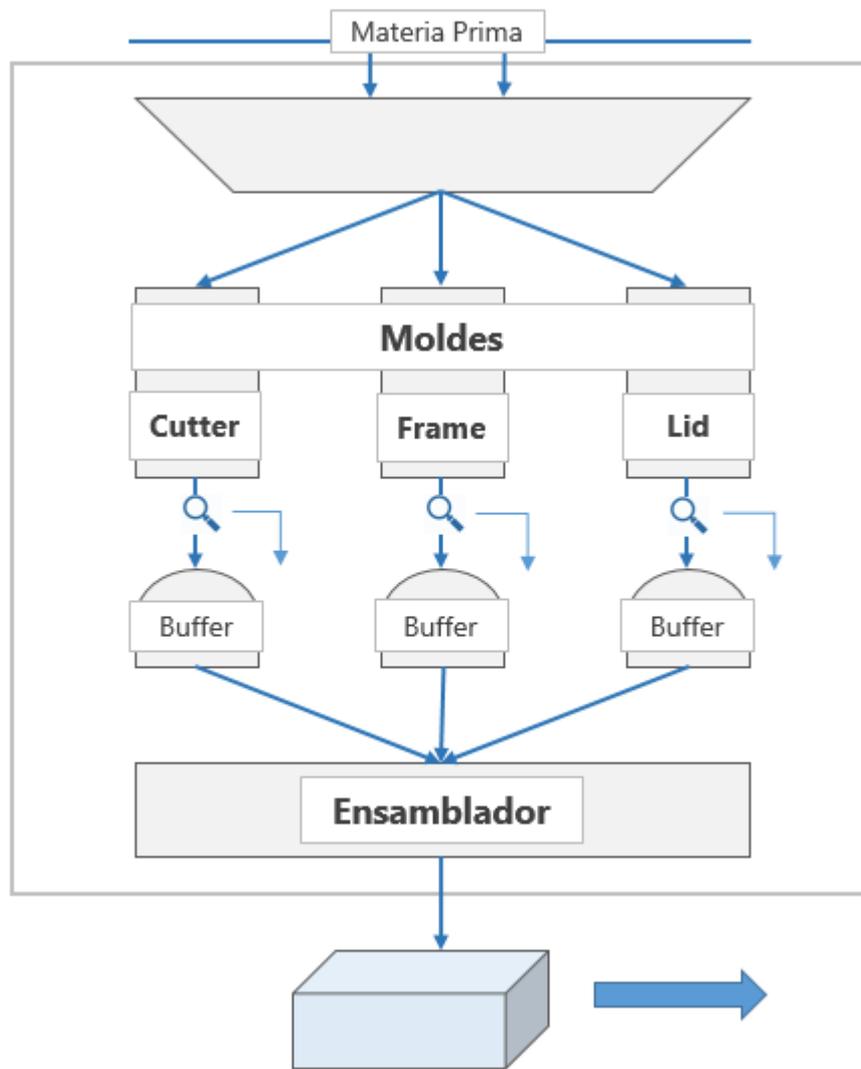


Figura 2.1 Esquema de una línea de fabricación

Las líneas no son interrumpibles, es decir, necesitan un flujo constante de material mientras están en funcionamiento. Además del buffer intermedio, para sincronizar el takt time de todos los procesos, las líneas disponen de modos o cambios de velocidad que permiten ajustarlas a las necesidades operativas. Las posibles paradas que sufren las líneas se deben a los siguientes motivos:

- **Fin de producción o falta de demanda.** La línea tiene un contador de unidades producidas que superan el control de calidad. Cuando se completa el pedido programado se para la línea y se desecha el excedente. Tras esto se procederá a la limpieza y cambio de referencia a producir o bien a apagar la línea. Debido al coste energético, la empresa siempre busca la opción de mantener el menor número de líneas operativas posibles. Para dejar una línea fuera de servicio se debe garantizar que no habrá suficiente demanda prevista por un periodo de al menos un mes.
- **Setup.** Se define setup como el tiempo necesario de preparación de recursos (personal, material) para ejecutar un trabajo (Allahverdi et al., 2008). En nuestro caso, corresponde al cambio de referencia en la línea, que conlleva la parada completa de la línea para la reconfiguración de las matrices y el depurado del depósito de la mezcla de colorante. Adicionalmente, el setup no es anticipable, es decir, no es posible realizar la limpieza del depósito mientras se sigue produciendo aguas abajo. La duración de esta transición va a depender de las diferencias que haya entre las referencias involucradas en el cambio. Aproximadamente, se asume 10 minutos para distintas referencias con el mismo color o similar, una hora para cambios de colores de intensidad claro a oscuro y 4 horas para cambios de

oscuro a claro. Esto es debido a la dificultad de la completa depuración de colores de mayor intensidad para no afectar a las piezas siguientes. Por otro lado, estos cambios conllevan una mano de obra de personal técnico que no pueden simultanear más de un cambio a la vez, además de un límite de cambios durante la duración de la jornada. Por ejemplo, solo serían posible dos cambios de referencias de color oscuro a claro durante la jornada ya que cada una ocupa cuatro horas.

- **Paradas de mantenimiento preventivo.** Una vez al mes, se realizan paradas de mantenimiento en las que se revisa el estado de la línea y se lleva a cabo una limpieza a fondo. Si hubiera material pendiente, se desecha. La duración y frecuencia de estas paradas están establecidas y son conocidas.
- **Paradas no previstas.** Debido a averías, roturas o atascos en alguna parte del flujo productivo.

## 2.4. Planteamiento del problema

Como se ha podido observar, el sistema de producción descrito está automatizado en un porcentaje muy alto de la cadena de valor del producto. Por tanto, hallar las ineficiencias y reducir los costes de producción se torna complicado. No obstante, la compañía ha podido detectar que existe un potencial margen de mejora en un campo, que es en el que nos vamos a centrar; la programación y secuenciación de los pedidos a producir en las líneas de fabricación.

El impacto de una buena o mala programación y secuenciación de los pedidos puede derivar en costes y otros problemas tales como:

- **Energético.** Por mantener en funcionamiento líneas que podrían estar apagadas mientras asumen la carga otras.
- **Setup.** En nuestro caso, los costes de tiempo y recursos que se incurren de realizar un setup son bastantes elevados, por lo que se podría decir que es la causa fundamental que motiva la realización de este trabajo.
- **Inventario.** Una programación de los pedidos poco eficiente puede causar un aumento en los plazos de entrega del material, esto es, el tiempo de reposición del stock del almacén de productos terminados.
- **Rechazo o retraso de pedidos.** Puede causar un elevado coste a la empresa, ya sea por la mala imagen generada, por pérdidas en las ventas, o incluso la potencial pérdida del cliente.

Actualmente, esta tarea se gestiona a través de un módulo dentro del sistema ERP que posee la empresa. A modo de resumen, el método de programación y secuenciación de pedidos mediante este sistema consiste en el procesamiento de los pedidos recibidos semanalmente. El sistema hace lectura de estos y los devuelve como ordenes de fabricación preasignadas a cada línea. La forma en la que se asignan los pedidos a las líneas sigue un algoritmo de optimización interno dentro de este módulo. No obstante, la inteligencia de este algoritmo es limitada y frecuentemente se recurre a la figura del planificador, el cual modifica y adapta a las necesidades reales diarias a los problemas operativos que puedan ir surgiendo.

De este modo, como se ha comentado, se dispone de un sistema informático integrado, pero poco versátil y ágil frente a cambios, ya que la realidad está sujeta a muchas variaciones y a la necesidad de intervenciones del personal cualificado a fin de responder correctamente a las necesidades del día a día.

Otra carencia del sistema actual es la ponderación de los factores que afectan en la secuenciación de los pedidos a la hora de realizar el cálculo de asignaciones, no solo el tiempo de finalización de los pedidos o el ajuste a las fechas de entrega. Estamos hablando en este caso de la importancia de considerar el tiempo de setup, como coste a tener en cuenta y como una ineficiencia en lo que respecta a los tiempos de la operación de

limpieza y puesta a punto de las líneas.

Dado el contexto de la problemática actual, se plantea abordar esta situación mediante técnicas de investigación operativa, elaborando un modelo de optimización de programación lineal mixta-entera (MILP). La naturaleza del sistema de producción nos dice que estaríamos ante un caso de Máquinas Paralelas Uniformes, considerando a las líneas como máquinas y los pedidos recibidos como trabajos. Entonces, tendríamos un modelo con las 12 líneas o máquinas en paralelo para el producto A, combinándose entre ellas para obtener una secuencia óptima de tratamiento de los pedidos. De igual forma ocurriría para las 5 líneas o máquinas necesarias para el producto B.

El hecho de considerar a las líneas como cajas negras y no subdividir las distintas fases productivas que sufre la pieza como otras máquinas (entorno Flow Shop o Job Shop) es debido a que las líneas requieren de un flujo constante de material y solo puede tener un tipo de producto en curso (se debe purgar la línea para comenzar a procesar una nueva pieza). Adicionalmente, un pedido no puede estar segregado entre las distintas líneas. Esta segregación solo nos daría un nivel de complejidad demasiado elevado sin aportar ninguna ventaja.

En cuanto a los trabajos, serían no solo los pedidos específicos del cliente, sino también los de reaprovisionamiento de los niveles de stock del almacén de productos terminados. Estos tendrán un tiempo de proceso en función del lote tratado y la línea en la que se procesa.

A pesar de ser, a priori, un problema de programación y secuenciación simple y de la amplia literatura existente, el tratamiento y tamaño de los pedidos a procesar, las especificaciones técnicas y otras restricciones a considerar aportan un grado de complejidad que resulta de interés en el presente trabajo.

A modo de resumen, se trata pues de un problema de máquinas paralelas uniformes, con tiempos de setup dependientes de la secuencia, mantenimientos preventivos y fechas de entregas. En apartados posteriores se especificarán todas las condiciones concretas del problema.

## 3 ESTADO DEL ARTE

---

En este capítulo se plasmará una breve revisión de literatura existente sobre la resolución del problema de programación y secuenciación para máquinas paralelas bajo distintas circunstancias. Veremos como otros autores han afrontado problemas con particularidades más o menos similares al problema que planteamos y comentaremos las técnicas utilizadas para resolverlos.

La especialización del problema que tratamos en el presente trabajo fin de master, ha dificultado encontrar ejemplos que se ajusten con exactitud a nuestro entorno, el cual concierne un problema de máquinas en paralelo uniformes, tiempos de setup dependientes, mantenimientos preventivos y fechas de entrega, entre otras apreciaciones que se detallarán en el siguiente capítulo. No obstante, podremos encontrar algunas publicaciones interesantes en la literatura, que se presentan en orden cronológico.

El problema de máquinas paralelas es un problema ampliamente estudiado, especialmente en los últimos años, que ha recibido una atención considerable. De partida, cabe decir que el problema de minimizar el tiempo de terminación para dos máquinas paralelas idénticas es NP-Hard (Lenstra, J., Rinnooy Kan, A., & Brucker, 1977). Es por esto que los autores proponen en su mayoría superar las dificultades para la obtención de soluciones para medias y largas instancias usando técnicas metaheurísticas. De este modo, aplican distintos algoritmos de búsqueda de soluciones, analizando cuál de ellos se comporta mejor y arroja mejores resultados en un tiempo computacional asumible. Por otro lado, también tratan de proporcionar diferentes casuísticas y entornos concretos del problema que reflejen situaciones más cercanas a la realidad, a fin de aportar herramientas de aplicación práctica en la industria.

Comenzamos la revisión de la literatura mencionando el trabajo de (McCormick & Pinedo, 1995). En él tratan el problema de programación de máquinas paralelas uniformes con trabajos interrumpibles, bajo los objetivos contrapuestos de minimizar la suma de los tiempos de terminación de cada trabajo,  $\sum C_j$ ; y minimizar el tiempo de terminación del último trabajo que se procesa,  $C_{max}$  o también llamado makespan. Proponen obtener la frontera de Pareto mediante la resolución del problema añadiendo una restricción al modelo con una cota superior del makespan, la cual van modificando. Para afrontar el problema diseñan un algoritmo que se fundamenta en la aplicación de dos reglas de despacho que funcionan para cada objetivo: Menor Tiempo de

Proceso en las Máquinas más Rápidas (SPT-FM) para la minimización de  $\sum C_j$ ; y Mayor Tiempo de Proceso Restante en las Máquinas más Rápidas (LRPT-FM) para la minimización del  $C_{max}$ . Aunque encontraremos la minimización de la suma de los tiempos de terminación de los trabajos en nuestro modelo, nuestro caso no está completamente alineado al planteamiento que aquí se realiza. Sin embargo, resultan de interés las indicaciones realizadas para los trabajos interrumpibles en máquinas con distintas velocidades, ya que posteriormente le dedicaremos un espacio a este asunto.

Una orientación distinta, teniendo en cuenta tiempos de setup dependientes de las máquinas, trabajos no interrumpibles y con comienzo según instante de disponibilidad, se presenta en (Balakrishnan et al., 1999). En este artículo, el objetivo es la terminación de los trabajos justo a tiempo en base a unas fechas de entrega para cada trabajo, penalizando el tiempo de retraso o adelanto según una ponderación de estos. Los autores aportan un modelo de optimización MILP compacto con respecto al modelo estándar que hasta entonces existía en la literatura y se compara su rendimiento para pequeñas instancias. Este modelo presenta una particularidad en su formulación, ya que utiliza variables de decisión de dos índices ( $x_{ij}$ ) frente a las usadas comúnmente en la literatura, de tres índices ( $x_{ijm}$ ). Esta propuesta reduce considerablemente la cantidad de variables binarias. Como adelanto, en el presente trabajo se ha optado por utilizar este tipo de variables.

Hasta ahora se han comentado modelos para máquinas paralelas en los que el tiempo de proceso está relacionado con las máquinas, normalmente asociado a una velocidad, pero es posible que esta relación no exista. Este tipo de modelos son clasificados como máquinas paralelas no relacionadas y, en la práctica, reciben un tratamiento similar a las máquinas paralelas uniformes. (Kim et al., 2003) recoge un caso de este tipo, incluyendo tiempos de setup y agrupación de trabajos en lotes, con el objetivo de minimizar la tardanza total de los trabajos en base a unas fechas de entrega. Cada lote consta de un número fijo de trabajos y tienen en común el tiempo de proceso y la fecha de entrega. Se propone un análisis comparativo de los resultados usando 4 métodos de búsqueda: las reglas de despacho EWDD (el trabajo con fecha de entrega más reciente, ponderada, se procesa primero) y WSPT (el trabajo con menor tiempo de proceso ponderado se procesa primero), una heurística en “dos niveles” para las asignaciones de los lotes y sus trabajos, y la técnica heurística Simulated Annealing (SA). Las reglas de despacho mostraron peor rendimiento debido a que funcionan para la asignación de los lotes, pero no en un segundo nivel para la asignación de trabajos de estos lotes. La heurística en “dos niveles” se mostró capaz de obtener buenos resultados rápidamente para una cantidad de trabajos menor a 100, sin embargo, en líneas generales SA es la heurística que supera al resto de las propuestas.

En un contexto industrial surgen muchas situaciones sobrevenidas, como posibles máquinas no disponibles por averías, mantenimientos o bajadas en el rendimiento que requieren ser consideradas, para lograr resultados válidos. (Xu et al., 2006) aborda el problema de máquinas paralelas con paradas de mantenimiento cada cierto número de periodos, pero no sincronizadas entre las máquinas. Con el objetivo de minimizar el makespan, desarrolla un algoritmo híbrido. Por un lado, realiza una aproximación al problema de la mochila en el que se establecen conjuntos de trabajos para ser encajados en los intervalos en los que se producen las paradas de mantenimiento (siguiendo la regla Best Fit Decreasing). Por otro lado, asigna estos conjuntos usando la regla de despacho Mayor Tiempo de Proceso (LPT). La composición de ambos algoritmos aporta buen rendimiento para instancias de 30 trabajos y 9 máquinas.

Para tiempos de setup dependientes, fechas de entrega en máquinas paralelas no relacionadas y buscando minimizar el makespan junto con la tardanza en los trabajos, (Rocha et al., 2008) exponen dos modelos MIP y un algoritmo de Branch & Bound para una resolución exacta. Además, realizan una comparativa del rendimiento entre ellos. El primer modelo está basado en el formulado por (Manne, 1960) para entornos job shop y el segundo modelo en el que añade una nueva dimensión en las variables de decisión identificando la posición que ocupa cada trabajo en cada máquina. Para el algoritmo de Branch & Bound customizado los autores utilizan la heurística GRASP y la regla de despacho EDD (Earliest Due Date) en las etapas de

inicialización y ramificación respectivamente. Para el análisis y realización de pruebas, establecen varios tamaños del problema: 4 máquinas con 4-12 trabajos, y 6 máquinas con 6-25 trabajos. Los modelos fueron resueltos usando el solver CPLEX. Para medianas instancias, el algoritmo B&B demostró mejor rendimiento al compararlo con GRASP, logrando alcanzar el óptimo hasta para 30 trabajos con una duración de ejecución de 2 horas. Sin embargo, al aumentar el tamaño de las instancias GRASP logra alcanzar mejores resultados.

(Mazdeh et al., 2010) consideran un problema que incluye el deterioro de las máquinas. Sugieren una ecuación que calcula un deterioro lineal en función de la tasa de salida. Para la resolución, marcan dos objetivos: minimizar la tardanza de los trabajos y el coste por deterioro de las máquinas. Elaboran un análisis comparativo donde enfrentan la resolución exacta mediante Lingo y la búsqueda del óptimo con la técnica de búsqueda Tabú. Para tamaños pequeños la resolución óptima es viable. Aumentando el tamaño se observa como la resolución exacta requiere varias horas mientras la búsqueda Tabú arroja buenos resultados en un tiempo más reducido.

Otro aspecto interesante a considerar para una aplicación en entornos reales es la inclusión de recursos limitados para la producción. (Yeh et al., 2015) exponen un estudio para máquinas paralelas uniformes con el objetivo de minimizar el makespan, donde existe un límite de consumo de los recursos necesarios para la programación que se pretende obtener. En el artículo se aplica un algoritmo genético, un algoritmo de optimización por enjambre de partículas y un algoritmo de optimización por enjambre simplificado. Se experimenta con instancias de 50 hasta 1000 trabajos y se llega a la conclusión de que no existen diferencias sustanciales en el rendimiento de las tres heurísticas, aunque el algoritmo genético se comporta mejor cuando el número de trabajos crece.

(Wang & Liu, 2015) profundizan en el campo de la limitación de recursos y lo relacionan con los mantenimientos preventivos. Contemplan dos tipos de recursos: un recurso para la máquina y otro recurso para el molde, que está asociado con la máquina. El objetivo es minimizar, simultáneamente, el makespan y la no disponibilidad de las máquinas y los moldes. En el artículo se desarrolla un modelo de optimización no lineal con tres objetivos y aplica una metaheurística basada en la técnica NSGA-II (algoritmo genético de clasificación no dominado) que integra la programación de la producción y la planificación de los mantenimientos preventivos en las máquinas y moldes simultáneamente. Para nuestro caso de estudio, se desacopla la planificación de los mantenimientos preventivos, siendo dado como parámetro de entrada al modelo.

En el artículo presentado por (Avalos-Rosales et al., 2018) se estudia un entorno de máquinas paralelas no relacionadas con setup y con mantenimientos preventivos en el que se busca minimizar el makespan. Estas operaciones de mantenimiento se asumen programadas a intervalos de tiempo fijos ya predefinidos, una situación muy común en la industria. Para el modelo lineal de programación MILP, los intervalos entre mantenimientos se tratan como bloques con los que separan las variables de decisión de asignación de trabajos y máquinas. Las propias tareas de mantenimiento son tratadas como un trabajo dummy de inicio de tales bloques. Frente a la imposibilidad de encontrar el óptimo para grandes instancias, los autores desarrollan un algoritmo Multi-Start (MSA) como alternativa. Los experimentos realizados muestran como la resolución exacta del modelo de optimización es capaz de encontrar el óptimo para un máximo de 25 trabajos y 6-8 máquinas. El MSA diseñado para grandes instancias, muestra un rendimiento similar para instancias pequeñas y medianas. Además, los autores adaptan otras metaheurísticas de la literatura como GRASP y un Multi-Start basado en Vecindad Variable Descendente (VND, del inglés Variable Neighborhood. Descent) para problemas sin las actividades de mantenimiento y los resultados reafirman el MSA como mejor técnica para abordar el problema. Un enfoque parecido propuesto, tratando las actividades de mantenimiento como trabajos es el que se ha adoptado en este trabajo, como se expondrá en el siguiente capítulo. Sin embargo, debido al corto horizonte de programación, en el cual es improbable que se den dos mantenimientos programados para la misma línea, y dado que las tareas de mantenimiento pueden no ser constantes, se ha descartado la consideración de los mismos tal como lo plantean los autores.

Volviendo a las máquinas paralelas no relacionadas con setup, encontramos un estudio elaborado por (Fanjul-Peyro et al., 2018). Los autores plantean una reformulación del problema para la minimización del makespan con la que se mejoran considerablemente los resultados de la literatura encontrada hasta entonces: soluciones muy cercanas al óptimo en instancias del tamaño de hasta 400 trabajos y 4 máquinas. Esta gran mejora viene explicada, en parte, por la calidad y la potencia de los solvers actuales (demuestra mejores resultados de Gurobi frente a CPLEX), y a que desarrollan un modelo más robusto, inspirado en el conocido problema del viajero, gracias a la analogía que aportó (Pinedo, 2005). Adicionalmente, utilizan un algoritmo basado en la descomposición del modelo con el que se pueden obtener soluciones cercanas al óptimo para 1000 trabajos y 8 máquinas. El algoritmo se fundamenta en la aplicación de cortes en el problema relajado, en una fase de asignación de trabajos a máquinas, seguido de otra fase posterior de secuenciación, que se aplican de forma iterativa hasta alcanzar la solución óptima o el límite de tiempo establecido.

Otro ejemplo del caso anterior lo vemos en (Báez et al., 2019). Formulan un modelo de programación basado en la propuesta que plantearon en (Baez et al., 2016), que de nuevo considera la estructura del problema del viajero. Sin embargo, conociendo que el modelo de optimización arroja resultados óptimos para instancias limitadas, los autores se centran en desarrollar un algoritmo híbrido (HMA) de dos técnicas metaheurísticas GRASP y VNS. El algoritmo se compara frente a los mejores resultados reportados hasta ese momento, obtenidos con la técnica Iterated Greedy (IG), superándolos en rendimiento.

En el caso que tratamos, debido a cómo está diseñado el sistema productivo, en el que la prioridad es satisfacer la demanda en todo momento, puede haber momentos de caídas en la producción o valles en los que no sea necesaria la utilización de la totalidad de la capacidad de fabricación de las líneas. Esta casuística y los costes derivados de mantener una línea en funcionamiento, han motivado el no descartar las paradas de las líneas durante la programación. Sobre este aspecto, encontramos la publicación (Osorio-Valenzuela et al., 2019) donde ahondan sobre este tema para máquinas paralelas idénticas. Proponen minimizar las máquinas utilizadas bajo un entorno con limitaciones en la capacidad de producción de cada máquina y unas ventanas de disponibilidad en la que los trabajos tienen que ser procesados, expresadas con los comunes términos de instante de disponibilidad y de entrega. Los autores proporcionan dos formulaciones matemáticas para este problema y tres procedimientos de resolución. Desarrollan y comparan las heurísticas First Fit y Best Fit, sin diferencias notables entre ellas, y un algoritmo exacto de Branch & Price, que se compara con la resolución del modelo mediante CPLEX.

Otro aspecto a tener en cuenta en nuestro problema es el balanceo de la carga de trabajos en las máquinas. Aunque es otro campo que, en cierta medida, se aleja de la línea de la programación de la producción, es interesante prestarle atención como veremos más adelante. (Christ et al., 2019) realizan una investigación sobre este asunto para máquinas paralelas uniformes. En primera instancia exponen un modelo Min-Max de equilibrado de la carga de trabajo (MMFWB) y deducen propiedades de las soluciones obtenidas. Seguidamente, proponen un procedimiento iterativo de búsqueda en el que se obtiene el óptimo a través de la construcción de la solución mediante subconjuntos de máquinas. Posteriormente, lo someten a experimentos, mostrando que es capaz de resolver problemas de grandes tamaños (cientos de productos y hasta 20 máquinas).

Por último, en un reciente artículo, (Fanjul-Peyro, 2020), los autores tratan el problema de máquinas paralelas no relacionadas con tiempos de setup y limitación de recursos desde una perspectiva aún más cercana a la realidad de nuestro caso. Definen unos recursos auxiliares a la producción, específicos para procesar los trabajos, otros específicos para realizar el setup, y otros no específicos compartidos para ambas tareas, todos ellos reutilizables tras ser empleados. Además, establecen unos recursos máximos a distribuir entre las máquinas de forma simultánea. Esto implica una dificultad añadida no solo porque hay que asignar y secuenciar estos recursos, sino también indicar en que instantes están o no disponibles, ya que la escasez de los mismos provoca paradas en la producción. Para la formulación del modelo, toman como referencia el conocido modelo MILP para el problema de máquinas no relacionadas con tiempos de setup, adaptado con las

limitaciones de los recursos. Con el planteamiento que los autores realizan, se tiene un modelo válido para cualquier combinación de los tipos de recursos mencionados, siendo posible añadir recursos nuevos con sus respectivas restricciones. Planteado el modelo, desarrollan un algoritmo para resolverlo de forma óptima en el que dividen el problema en tres fases: asignación de trabajos a máquinas (con el modelo MILP relajado, sin considerar las restricciones de tiempos ni capacidad de recursos), secuenciación del orden de los trabajos y factibilidad de recursos para los tiempos de proceso y setup coordinados (con el modelo MILP completo y la programación con restricciones para los recursos). La resolución fue realizada con CPLEX y Gurobi (al ser enfrentados, Gurobi muestra ser mejor), para pequeñas, medianas y largas instancias, considerando entre 10 400 trabajos y entre 4 y 8 máquinas. Se establece un límite de 3 horas con el que se logra el óptimo para pequeñas instancias. A partir de 30 trabajos y teniendo en cuenta todas las limitaciones en los recursos, el problema resulta no factible. Se demuestra como los recursos compartidos y el tiempo de setup son los que complican en mayor medida la obtención de buenos resultados para el modelo. En el caso que nos ocupa, sería posible considerar este tipo de restricciones para los setup y los mantenimientos preventivos, dado que el equipamiento material y la mano de obra del personal necesario para la limpieza y preparación de las líneas es limitado, sumado al hecho de que puedan ser necesarios simultáneamente en varias líneas.

Como hemos podido ver, las publicaciones comentadas son de diversa índole y, aunque no se ajustan exactamente al caso de estudio, engloban un conjunto de situaciones que sirven de contextualización del estado del arte para este problema de programación. Una de conclusiones que se pueden sacar de esta revisión es que los modelos de programación matemática no pueden competir con las potentes y variadas técnicas metaheurísticas que han sido diseñadas para resolver este tipo de problemas cuando se tratan instancias de gran tamaño. No obstante, en este estudio se abordará un enfoque de resolución exacto ya que se manejan instancias de tamaño medio.



---

## 4 MODELOS MATEMÁTICOS

---

**E**n este capítulo se desarrollarán los modelos matemáticos de programación lineal con los que se propone resolver el problema descrito en la segunda sección. Para reducir el grado de complejidad, se elaborará una descomposición del problema en dos etapas. Con el primer modelo se pretende fijar para cada línea una cantidad limitada de colores garantizando un equilibrado entre las mismas. El segundo modelo corresponde al problema de programación, la estrategia descrita aporta una nueva perspectiva respecto a la literatura comentada anteriormente.

Previamente al desarrollo de la solución del problema, se deben valorar y tener en consideración una serie de aspectos que son fundamentales para la correcta elaboración de una propuesta que permita resolver de forma efectiva el problema en estudio. A continuación, se comentarán estos puntos a modo de introducción antes de profundizar en los modelos.

Cuando decimos que el problema que se pretende atajar está enfocado en la programación de la producción, estamos delimitando y definiendo el problema a una decisión sobre un proceso productivo concreto dentro del sistema de producción de la planta. Este sistema productivo comprende distintos ámbitos compuestos por distintos procesos productivos y la toma de decisiones en alguno de ellos puede tener implicaciones en el resto. Como contextualización, cabe mencionar la clasificación y relación entre los procesos productivos según las etapas productivas y las frecuencias temporales realizada por (Hartmut Stadtler, 2005) que se muestra en la siguiente figura:

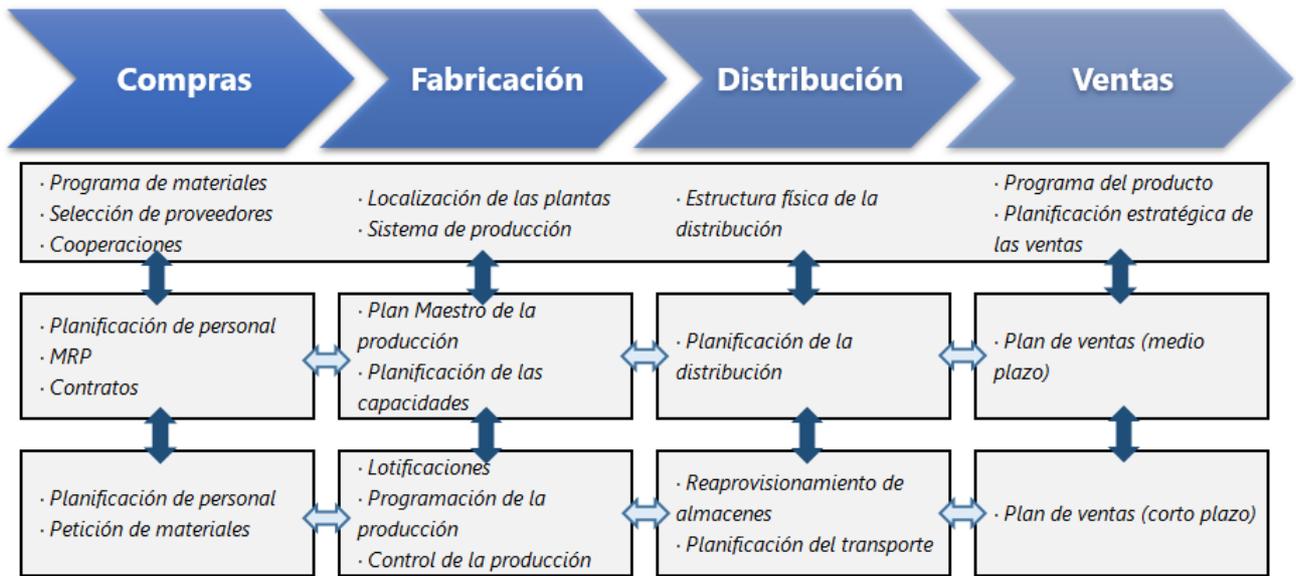


Figura 4.1 Clasificación de los procesos del sistema productivo (Hartmut Stadler, 2005)

Como se puede observar, la programación de la producción corresponde a una decisión puramente operativa, ya que se debe conocer en todo momento que productos se deben de procesar y en qué orden de forma que se pueda cumplir con lo que demanda la cadena de suministro y con las entregas comprometidas a cliente. Acompañando a la programación de la producción se encuentra el control de la producción, que trata de dar solución a las incidencias e imprevistos que surgen en el día a día. La solución que se propone deberá contribuir también a facilitar el manejo de estas incidencias, proporcionando soluciones en un corto plazo de tiempo, de manera que sea posible el lanzamiento de reprogramaciones.

La situación actual de la compañía parte de un sistema de programación de la producción que propone la ejecución de órdenes de fabricación con una actualización semanal. Cualquier desajuste sobre la programación pasa por la figura del planificador, que se encarga de reprogramar la propuesta dada por el sistema absorbiendo toda la información nueva que va surgiendo durante la propia semana.

Uno de los desajustes sobre la programación que aparece de forma recurrente son las paradas para el mantenimiento preventivo de las líneas. Normalmente, la secuencia de producción y la programación del mantenimiento se han tratado por separado en la industria. Estas dos tareas entran en conflicto ya que las actividades de mantenimiento consumen tiempo de producción, mientras que retrasar el mantenimiento hasta que se disponga de tiempo ocioso en el proceso de producción puede aumentar la probabilidad de fallo de las líneas. Por este motivo parece más apropiado tratar ambas tareas simultáneamente. Esto se tendrá en consideración introduciendo en el horizonte a programar las propuestas de mantenimiento de cada línea. En el caso de paradas imprevistas debidas a averías o roturas, podrán extraer del modelo las líneas afectadas y lanzar una nueva reprogramación.

En un primer momento, otras decisiones que se consideraron incluir en la formulación del problema son las de establecer los niveles de stock para ciertas referencias y la planificación de la producción a nivel táctico. Esto se explica ya que los niveles de stock se calculan teniendo en cuenta unos tiempos de aprovisionamiento, que pueden verse afectados si la programación de la producción logra reducir significativamente el tiempo de terminación de los pedidos. No obstante, este tipo de interacciones se escapan del objetivo primordial marcado, por lo que quedan fuera de este estudio.

Otro ámbito que se ve directamente afectado por la programación corresponde a los recursos humanos y materiales necesarios para llevar a cabo la producción. Dado que en nuestro caso existe capacidad suficiente de almacenamiento de materias primas y se dispone de personal cualificado, no se considera necesario estudiar el

problema junto a los recursos. La inclusión de éstos, al igual que se ha comentado antes, implica aumentar la abstracción del problema, abarcando gran cantidad de factores, a menudo estocásticos, dificultando enormemente la formulación y la resolución del problema planteado.

En definitiva, se establece para nuestro problema un horizonte temporal de programación bisemanal, acorde a los 15 días de margen para el suministro de los pedidos, pero ejecutable en cualquier momento de la semana, permitiendo tantas reprogramaciones como sean necesarias. La solución que se propone se adapta de forma más precisa a los requisitos del sistema de producción, ya que incluye aspectos que el actual sistema informático no contempla.

El modelado de un problema real como este, presenta a menudo dificultad en el proceso de abstracción e interpretación de las características y comportamientos que lo definen. Otro punto a tener en cuenta es si la compañía con la que se trabaja es capaz de proporcionar los recursos y datos necesarios que requiere la implementación de la herramienta desarrollada. En definitiva, existen una serie de riesgos o discrepancias que pueden surgir durante todo este proceso que hay que mitigar en la medida de lo posible.

(Hillier, F. S. and Lieberman, 2010) presenta unas pautas a seguir para una correcta elaboración de una solución que satisfaga las necesidades en calidad, tiempo y forma que podemos tomar como referencia. Estas se pueden resumir como:

- Una primera etapa de búsqueda y comprensión de los datos. Es básico para poder plantear las restricciones y objetivos apropiados, tener a disposición todos los datos posibles y extraer la información de ellos. De lo contrario, podríamos estar resolviendo un problema que no se corresponde con lo que la realidad requiere. Como ya se comentó en el apartado 2, se cuenta con un MRP y otros archivos de gestión del planificador para la obtención de los datos de los pedidos y las líneas disponibles. Otros parámetros como la estandarización de tiempos de setup por colores y las tasas de salida han conllevado un mayor esfuerzo. Más adelante se profundizará en estos aspectos.
- Todo este conjunto de datos hay que trasladarlo adecuadamente a un lenguaje que adopte las características y el comportamiento del problema. Como ya sabemos, en nuestro caso se trata de una formulación matemática con una estructura basada en expresiones matemáticas con la finalidad de aplicar un método de resolución exacto. En mayor detalle podemos decir que, conseguida la información que define el problema, estamos ante un modelo determinista de programación lineal mixta-entera en el que se aplican una serie de restricciones que delimitan las limitaciones del sistema.
- En muchos casos, la complejidad de los problemas reales y los volúmenes de datos que tienen que contemplarse y relacionarse hacen imposible resolver el problema desde un enfoque exacto. En esta etapa hay que poner en valor y determinar, en primer lugar, si es viable plantear la búsqueda de una solución óptima y, en segundo lugar, y si la respuesta es afirmativa, la ventaja que supone el tiempo de resolución para lograr el óptimo frente a la calidad de la solución obtenible por otros medios menos ambiciosos. Cabe recordar la programación lineal entera, con algoritmos exactos como el Branch & Bound, que son capaces de lograr el óptimo para instancias de tamaño mediano, y otros algoritmos aproximados como las heurísticas y metaheurísticas. Para el caso a tratar, ya hemos adelantado la propuesta de dos modelos de programación lineal, con un primer modelo que tiene como finalidad reducir el orden de complejidad del segundo modelo de programación de la producción.
- Ya nos hayamos decantado por un método de resolución o por varios, un análisis posterior para corroborar la validez de las soluciones es imprescindible. Los resultados obtenidos dependen en parte de los parámetros de entrada, que puede ser distintos o aleatorios en cada ejecución. Para garantizar que la solución propuesta es consistente y fiable debemos someterlo a distintas pruebas y experimentos y comprobar que tiene una respuesta si no buena, coherente, en cada ejecución. Para ello, lo apropiado es comenzar con pequeñas instancias e ir escalando a grandes tamaños. Aunque el tamaño del problema es conocido, se experimenta la capacidad para resolver tamaños de instancias

superiores para la que estaría diseñado en un primer momento y se variarán los parámetros de entrada con el objetivo de poner a prueba la reacción ante los casos más ajustados que se pudieran dar en la realidad.

- El entorno o realidad que modelemos puede variar con el tiempo. Una vez la implantación haya sido satisfactoria, hay que tener en cuenta en esta última etapa la adaptación a nuevas restricciones o funcionalidades y el mantenimiento de la solución dada. En este sentido, ya se mencionó anteriormente que el modelo debe adaptarse a distintas reprogramaciones.

Tras situar el problema en el sistema de producción de la empresa y entender las repercusiones de seguir una correcta metodología para obtener una solución que satisfaga al cliente, se va a detallar la propuesta de aplicación en dos fases de este trabajo fin de master, con sus respectivos modelos matemáticos. En adelante, nos centraremos únicamente en el entorno del producto A, ya que comparte, características con el producto B. La diferencia la encontramos en los tipos de instancias (número de líneas, pedidos, etc.), por lo que sería redundante repetir el planteamiento para el caso de B.

## 4.1 Modelo Preliminar. Equilibrado y pre-asignación de colores a líneas

De la revisión de la literatura existente hemos podido sacar en claro que la resolución del problema planteando un modelo de programación exacto para máquinas paralelas con tiempos de setup requiere de un tiempo de ejecución muy elevado a partir de unos 20-30 trabajos a procesar. Si a esto le sumamos que este caso concreto posee ciertas particularidades que aumentan el grado de complejidad, no se puede tener la certeza de obtener la solución óptima en un tiempo asumible.

Hemos visto en diversas publicaciones como los autores suelen optar por aplicar técnicas metaheurísticas, sin embargo, dado que el número de pedidos no es excesivo, en esta ocasión se va a abordar desde otro punto de vista. A continuación, se va a desarrollar un primer modelo con el que se consigue una simplificación del problema de programación, con el fin de reducir el número de variables y restricciones y, por lo tanto, el espacio de soluciones a explorar.

En el capítulo 2 sobre la descripción del problema, decíamos como este consiste en la programación y secuenciación de pedidos de dos tipos de productos, con multitud de referencias posibles para cada uno, según unas propiedades que afectan a los tiempos de preparación, o setup, entre dos pedidos consecutivos de distintas referencias. La propiedad más relevante y que verdaderamente tiene impacto en el tiempo de setup es el color de la pieza, debido a que el cambio de color es el causante de que se necesite invertir tiempo para la puesta en servicio de la línea. Si varía la referencia, pero comparten un mismo color, la preparación de la línea se reduce a un cambio de configuración de las matrices del molde y del grabado, lo cual implica un menor tiempo de setup. No obstante, como veremos en el siguiente modelo, el tiempo en las transiciones entre colores no es homogéneo para todos los colores.

En este primer planteamiento, se clasifican los pedidos según el color al que pertenece. Las cantidades a producir de cada color en cada pedido se agrupan para disponer del total a procesar de cada color, el cual debe ser asignado a una o varias líneas. Se plantea un modelo donde el objetivo es el equilibrado de las cargas de las líneas, repartiendo los totales de cada color en una o varias líneas, según convenga. Adicionalmente, se permite la no activación de algunas líneas en caso de no ser necesario. De este modo, al permitir al modelo que contemple líneas paradas, pueden repercutirse unos importantes ahorros en el consumo energético.

Se trabaja en un contexto donde se busca una herramienta que aporte soluciones en tiempos de computación pequeños, que permita continuas reprogramaciones en un horizonte rodante. Para ello, es necesario que se refleje la información que caracteriza la situación actual en el momento de la ejecución del modelo. Entre estos

datos tenemos el estado de las líneas en términos de producción en curso y productividad.

Para este primer modelo, que pretende obtener que colores se deben fabricar cada línea, se debe considerar el color que en el momento de la ejecución estén procesando cada línea. Además, sin olvidar el objetivo de reducir el setup, dar continuidad en las líneas a los colores que se estén procesando puede ser recomendable. Por estos motivos, de partida se asigna el color en proceso a la línea. Como a su vez se permite la asignación de cada color a varias líneas, podemos decir que no se obliga la asignación de un color a una línea en la que quede un pequeño remanente por producir.

Para la productividad se tiene una tasa de salida calculada para cada línea que afecta al equilibrado de estas. Hay que tener en cuenta que un equilibrado de las cargas de fabricación busca la menor diferencia posible entre la línea que más produce frente a la que menos, por lo que si tenemos líneas asimétricas con diferentes velocidades hay que expresar el equilibrado en función de la eficiencia de cada línea. Asimismo, no utilizar las líneas con mejor rendimiento en el momento actual, podría derivar en un equilibrado en el que se dejen inactivas las líneas más veloces.

Para subsanar esta carencia, se resolverá el modelo de forma escalonada en dos etapas. En una primera etapa, se optimizará puramente el equilibrado de las líneas. De esta primera etapa se almacena el número de líneas activas que el modelo ha considerado necesarias para tal equilibrio. En la segunda etapa, se fuerza el número de líneas activas al obtenido previamente y se minimiza la producción de la línea más sobrecargada, de modo que el modelo seleccionará las líneas con mejor rendimiento. Al resolverlo de forma escalonada y no con ambos objetivos simultáneamente, evitamos el riesgo de reducir las líneas en exceso, provocando la congestión del sistema.

#### 4.1.1 Formulación matemática del modelo preliminar

Tras detallar las condiciones que rodean al modelo, los datos de entrada de los que se parte son los siguientes:

- Un conjunto de  $NC = \{1, \dots, i, \dots, |NC|\}$  colores para ser asignados a las líneas.
- Un conjunto de  $L = \{1, \dots, l, \dots, |L|\}$  líneas.
- $lcm$  es un parámetro que indica el número de líneas a las que se puede asignar un mismo color.
- Para cada color  $i$  con pedidos en curso (en producción) se define un subconjunto de líneas  $EC_i = \{1, \dots, l, \dots, |EC_i|\}$  a las que puede ser asignado. Por consiguiente,  $EC_i = \{\emptyset\}$  para los colores  $i$  sin pedidos en curso.
- $qec_l$  es la cantidad restante que falta por producir del pedido actualmente en curso en la línea  $l$ .
- $ct_i$  es la cantidad total de color  $i$  para asignar en las líneas.
- $ts_l$  es la tasa de salida para la línea  $l$ .
- $cte$  es una constante de valor alto que se precisa en las restricciones del modelo. Se ha tomado como  $cte = 1000 \cdot \sum_{i \in NC} ct_i$ . De esta forma, como se verá luego, sumando la cantidad de todos los colores, se garantiza una constante inalcanzable.
- $cap$  es la capacidad promedio de fabricación del sistema para el horizonte temporal establecido. Se

obtiene del cálculo  $cap = \frac{tom \cdot \bar{ts}}{1000}$ , donde  $\bar{ts}$  es la tasa de salida promedio para las líneas utilizables

y  $tom$  es el tiempo operativo máximo del que se dispone en el horizonte temporal. Al funcionar las líneas 24 horas, se obtiene  $tom = 24 \cdot 15 = 360h$ .

- $it$  es el valor de la etapa de resolución del modelo. Tomará el valor cero en la primera etapa, y uno en la segunda.
- $lp$  es el número de líneas resultado de ejecutar el modelo en la primera etapa. Por tanto, en la primera etapa este parámetro no aparece.

En cuanto a las variables de decisión, se plantean para el modelo:

- $C_{il} = 1$  si el color  $i$  es asignado a la línea  $l$ . En caso contrario toma el valor cero.
- $Q_{il} \geq 0$  es la cantidad de color  $i$  asignada a la línea  $l$ .
- $\partial_l = 1$  si la línea  $l$  se activa para procesar pedidos. En caso contrario toma el valor cero.
- $U \geq 0$  es la cota inferior para la evaluación del equilibrado de las líneas. Es la cantidad asignada a la línea que menos procesa.
- $V \geq 0$  es la cota superior para la evaluación del equilibrado de las líneas. Es la cantidad asignada a la línea que más procesa.

La formulación del modelo quedaría de la siguiente forma:

$$Z_1 = \text{Min } V - (1 - it)U \quad (1.1)$$

s.a.:

$$C_{il} = 1, \quad i \in NC, \quad l \in EC_{i|EC_i|>0} \quad (1.2)$$

$$Q_{il} = qec_l, \quad i \in NC, \quad l \in EC_{i|EC_i|>0} \quad (1.3)$$

$$\sum_{l \in L} C_{il} \geq 1, \quad i \in NC \quad (1.4)$$

$$\sum_{l \in L} C_{il} \leq lcm, \quad i \in NC \quad (1.5)$$

$$\sum_{l \in L} Q_{il} = ct_i, \quad i \in NC \quad (1.6)$$

$$C_{il} \leq \partial_l, \quad i \in NC, \quad l \in L \quad (1.7)$$

$$Q_{il} \leq ct_i \cdot C_{il}, \quad i \in NC, \quad l \in L \quad (1.8)$$

$$\frac{1000}{ts_l} \cdot \sum_{i \in NC} Q_{il} \geq U - cte \cdot (1 - \partial_l), \quad l \in L, \quad it = 0 \quad (1.9)$$

$$\frac{1000}{ts_l} \cdot \sum_{i \in NC} Q_{il} \leq V, \quad l \in L \quad (1.10)$$

$$\sum_{i \in NC} Q_{il} \leq cap, \quad l \in L \quad (1.11)$$

$$\sum_{l \in L} \partial_l = lp \quad it = 1 \quad (1.12)$$

$$C_{il} \in \{0,1\}, \quad \partial_l \in \{0,1\} \quad (1.13)$$

$$Q_{il} \geq 0, \quad U \geq 0, \quad V \geq 0 \quad (1.14)$$

En el modelo, la expresión (1.1) muestra la función objetivo. Aplica la diferencia entre los valores de las líneas más ( $V$ ) y menos ( $U$ ) cargadas. En este último se agrega un producto para alterar la función objetivo dependiendo de la etapa, cambiando del equilibrado a la minimización de la carga máxima.

(1.2) y (1.3) son las restricciones que preestablecen los pedidos en curso en el modelo, asignando el color en la línea y añadiendo la cantidad correspondiente a lo que resta por procesar, respectivamente. Previamente, se ha definido para cada color los subconjuntos  $EC_i$  con las líneas en las que se está produciendo, de forma que si los colores no tienen producción en curso las restricciones no se aplican. Para que el modelo sea factible, el máximo de colores asignables en las líneas que se hayan establecido debe ser igual o superior al color que tenga más pedidos en curso, por tanto, la cardinalidad de estos conjuntos debe cumplir:  $|EC_i| \leq lcm$ .

En la restricción (1.4) se fuerza que todos los colores se asignen en, al menos, a una línea. A su vez, (1.5) impide que se supere el número de asignaciones a líneas por encima del máximo establecido. La restricción (1.6) define la cantidad total a repartir de cada color. En (1.7) se establece la relación de la asignación de colores a líneas con la activación de las líneas, de forma que si se ha asignado algún color obliga a que la línea a la que se asigne esté activa. (1.8) limita que la cantidad de color  $i$  repartido en cada una de las líneas supere el total por procesar de ese color, incluyendo la cantidad restante de los pedidos en curso.

Las restricciones (1.9) y (1.10) son las que calculan la cota inferior y superior de carga de las líneas, respectivamente. Ambas cotas se expresan en cantidad producida por unidad de tiempo y, para ello, se utiliza la tasa de salida para convertir las cantidades asignadas. La cota inferior solo se aplica en la primera etapa ya que para la segunda es irrelevante. Utiliza una constante de valor alto que se tendrá en cuenta cuando la línea no esté activa, de esta forma no se valoran como cota inferior las líneas descartadas para producir. En (1.11) se impone que la carga de la línea supere el horizonte de programación.

Por último, la restricción (1.12) se aplica solo en la segunda etapa y es la que fuerza el uso del número de líneas obtenido en la primera etapa.

En (1.13) y (1.14) se definen las variables binarias y continuas, respectivamente.

Tras resolver el modelo de asignación de colores a las líneas se obtienen varios subconjuntos de pedidos a secuenciar en cada una de las líneas permitidas. Una característica de este resultado es que se pueden generar subconjuntos independientes de líneas, a las que le pertenecen una serie de pedidos exclusivos. Es decir, se pueden obtener clústers de líneas con respecto a los colores procesados. De esta forma, se tiene un sistema compuesto por grupos de líneas, lo que permite el resolver el modelo de programación para cada uno de los grupos independientemente.

Por ejemplo, si se parte de 6 colores y 12 líneas posibles y, el resultado de aplicar el modelo proporciona una asignación de los colores como:  $C_1 \in \{L_1, L_2\}$ ;  $C_2 \in \{L_2, L_3\}$ ;  $C_3 \in \{L_1, L_3\}$ ;  $C_4 \in \{L_5, L_7\}$ ;  $C_5 \in \{L_5, L_{10}\}$ ;  $C_6 \in \{L_{12}\}$ , podemos agrupar las líneas  $\{L_1, L_2, L_3\}$ ,  $\{L_5, L_7, L_{10}\}$  y  $\{L_{12}\}$  en grupos que pueden trabajar de manera independiente, ya que los colores de tales grupos no se comparten.

El caso ejemplificado aporta una forma eficiente de resolver el problema, pero conviene indicar que no se puede garantizar en la mayoría de las ocasiones. A pesar de ello, la sólo asignación de colores a líneas aporta una reducción importante en la complejidad del problema.

La justificación de este modelo preliminar se debe, por un lado, a que una línea de fabricación es un complejo sistema cuyo funcionamiento le supone a la empresa un coste fijo elevado, por lo que se busca que las líneas programadas trabajen con el menor tiempo ocioso posible, de forma que se aproveche al máximo su uso. Esto se ve complementado con la reducción de líneas a utilizar, como se contempla en el modelo. Por otro lado, el

orden de magnitud de número de colores es parecido al número de líneas disponibles. Todo esto nos da pie a poder asumir que, si nuestro objetivo último es la reducción de tiempos de setup, no debería existir la necesidad de que un color sea procesado en varias líneas distintas. Sin embargo, dado que tanto las cantidades a procesar de cada color como las fechas de entrega de cada pedido son diferentes será en general necesario asignar el mismo color a varias líneas. Por este motivo, se habilita una opción para ir incrementando el número de líneas que puede utilizar un color en caso de que sea necesario.

## 4.2 Modelo Principal. Programación de la producción

Conociendo qué colores se van a procesar en según qué líneas, en este modelo se pasa a un segundo nivel en el cual se debe decidir en qué instante y máquina se procesa cada pedido, respetando las fechas de entrega de cada pedido. Sumado a esto, tenemos unas propuestas de paradas de mantenimiento que hay que integrar en la programación. A esta fase la denominamos “modelo principal” ya que es la que realmente detalla la programación de la producción que se persigue.

Para expresar el modelo principal de programación de la producción utilizaremos la notación propuesta por (R.L. Graham, E.L. Lawler, J.K. Lenstra, 1979). Esta notación permite clasificar de manera compacta los problemas según los objetivos, el entorno y las restricciones que lo definen. El formato sigue la estructura  $\alpha | \beta | \gamma$ . A continuación, se detallará a qué hace alusión cada campo, los principales valores en cada uno de ellos y cómo se traslada al caso práctico.

### 4.2.1 Entornos

Corresponde al símbolo  $\alpha$  en la notación mencionada. Los entornos describen la manera en la que las operaciones son secuenciadas en el conjunto de máquinas. Este punto es determinante y fundamental para conocer el problema ya que define la producción, como se procesan los trabajos. Nos podemos encontrar principalmente los siguientes casos:

- **Single Machine** ( $\alpha_1 = 1$  o  $\alpha_1 = \emptyset$ ). Entorno donde todas las operaciones deben ser procesadas en una sola máquina. Es el caso más simple y, por tanto, ha sido extensamente estudiado en la literatura.
- **Parallel Machine**. En este caso se cuenta con dos o más máquinas para procesar las operaciones. Se pueden ver tres variantes de este tipo de entornos:
  - o **Máquinas en paralelo idénticas** ( $\alpha_1 = P$ ). Los tiempos de proceso son iguales para todas las máquinas.
  - o **Máquinas en paralelo uniformes** ( $\alpha_1 = Q$ ). En esta variante, las máquinas poseen un factor de velocidad a la hora de procesar los trabajos que las diferencia. Como ya se ha mencionado, este es el caso que nos ocupa, pues cada línea tiene una tasa de salida propia en función a unos parámetros que veremos más adelante.
  - o **Máquinas en paralelo no relacionadas** ( $\alpha_1 = R$ ). También comentado anteriormente, los tiempos de proceso en este caso no siguen una relación en función de la velocidad de la máquina. Es el caso más cercano a la realidad.
- **Flow Shop** ( $\alpha_1 = F$ ). En este entorno, a diferencia del entorno anterior, los trabajos tienen que pasar por todas las máquinas. Estos deben seguir una ruta predefinida e igual para cada trabajo. De este modo, cuando un trabajo es procesado en una máquina, pasa a estar en espera de que la siguiente

máquina esté disponible para su proceso.

- **Job Shop** ( $\alpha_1 = J$ ). Similar al entorno anterior, pero en este caso cada trabajo tiene una ruta diferente. También llamado entorno tipo taller, ya que comparte el mismo fundamento, donde los trabajos pueden estar referidos a productos distintos y por ello requieran seguir unas fases de fabricación propias.
- **Open Shop** ( $\alpha_1 = O$ ). Entorno en el que cada trabajo se procesa en cada máquina, pero sin un orden establecido. Es el entorno más complejo, ya que abarca el caso más amplio en cuanto a conjunto de posibles soluciones.

En cada uno de estos tipos de entorno, a excepción del Single Machine, se especifica el número de máquinas que lo componen con un segundo campo:  $\alpha_2 = 1, 2 \dots, m$ .

Además de los casos comentados, es posible encontrar casos de distribuciones híbridas que combinan flujos de distintos entornos o segregan el entorno en estaciones, esto lo desarrollan (Vignier et al., 1999). Adicionalmente, también se pueden dar ciertas particularidades en las máquinas que afectan a la secuenciación de los trabajos, como es nuestro caso con las paradas de mantenimiento.

#### 4.2.2 Restricciones

Existen numerosas limitaciones que se pueden dar en la programación de la producción. En este punto se incluyen una serie de restricciones que han sido ampliamente estudiadas en la literatura. Estas restricciones se resumen en el campo  $\beta$ .

De partida, considerar un problema sin restricciones ( $\beta = \emptyset$ ) supone partir de unas suposiciones generales que se deben cumplir a nivel general para la mayoría de problemas de programación:

- Todos los trabajos y máquinas están disponibles para ser asignados desde el instante cero.
- El trabajo comenzado no puede ser interrumpido hasta su finalización.
- Cada máquina solo puede procesar una operación a la vez y cada operación de cada trabajo solo puede ser procesada en una máquina.
- No existe una limitación en la cantidad de operaciones que pueden estar esperando la disponibilidad de una máquina para ser procesada.
- Los tiempos de transporte entre máquinas son despreciables.

Estas condiciones base se pueden cumplir o no, cualquier modificación de estas o alguna restricción añadida queda reflejada en el campo. A continuación, se citan algunas de las restricciones más relevantes y si pertenecen al caso de estudio. Para ello, seguiremos la clasificación propuesta por (Framinan et al., 2014).

- **Restricciones en el proceso.** Son restricciones que afectan al flujo de los trabajos que se procesan.
  - o **Relaciones de precedencias** ( $\beta = prec$ ). Los trabajos se tienen que procesar en un orden establecido, normalmente asociado a la fabricación por ensamblado. Existen varios tipos de disposiciones: en serie ( $\beta = chain$ ), imposición de procesar varios trabajos para comenzar el siguiente ( $\beta =intree$ ), imposición de acabar un trabajo para comenzar los siguientes ( $\beta = outtree$ ), mezcla de los dos anteriores, donde múltiples relaciones de precedencia son posibles ( $\beta = tree$ ). No es aplicable en el caso de estudio.
  - o **Tiempos de cambio o setup.** Como se viene comentando son los tiempos de preparación o configuración de las máquinas para poder procesar los trabajos. Según el momento de ejecución se dan los anticipatorios, es decir, se puede adelantar la tarea de setup antes de

terminar el trabajo que lo precede (aplicable a entornos shop); o no anticipatorio, como es nuestro caso, cuyo setup comienza con la tarea de purga de los restos del trabajo que termina cuando el trabajo a procesar está preparado. Por otro lado, estos tiempos de setup pueden ser independientes de la secuencia ( $\beta = S_{nsd}$ ) o dependientes de la secuencia ( $\beta = S_{sd}$ ), como es nuestro caso.

- **Elegibilidad de las máquinas ( $\beta = M_i$ ).** Limitación por la que un trabajo  $i$  solo puede ser procesado en un subconjunto de máquinas. Esta restricción se ha forzado a través del primer modelo para simplificar el problema, haciendo que los pedidos solo sean procesados en máquinas según el color del material.
- **Secuencias de permutación ( $\beta = permu$ ).** Restricción para entornos Flow Shop. Si esta restricción se da, obliga a que cada máquina tenga que procesar los trabajos siguiendo la misma secuencia.
- **Disponibilidad de las máquinas ( $\beta = brkdwn$ ).** Este tipo de restricción es la que hace referencia a los posibles mantenimientos o averías que pueden sufrir las máquinas. Se pueden considerar intervalos de disponibilidad conocidos o aleatorios. En nuestro modelo, las paradas por mantenimiento no se entienden como disponibilidad de las máquinas, sino inversamente, como la no disponibilidad de ellas traducidas en forma de trabajos que entran en la programación en un cierto intervalo orientativo.
- **Reprocesado de trabajos ( $\beta = recrc$ ).** Se pueden entender dentro de este tipo trabajos que deben repetirse en ciertas máquinas por propio requerimiento del flujo productivo, o de forma estocástica, como pueden ser rechazos de producción por motivos de calidad del producto. En nuestro caso, se podría trasladar considerando los rechazos aleatorios de piezas concretas, no a nivel de pedidos completos. Esta interpretación se traduce en una modificación de la tasa de salida, es decir, tiempos de proceso aleatorios.
- **Trabajos omitidos ( $\beta = skip$ ).** Para trabajos que pudieran ser omitidos de la programación por algún motivo. No aplicaría en nuestra situación.
- **Máquinas no ociosas ( $\beta = no - idle$ ).** Para máquinas que, desde que comienzan a procesar su primer trabajo, no puede tener tiempos de parada, esto es debido al coste de arranque o que deben preservar unas ciertas condiciones durante todo el proceso. Es una restricción que nos aplica, ya que las líneas son complejas máquinas que producen las piezas en un flujo continuo, por lo que solo puede estar parada en caso de realizarse un cambio de producto o mantenimiento.
- **Procesado en lotes ( $\beta = batch$ ).** Se diferencian por una lotificación en paralelo ( $\beta = p - batch$ ), en el que una máquina es capaz de procesar varios trabajos en paralelo, siendo el tiempo de proceso total el del trabajo con mayor tiempo; y en serie ( $\beta = s - batch$ ), donde se procesan en serie y el tiempo total es la suma de todos los trabajos incluidos en el lote. La situación que modelamos no considera la restricción como tal, pero previamente se establece una unidad mínima productiva, que son los lotes de piezas entregables al cliente.
- **Restricciones en las operaciones.** Hacen referencias a las restricciones ligadas a las propias operaciones.
  - **Tiempos de proceso especiales.** Pueden darse los tiempos de proceso dependientes de la máquina, como nuestro caso ( $\beta = p_{im}$ ); independientes de las máquinas ( $\beta = p_{im} = p_i$ ); iguales para todos los trabajos ( $\beta = p_{im} = p$ ); unitarios ( $\beta = p_{im} = 1$ ); o binarios ( $\beta = p_{ij} = \{0,1\}$ ). Otra variante más cercana a la realidad sería la de considerar un factor de

aprendizaje o de deterioro de las máquinas, afectando al tiempo de proceso.

- **Trabajos interrumpibles** ( $\beta = pmtn$ ). Esta restricción permite la interrupción de los trabajos mientras está siendo procesados. Los tipos de interrupciones pueden ser: trabajos no recuperables tras la reanudación ( $\beta = pmtn - non - resumable$ ); trabajos recuperables parcialmente ( $\beta = pmtn - semi - resumable$ ); trabajos que conservan lo procesado tras la reanudación ( $\beta = pmtn - resumable$ ). Las interrupciones con productos no recuperables para procesar pedidos urgentes es una situación que se ha llegado a dar en la compañía. Para simplificar, no se han considerado interrupciones, pero se han realizado particiones de los pedidos que superan un determinado tamaño, en función de la unidad mínima productiva (que es el palet de piezas completo).
- **Instantes de inicio y entrega o finalización de trabajos.** En este punto se incluyen los instantes de inicio o lanzamiento de un trabajo superiores al inicio de la programación ( $\beta = r_i$ ); las ya ampliamente introducidas fechas de entrega,  $\beta = d_i$  si se permite sobrepasarlas;  $\beta = \bar{d}_i$  si son fechas tope. El producto que fabrica la compañía es de muy poco valor con respecto a la paralización que pueden causar los retrasos en los pedidos, por lo que se han considerado fechas tope, aunque con la interpretación de fechas de entrega como explicaremos más adelante.
- **Trabajos sin esperas** ( $\beta = nwt$ ). Restricción para entornos shop. Los trabajos que se deben secuenciar en varias máquinas requieren que se procesen de manera continua.
- **Restricciones de transporte.** Es posible que debido a los medios físicos los trabajos necesiten un tiempo adicional para ser procesados en distintas máquinas. Haciendo alusión al caso de estudio, si se hubieran interpretado las líneas como entornos shop, las piezas dentro de las líneas son transportadas para las distintas operaciones que se realizan, pero el tiempo dedicado es despreciable, de modo que las líneas se tratan como cajas negras a ojos del modelo.
- **Restricciones de almacenaje.** Otro punto importante a valorar son los inventarios intermedios o buffers que puedan existir durante la producción. Al igual que en la restricción anterior, en el interior del sistema de la línea existe un buffer para mitigar las bajadas en la producción de la línea y mantener una tasa de salida constante, pero se ha omitido al tratarlos como máquinas paralelas. Esta restricción ha sido estudiada comúnmente de la siguiente forma:
  - $\beta = buffer = b$ . Establece una capacidad de trabajos que pueden estar en cola esperando la liberación de la máquina.
  - $\beta = buffer = b_j$ . Cada máquina dispone de su propia capacidad para almacenar trabajos en espera.
  - $\beta = block$ . Bloquea la máquina cuando el trabajo se finaliza.
- **Otras restricciones.** Recogen situaciones que no han sido contempladas anteriormente, como por ejemplo la limitación de los recursos, ya comentada anteriormente.

### 4.2.3 Objetivos

Hace alusión al campo  $\gamma$ . Los objetivos para la programación de la producción se pueden clasificar según unos criterios de coste (coste de entregas retrasadas, tiempos ociosos, setups, ...), tiempo (makespan, retrasos, utilización de máquinas, tiempos de espera, setups, ...), calidad (nivel de servicio, propiedades del producto, aspectos sociales, ...) o flexibilidad. Es importante identificar la finalidad y prioridad que se busca al establecer los objetivos para el modelo, ya que existen objetivos que entran en conflicto o que pueden

perjudicar la programación para otros criterios o puntos de vista. La respuesta obtenida de los objetivos elegidos y sus repercusiones sobre la programación también vendrá relacionada con la naturaleza del sistema de producción en el que se trabaja.

Previamente a indicar los objetivos más relevantes, se pueden señalar una serie de medidas características que sirven para poder evaluar la bondad de la programación obtenida. Se destacan:

- $C_i$ : Tiempo de terminación del procesado del trabajo  $i$ .
- $F_i$ : Tiempo de flujo del trabajo  $i$ . Es el tiempo que el trabajo está presente en la programación. Expresado como  $F_i = C_i - r_i$ . Donde  $r_i$  representa el instante de inicio.
- $L_i$ : Retraso del trabajo  $i$ . Mide la desviación sobre las fechas de entrega. Puede tomar valores positivos o negativos. Expresados como  $L_i = C_i - d_i$ .
- $T_i$ : Tardanza del trabajo  $i$ . Se centra exclusivamente en los trabajos que sobrepasan las fechas de entrega. Expresados como  $T_i = \max\{0, L_i\}$ .
- $E_i$ : Adelanto del trabajo  $i$ . Se centra exclusivamente en los trabajos que han sido finalizados antes de su fecha de entrega. Expresados como  $E_i = \max\{0, -L_i\}$ .
- $U_i$ : Número de trabajos tardíos. Expresados como  $U_i = 1$  si  $T_i > 0$ ,  $U_i = 0$  en otro caso.

Tras revisar estos conceptos, se pueden establecer varios objetivos. Muchos de ellos ya mencionados anteriormente, seguiremos la clasificación dada por (Framinan et al., 2014) estructurándolos en las siguientes categorías:

- **Factibilidad** ( $\gamma = -$ ). La única aspiración es encontrar una programación válida que satisfaga las condiciones impuestas, sin establecer una orientación o preferencia concreta.
- **Objetivos no relacionados con fechas de entrega**. Objetivos en los que las fechas de entrega de los trabajos no es un criterio de búsqueda de la mejor solución.
  - **Minimización del Makespan** ( $\gamma = C_{max}$ ). Tiempo de terminación del último trabajo a procesar en la última máquina que visita. Largamente comentado es uno de los objetivos más estudiados en la literatura, aunque no es de aplicación en nuestro modelo.
  - **Minimización del tiempo de flujo máximo** ( $\gamma = F_{max}$ ). Similar al caso anterior, se centra en los trabajos que más tiempo permanecen en la programación. Si los trabajos tienen instantes de lanzamiento o inicio igual a cero ( $r_i = 0$ ), el objetivo es el mismo que el makespan.
  - **Minimización de la suma de tiempo total de terminación** ( $\gamma = \sum C_i$ ). Es uno de los objetivos que tendremos en cuenta, como se detallará en el siguiente apartado. De igual forma, se puede presentar su equivalente para los tiempos de flujo.
  - **Minimización del tiempo medio de terminación** ( $\gamma = \bar{C}$ ). Valora un promedio de los tiempos de terminación de cada trabajo. De igual forma, se puede presentar su equivalente para los tiempos de flujo.
- **Objetivos relacionados con fechas de entrega**. Involucra las fechas de entrega en los objetivos. Es común su uso como indicador del nivel de servicio o para productos con alta criticidad.
  - **Minimización el retraso máximo** ( $\gamma = L_{max}$ ). Puede tomar valores negativos por lo que tenderá a que se produzcan más adelantos en los trabajos.
  - **Minimización de la tardanza máxima** ( $\gamma = T_{max}$ ). Busca minimizar el tiempo del trabajo más tardío.

- **Minimización del adelanto máximo** ( $\gamma = E_{max}$ ). Busca minimizar el tiempo del trabajo más adelantado.
  - **Minimización de la suma o promedio de retrasos** ( $\gamma = \sum L_i / \gamma = \bar{L}$ ). Reduce el tiempo total o promedio de trabajos tardíos.
  - **Minimización de la suma o promedio de tardanzas** ( $\gamma = \sum T_i / \gamma = \bar{T}$ ). Reduce el tiempo total o promedio de trabajos tardíos.
  - **Minimización de la suma o promedio de adelantos** ( $\gamma = \sum E_i / \gamma = \bar{E}$ ). Reduce el tiempo total o el promedio de trabajos adelantados.
  - **Minimización de la suma de trabajos tardíos** ( $\gamma = \sum U_i$ ) **o adelantados** ( $\gamma = \sum V_i$ ). Este caso se enfoca en el número de trabajos que sobrepasan o se adelantan a las fechas de entrega.
- **Objetivos sobre las reprogramaciones.** En ellos se comparan dos secuencias y se evalúa la disonancia entre ellas en términos de orden o tiempos. Este tipo de objetivos no han sido tenidos en cuenta para el trabajo que nos ocupa.
  - **Objetivos adicionales.** Cuando se trabaja con modelos más complejos, con restricciones o condiciones especiales, entran en juego otros objetivos posibles para resolver el problema. Uno de ellos, el cual nos ocupa, son los tiempos de setup, medidos en términos de coste de realización de la tarea. Esto se extiende a la búsqueda de minimizar los costes de producción, de utilización de recursos, etc.

En los objetivos descritos se valoran todos los trabajos por igual, pero es probable que no todos los trabajos tengan la misma importancia, por ejemplo, si representan productos prioritarios para la compañía. Es por esto, que estos objetivos se pueden adaptar con unos pesos de forma que se pondere cada trabajo por su importancia relativa. En nuestro entorno no se tiene en cuenta.

Es frecuente que la solución buscada este orientada a optimizar distintos aspectos. La manera en la que reaccionan los modelos ante la aplicación de distintas medidas evaluadas simultáneamente puede dar lugar a soluciones que no satisfagan nuestras expectativas. Por esto, es conveniente saber tratar los conflictos que puedan surgir en los problemas multiobjetivo.

En primera instancia, hay que clarificar cuales son los objetivos reales que precisa el cliente que se va a servir, ya que normalmente surgirá una discusión de prioridades. La compañía que nos solicita una solución para este problema de programación claramente indicó la necesidad de cubrir la carencia existente en los tiempos invertidos en setups. Este objetivo por sí solo, deja de lado el resto de objetivos con los que se puede medir la programación que, aunque no sean prioritarios, conviene tenerlos en cuenta para diseñar un modelo robusto. Por ejemplo, minimizar solo el tiempo de setups es un criterio miope a los tiempos de finalización de los pedidos, permitiendo tiempos ociosos.

Un ejemplo de este tipo de conflictos se observa en el estado del arte, donde el objetivo de minimizar la suma de tiempos de terminación no está alineado e incluso puede entrar en conflicto con el makespan. Esto es debido a que minimizar la suma de tiempos buscará asignar el mayor número de trabajos lo antes posible, desplazando todos los trabajos lo más al inicio de la programación posible, mientras que el makespan buscará acabar la programación del conjunto lo antes posible, mediante un reparto de los trabajos en las máquinas, con un cierto equilibrado, sin tener en consideración que esto pueda perjudicar en la finalización de ciertos trabajos. Otro ejemplo podemos verlo al minimizar los trabajos tardíos y adelantados, un modelo con este planteamiento multiobjetivo con objetivos contrapuestos busca una programación orientada a la filosofía Just In Time (JIT). En otros casos, para minimizar la desviación absoluta se utiliza la expresión  $\gamma = E_{max} + T_{max} = ET_i$ .

Por tanto, en nuestro modelo, se minimizarán los tiempos de setup y, para no sacrificar los tiempos de

terminación de los pedidos de forma que sea una programación ágil, se minimizará también la suma de estos. La evaluación del cumplimiento de las fechas de entrega se ha marcado como no negociables, por lo que directamente se imponen cero tardanzas a través de restricciones en el modelo.

#### 4.2.4 Formulación matemática del modelo principal

Tras describir la estructura de la notación que resume un modelo de programación, podemos presentar el problema objeto de este TFM como:  $Q12 \mid p_{im}, d_i, S_{sd} = s_{ij}, brkdwn \mid w \sum \sum TS_{ij} + \sum C_i$ , (considerando un solo producto).

Anteriormente se mencionó para el modelo preliminar como el color afecta al tiempo de setup o preparación de la línea. En más detalle, el proceso de preparación de la línea que se produce cuando se requiere un cambio de color pasa por una parada completa de la línea, seguidamente se retiran los excedentes de productos intermedios que quedarán repartidos por la línea. Posteriormente, se realiza una limpieza en profundidad de todo el polvo y restos de colorante del circuito y se purga el usillo, cavidades y los conductos por los que haya pasado el producto. Tras esto, la línea se pone en marcha en una fase de arranque que dura hasta que se elimina el resto de impurezas y se alcanza el color preciso.

La duración de este proceso varía significativamente según la diferencia en los tonos de los colores. Por esto, se establece una clasificación de los colores según si es de tono claro o de tono oscuro. Por tanto, los tiempos de setup son dependientes no tanto de su color como de su tono.

Para cada línea, se generará un pedido propio que marca el origen de la secuencia en la línea. Este pedido corresponde a los pedidos en curso que procesa cada línea en el momento de la replanificación, como ya se había comentado en el modelo preliminar, con un tiempo de proceso igual al tiempo de proceso. Si la línea se encuentra parada en el momento de la programación se indica un pedido de inicio de duración cero, normalmente llamado “dummy”. Estos pedidos pasan a ser un subconjunto del total de pedidos de la instancia.

Para las paradas de mantenimiento preventivo, se ha optado por modelarlas de la misma forma, como pedidos especiales dentro del conjunto de pedidos, con unos instantes de inicio y fin sugeridos, pero ligeramente modificables según las condiciones de la programación. El hecho de que las paradas deban procesarse en una ventana de tiempo, podría haber llevado a considerar un modelo con instantes de disponibilidad y vencimiento  $[r_i, d_i]$  como un intervalo de tiempo continuo restringido para el procesamiento (Chuzhoy & Codenotti, 2009). Sin embargo, esto obliga al usuario a decidir o calcular la amplitud del intervalo para cada parada de mantenimiento, sin especificar el momento idealmente planificado. Por este motivo, se ha optado por crear unas variables de tolerancia sobre los instantes de inicio y fin del mantenimiento planificado e imponer la ejecución en dicho tramo, de forma que el propio modelo decida cuanto se retrasa o adelanta el mantenimiento, pero sin que la tolerancia establecida supere un tiempo de desfase total sobre lo planificado.

Explicados los detalles introductorios del modelo, pasamos a concretar el modelo matemático. Los datos de entrada son:

- Un conjunto de pedidos  $P = \{1, \dots, i, \dots, |P|\}$  para ser asignados a las líneas.
- Un pedido de origen  $o$  que representa el inicio de la secuencia para cada línea.
- Dentro del conjunto  $P$ , un subconjunto de  $|E|$  pedidos con los pedidos iniciales o en curso,  $E = \{1, \dots, e, \dots, |E|\} \subseteq P$ .
- Dentro del conjunto  $P$ , un subconjunto de  $M = \{1, \dots, m, \dots, |M|\} \subseteq P$  pedidos con los pedidos que

hacen referencia a las paradas para mantenimientos preventivos. Se numeran de manera que exista una correspondencia tal que el número de pedido es igual a la línea a la pertenece el mantenimiento,  $m = l$ .

- Un conjunto de  $L = \{1, \dots, l, \dots, |L|\}$  líneas.
- Cada pedido  $i$  tiene un subconjunto de líneas  $PL_i = \{1, \dots, l, \dots, |PL_i|\}$  a las que se puede asignar como consecuencia de la aplicación del modelo preliminar, a excepción del pedido origen  $o$ .
- $w$  es el peso que se le asigna al sumatorio de tiempos de setup en la función objetivo para ponderar su importancia. La suma de los tiempos de setup será mucho menor que la de los tiempos de finalización de los pedidos. Para alcanzar un orden de magnitud comparable entre ambas medidas, se calcula el valor como  $w = \frac{lp \cdot \sum ct_i}{ts}$ , donde  $lp$  es el número de líneas en funcionamiento, obtenido en el modelo preliminar;  $\sum ct_i$  es la suma de las cantidades de todos los colores, es decir, el total de unidades a procesar y  $\overline{ts}$  es la tasa de salida promedio de las líneas en funcionamiento.
- $s_{ij}$  es el tiempo de setup en que se incurre al procesar el pedido  $i$  justo antes de  $j$ .
- $spm_m$  es el instante de inicio propuesto para el pedido de mantenimiento correspondiente a la línea  $m$ .
- $fpm_m$  es el instante de finalización propuesto para el pedido de mantenimiento correspondiente a la línea  $m$ .
- $tpm$  es el tiempo total de desviación admisible para la ejecución del mantenimiento preventivo frente a la propuesta dada.
- $p_{il}$  es el tiempo de proceso del pedido  $i$  en la línea  $l$ .
- $cte$  es una constante de valor alto.
- $d_i$  es la fecha de entrega para el pedido  $i$ .
- $tm$  es la tardanza máxima que se permite a los pedidos. En primera instancia, toma el valor  $tm = 0$  para forzar el cumplimiento de todas las fechas comprometidas.

En cuanto a las variables de decisión, se definen para el modelo:

- $X_{ij} = 1$  si el pedido  $i$  es procesado inmediatamente antes al pedido  $j$ . En caso contrario toma el valor cero.
- $Y_{il} = 1$  si el pedido  $i$  es asignado a la línea  $l$ . En caso contrario toma el valor cero.
- $S_i \geq 0$  es el instante de comienzo del pedido  $i$ .
- $F_i \geq 0$  es el instante de finalización del pedido  $i$ . Para lo visto anteriormente, corresponde a los tiempos de terminación  $C_i$ .
- $WS_m \geq 0$  es la variable que actúa como origen de la ventana de tiempo del mantenimiento preventivo  $m$ .

- $WF_m \geq 0$  es la variable que actúa como fin de la ventana de tiempo del mantenimiento preventivo  $m$ .
- $T_i \geq 0$  es la tardanza sobre la fecha de entrega de cada pedido  $i$ .

Con esto, el modelo se formula como:

$$Z_2 = \text{Min} \sum_{\substack{i \in P \\ i \neq o}} F_i + w \cdot \sum_{\substack{i \in P \\ i \neq o}} \sum_{\substack{j \in P \\ PL_i \cap PL_j \\ i \neq j \\ j \neq o}} s_{ij} \cdot X_{ij} \quad (1.15)$$

s.a.:

$$X_{oe} = 1, \quad e \in E \subseteq P \quad (1.16)$$

$$Y_{il} = 1, \quad i \in P, \quad l \in PL_i, \quad |PL_i| = 1 \quad (1.17)$$

$$Y_{il} = 0, \quad i \in P, \quad l \notin PL_i \quad (1.18)$$

$$S_e = 0, \quad e \in E \subseteq P \quad (1.19)$$

$$S_m \geq spm_m - WS_m, \quad m \in M \subseteq P \quad (1.20)$$

$$F_m \leq fpm_m + WF_m, \quad m \in M \subseteq P \quad (1.21)$$

$$WS_m + WF_m \leq tpm, \quad m \in M \subseteq P \quad (1.22)$$

$$\sum_{i \in P} \sum_{\substack{j \in P \\ PL_i \cap PL_j \\ i \neq j \\ j \neq o}} X_{ij} = |P| \quad (1.23)$$

$$\sum_{l \in L} Y_{il} = 1, \quad i \in P \quad (1.24)$$

$$\sum_{i \in P} X_{ij} \leq 1, \quad j \in P \quad (1.25)$$

$$\sum_{j \in P} X_{ij} \leq 1, \quad i \in P \quad (1.26)$$

$$Y_{il} + \sum_{l' \in PL_i \cap PL_j, l' \neq l} Y_{jl'} + X_{ij} \leq 2, \quad l \in PL_i \cap PL_j, \quad i, j \in P, \quad i, j \neq o, \quad i \neq j \quad (1.27)$$

$$F_i - S_i - \sum_{l \in LP_i} p_{il} \cdot Y_{il} \geq 0, \quad i \in P \quad (1.28)$$

$$S_j - F_i + cte \cdot (1 - X_{ij}) - s_{ij} \cdot X_{ij} \geq 0, \quad i, j \in P, \quad PL_i \cap PL_j, \quad i, j \neq o, \quad i \neq j \quad (1.29)$$

$$F_i - T_i \leq d_i, \quad i \in P \quad (1.30)$$

$$T_i \leq tm, \quad i \in P \quad (1.31)$$

$$X_{ij} \in \{0, 1\}, \quad Y_{il} \in \{0, 1\} \quad (1.32)$$

$$S_i \geq 0, \quad F_i \geq 0, \quad WS_m \geq 0, \quad WF_m \geq 0, \quad T_i \geq 0 \quad (1.33)$$

La función objetivo en (1.15) consiste en una combinación convexa de dos medidas para darle un enfoque multiobjetivo. Como ya vimos, reducir los tiempos de setup como única dinámica para el modelo llevaría a obviar el conjunto de la programación, de ahí la minimización de los tiempos de finalización, lo que agrega importancia a finalizar los pedidos lo antes posible. Cabe mencionar como este objetivo aplicado a las

máquinas paralelas uniformes adoptará un comportamiento igual a la regla de despacho SPT-FM, ya mencionada en el estado del arte, que sigue un procesamiento en escalera desde las máquinas más rápidas (con más trabajos asignados) a las más lentas (menos trabajos asignados).

La restricción (1.16) inicializa las secuencias en las líneas con el pedido de inicio o el pedido en curso. En (1.17) se asignan a las líneas los pedidos que tienen una sola línea permitida en su conjunto de líneas asignables  $PL_i$ , por eso es necesario que se cumpla  $|PL_i| = 1$ . Entre estos pedidos están los de inicio o en curso, los pedidos de mantenimiento preventivo y los pedidos resultantes del primer modelo, para los colores que se procesen en una sola línea. Consecuentemente, en (1.18) se impone que los pedidos no se asignen a las líneas que no están en sus propios conjuntos de líneas admisibles. Volviendo a los pedidos iniciales, en la restricción (1.19) se fuerza que comiencen en el instante cero, es decir, al inicio de la programación.

Con (1.20) y (1.21) se modelan las ventanas temporales de los comienzos y fin de las paradas de mantenimiento, respectivamente. Como ya se comentó, se agregan unas variables de holgura que permiten desplazar la actividad en el tiempo, haciendo que la programación sea más flexible y evitando tiempos ociosos previo al inicio del mantenimiento como consecuencia de no encontrar ningún pedido que encaje en ese tiempo. Esta holgura es delimitada en (1.22), de forma que la suma de ambas holguras tiene que ser inferior a una tolerancia máxima permitida.

Las siguientes restricciones son más parecidas a las que usualmente forman parte de los modelos de programación de máquinas paralelas. En (1.23) se fuerza que todos los pedidos sean programados (con  $|P|$  indica el total de pedidos a programar). Se muestra como las relaciones entre pedidos solo puede ser viables para los pedidos cuyas líneas permitidas de asignación intersecten,  $PL_i \cap PL_j \subseteq L$ . Con la restricción (1.24) se indica que un pedido puede ser procesado en una sola línea. Las restricciones (1.25) y (1.26) garantizan que la relación de precedencia entre un pedido  $i$  y  $j$  solo pueda darse, como máximo, una vez. De esta forma se evita el solape de pedidos.

La restricción (1.27) sirve de conexión entre las variables binarias  $X_{ij}$  e  $Y_{il}$ . Con esta restricción se consigue que si se activa la relación de precedencia entre dos pedidos ( $X_{ij} = 1$ ), forzosamente estos pedidos deben pertenecer a la misma línea, o bien que ninguno de ellos pertenezca a la misma línea que se evalúa. Podemos ver como esto ocurre ya que si el pedido  $i$  está asignado en la línea  $l$ , y el pedido  $j$  está asignado a una línea distinta de  $l$ , entonces el termino dependiente vale 3, superando el límite de 2. Si  $X_{ij} = 0$ , la restricción es redundante ya que no existe limitación y los pedidos pueden asignarse a cualquier línea. Una alternativa un tanto menos eficiente para plantear esta condición es con el uso de las siguientes dos restricciones disyuntivas:

$$Y_{il} + (1 - Y_{jl}) \leq 1 + (1 - X_{ij})$$

$$(1 - Y_{il}) + Y_{jl} \leq 1 + (1 - X_{ij})$$

Siguiendo el modelo, la restricción (1.28) realiza el cálculo de la duración de los pedidos en base a los instantes de comienzo y finalización. Por otro lado, en (1.29) relaciona pedidos consecutivos en la programación, teniendo en cuenta los tiempos de setup. De esta forma, el inicio del pedido  $j$  se dará en el momento en el que finalice el pedido  $i$  sumado a la actividad de setup entre ellos. Con la constante  $cte$  los pedidos se liberan de esta restricción si no existe una precedencia estricta entre ellos.

Las últimas restricciones abordan la tardanza sobre la fecha de entrega. Con (1.30) se obtienen las tardanzas  $T_i$ , siendo positivas sólo cuando se sobrepasan las fechas de entrega. Finalmente, la restricción (1.31) pone límite a la tardanza permitida para cada pedido.

En (1.32) y (1.33) se definen las variables como binarias y continuas, respectivamente.

Si en el modelo previo se generaron grupos de pedidos y líneas independientes, este modelo se ejecutará tantas veces como grupos haya, pero de forma mucho más veloz ya que se trabajaría con instancias considerablemente menores.

La resolución de este modelo proporciona una programación de la producción con los instantes de inicio y fin de cada pedido, considerando la producción en curso y detallando las paradas de mantenimiento. Todo esto con el objetivo de minimización de los tiempos de setup, por lo que tenderá a la especialización de las líneas en ciertos productos si las fechas de entrega lo permiten. Adicionalmente, este objetivo se acompaña con un intento por ajustar la programación para reducir los tiempos de finalización, lo que permitirá un margen de comodidad para la entrega de los pedidos al cliente.

## 5 RESOLUCIÓN DEL PROBLEMA

---

Una vez presentado el método de resolución, nos disponemos a exponer como se ha llevado a la práctica y qué resultados ofrece la solución propuesta. Para ello, comenzaremos exponiendo los medios y herramientas empleados en el desarrollo, así como la obtención de la información necesaria para la ejecución de los modelos. Seguidamente se explicará de forma breve el desarrollo y la implementación realizada. Por último, se expondrán y analizarán los resultados obtenidos en comparación con el procedimiento habitual utilizado en la empresa.

### 5.1. Estructuración y obtención de los datos iniciales

La información sobre recepción de pedidos, disponibilidad de las líneas y sus estados y planificación de los mantenimientos, procede de distintos orígenes y es tratada por distintos departamentos en la organización.

Para una aplicación que sea práctica, todos estos datos se canalizan en un repositorio que funciona como fuente de origen para la herramienta diseñada. Se ha decidido que esta sea un libro Excel, preparado para recoger y organizar todos los datos de forma estandarizada, facilitando el proceso de recopilación de datos y lectura de los mismos en cada ejecución.

#### 5.1.1 Datos de los pedidos entrantes

La compañía con la que se colabora es una de las más fuertes de su sector y dispone de unos potentes sistemas de información, gracias a los cuales ha sido posible la obtención de datos de gran calidad y fiabilidad.

La compañía utiliza SAP como MRP y sistema de programación de la producción, pero este no puede resolver el problema de forma eficaz al no ser capaz de integrar todas las características internas y externas al proceso productivo ni adaptarse a las incidencias o urgencias que puedan surgir, de ahí la necesidad de una nueva metodología que permita enfrentar esta problemática y complete una programación factible.

Los pedidos de los clientes y las ordenes de reaprovisionamiento que elabora el planificador determinan las ordenes de fabricación definitivas. Éstas se desglosan en una tabla de Excel que leerá la herramienta que ejecutará los modelos. El listado contiene los siguientes campos:

- **Producto.** Se diferencia si el pedido es para un producto base de tipo A o B. Según el caso será programado para el conjunto de líneas destinadas a tal producto.

- **Color.** Indica el color de la pieza a partir de un catálogo de colores estandarizado. Este catálogo diferencia los principales colores existentes a fin de determinar los tiempos de setup. El listado de colores tenidos en cuenta se muestra en la siguiente tabla:

*Tabla 5.1 Listado de colores y tonalidades*

Color	Tonalidad	
White	Claro	
Blue	Oscuro	
Yellow	Claro	
Red	Oscuro	
Black	Oscuro	
Gray	Claro	
Green	Oscuro	
Orange	Oscuro	
Mocha	Oscuro	
Pink	Oscuro	
Gold	Oscuro	

Como se puede observar, en la misma tabla se indica la tonalidad a la que pertenece cada color. Esta tonalidad es la magnitud que realmente determina el tiempo de setup. Esta tonalidad permite expresar la magnitud de los tiempos de setup de acuerdo a la transición entre distintas tonalidades. Concretamente, a modo de ejemplo, estos tiempos se especifican en la siguiente tabla:

*Tabla 5.2 Ejemplo de tiempos de setup*

Tiempos de Setup (min)	Ref. A10000036	Ref. A100000131	Ref. A100000500	Ref. A100000874
Desde \ Hasta	– White	– Blue	– Yellow	– Black
Ref. A10000036 – White	0	60	10	60
Ref. A100000131 – Blue	240	0	240	10
Ref. A100000500 – Yellow	10	60	0	60
Ref. A100000874 – Black	240	10	240	0

Como vemos, si el color a sustituir es de tono claro (Ref. A100000036) y se da paso a uno de tono oscuro (Ref. A100000131), se incurre en un tiempo de setup de 60 minutos. Inversamente, si la referencia es de tono oscuro y la siguiente a ser procesada tiene un tono claro, la dificultad de erradicar los rastros de un color oscuro provocan que el tiempo estándar de la operación de limpieza sea de 240 minutos. Existen referencias que son distintas con colores muy parecidos, para simplificar el listado de colores se tratarán como si fueran del mismo color. La duración del setup entre ellas se reduce a 10 minutos. Las paradas de mantenimiento que han sido modeladas como un pedido más a procesar, se han tratado como colores de tonalidad clara, ya que, con el mantenimiento, se realiza una limpieza de las líneas y por tanto el tiempo de arranque se puede aproximar al de una referencia de esta tonalidad.

- **Pedido/Item.** Es el código generado por el sistema que identifica el pedido concreto a programar.
- **Referencia.** Detalla, mediante un código unívoco, la referencia concreta de la pieza del pedido. Esta referencia es una configuración concreta de color y grabados que definen la pieza de forma unívoca.
- **Descripción.** Breve descripción técnica de la referencia a procesar.
- **Tipo de pedido.** En este campo se indica si es un pedido extraordinario o si se trata de un pedido que proviene de un reaprovisionamiento de almacén para una referencia en stock.
- **Fecha de creación.** Indica la fecha de creación del pedido en el sistema.
- **Fecha de entrega.** Indica la fecha planificada y comprometida para la entrega al cliente.
- **Cantidad requerida.** Volumen de producto solicitado. Es la cantidad de piezas total a despachar expresada en miles de unidades. Cada requerimiento debe ser múltiplo de la unidad mínima productiva, *ump*. Cada producto A y B, tienen una unidad mínima productiva propia, que corresponde al número de piezas que incluye el lote de despacho mínimo, es decir, un pallet de cajas. Para cada producto se tiene:

$$ump_A = n \cdot uc_A = 24 \cdot 4375 = 105000 \text{ piezas}$$

$$ump_B = n \cdot uc_B = 24 \cdot 3000 = 72000 \text{ piezas}$$

Donde *n* es el número de cajas por pallet y *uc* son las piezas que contiene una caja. Por ejemplo, si la cantidad requerida de un pedido *i* del producto A es de 315, la solicitud corresponde a

$$\frac{CantReq_i \cdot 1000}{ump_A} = \frac{315 \cdot 1000}{105000} = 3 \text{ pallets} .$$

La tabla de pedidos se debe alimentar con los datos de los pedidos pendientes de programar cada vez que se vaya a ejecutar una reprogramación. Los pedidos en curso y de mantenimiento preventivo se explican en el siguiente apartado.

### 5.1.2 Datos de la configuración del entorno productivo

Junto con los pedidos pendientes de programar se necesita precisar la información sobre el estado de las líneas en el momento de la ejecución y otros requerimientos para los modelos. Esta información se detalla en otra hoja del mismo documento Excel.

Los datos incluidos en esta hoja Excel se han desglosado según los productos A y B. Estos se encuentran a su vez separados en los siguientes bloques:

- **Información general de los pedidos.** Detalla información adicional que sirve para la definición de

las características de los pedidos, más allá del listado de pedidos pendientes de programar. Se dispone de:

- Catálogo de colores y cantidad total a procesar de cada uno de ellos. Se utiliza para el modelo preliminar. La cantidad de cada color se obtiene automáticamente mediante una fórmula de conteo de Excel sobre la hoja de pedidos pendientes. La lectura de estos se produce siempre que la cantidad sea mayor que cero, de forma que solo se tengan en cuenta los colores que apliquen en cada instancia.
- Matriz de tiempos de setup. Relación de tiempos de setup para las transiciones: misma tonalidad para distintas referencias, tonalidad clara a oscura y tonalidad oscura a clara.
- Fraccionado de pedidos. Con el fin de aportar un grado más de libertad al modelo ante mantenimientos preventivos en intervalos en los que no sea posible encajar un pedido o por posibles congestiones en las fechas de entrega, se realizan particiones artificiales de los pedidos cuando la cantidad solicitada en un mismo pedido es demasiado elevada. Sin estas particiones, pueden recibirse pedidos de enorme tamaño que den lugar a problemas infactibles. El parámetro que establece dicho límite se calcula mediante la fórmula:

$$PedidoMax = p \cdot \frac{cap}{h}$$

Donde  $p$  es el número de días máximo que se permite procesar un mismo pedido, con un valor arbitrario a decidir por el usuario (las pruebas nos indican que un valor de 3 días da buenos resultados).  $cap$  es la capacidad promedio de fabricación, ya comentada en el capítulo anterior, calculada también en esta hoja. El parámetro  $h$  es el horizonte de programación establecido, en este caso 15 días. El cociente  $\frac{cap}{h}$  recoge la capacidad diaria promedio de fabricación del sistema. El valor de  $PedidoMax$  debe ser múltiplo de la unidad mínima productiva,  $ump$ , comentada anteriormente, para no generar particiones en cantidades que no se ajusten a los lotes por pallet predefinidos. Si no lo fuera, se trunca al múltiplo inferior.

- **Información sobre las líneas.** En este punto se detallan los factores que influyen en la velocidad de cada línea y los pedidos en curso o paradas de mantenimiento planificadas en cada una de ellas.
  - Tasas de salida reales. La tasa de salida real indica la velocidad de cada línea. Esta tasa, junto con la cantidad de pedido es la que da lugar al tiempo de proceso,  $p_{il}$ . El cálculo del tiempo de proceso se podría expresar en términos generales de la siguiente forma:

$$p_{il} = f(CantReq_i, tsn, P_i^R, P_i^C)$$

Donde  $CantReq_i$  es la cantidad requerida de cada pedido, la cual debe multiplicarse por 1000 para obtener el número de piezas total.

El término  $tsn$  indica la tasa de salida nominal o ideal de la línea, medida en unidades por minuto. Se calcula como  $tsn = \frac{Cav}{TC}$ , en el que  $Cav$  es el número de cavidades del molde,

con un total de 48 cavidades teóricas para los dos tipos de líneas (productos A y B), y  $TC$  es el tiempo de ciclo de moldeado, con un valor de 3,8 segundos la unidad en las líneas del producto A y 4,5 en las de B. Cabe reseñar que efectivamente es el proceso de moldeado el

que marca el takt time de la línea, debido a que los procesos posteriores de ensamblaje y grabado cuentan con un buffer intermedio y distintas velocidades para ajustarse a este.

Sin embargo, en la realidad existen pérdidas durante la fabricación por diversos motivos. Estas pérdidas están controladas y monitorizadas por la organización y se pueden clasificar como pérdidas en el rendimiento y pérdidas de calidad ( $P_i^R$  y  $P_i^C$ ). Las primeras son las pérdidas en el rendimiento y se calculan como el cociente entre las cavidades disponibles y las teóricas. Normalmente, algunas de las líneas tienen cavidades tapadas por motivos de calidad o reparaciones.  $P_i^C$  son las pérdidas de calidad, en estas se valoran las piezas defectuosas que se rechazan tras los controles de calidad.

En la hoja de cálculo vendrá indicada la tasa de salida real, que es el producto de la tasa de salida nominal y las pérdidas,  $ts_i = tsn \cdot P_i^R \cdot P_i^C$ . Si durante el momento de la ejecución la línea está completamente fuera de servicio, basta con indicar  $ts_i = 0$  para que se interprete como que una línea inoperativa.

- Pedidos en curso. En cada línea se deben incluir, si aplica, los pedidos que están en proceso en el momento de la ejecución del script. Se debe incluir la información mínima necesaria: la referencia del producto, el color y la cantidad restante por procesar.
- Paradas de mantenimiento preventivo. Igualmente, para las líneas en las que se tenga planificado un mantenimiento, se debe introducir el instante de inicio según la planificación y el de finalización.
- **Otros parámetros configurables.** Son otros parámetros que debe establecer el usuario para calibrar el modelo. Estos son:
  - Número máximo de líneas que pueden procesar un mismo color ( $lcm$ ). Como se comentó en el modelo preliminar, corresponde al número de líneas a las que es posible asignar un determinado color.
  - Amplitud de ventana para la programación de las paradas de mantenimiento ( $tpm$ ). Se puede introducir una por cada línea. Las paradas de mantenimiento preventivo están planificadas una vez al mes por cada línea, por lo que no tiene sentido considerar más de una.

Otros parámetros de entrada como son las constantes usadas en los modelos de optimización están declarados en el propio script que pasaremos a comentar a continuación.

## 5.2. Desarrollo e implementación

La herramienta para la resolución del problema es un script codificado en Python 3.6 y desarrollado mediante el editor de texto Visual Studio Code. Python es un lenguaje de programación en auge desde hace años y que continúa creciendo debido a su sencillez y versatilidad. Destaca por la gran cantidad de librerías, en especial las dedicadas al uso estadístico y aprendizaje automático.

Junto a Python, se utiliza Gurobi 9.0 como solver para la ejecución de los modelos de optimización. Este permite que las expresiones matemáticas que definen los modelos sean interpretadas por Python. Como hemos visto en la revisión del estado del arte, Gurobi ha demostrado ser la elección más eficiente ante otras alternativas.

El script realiza una serie de funciones que se comentan a continuación, a modo de resumen de todo lo

expuesto anteriormente:

1. Lectura y almacenaje de los pedidos y líneas de la hoja de cálculo. Se diferencian los del producto A y del B, ya que se ejecutan dos programaciones por separado.
2. Generación de pedidos iniciales o en curso y pedidos de mantenimiento.
3. Generación de pedidos fraccionados si se supera el límite de pedido máximo permitido. Cuando un pedido supera el límite establecido, el script de lectura de datos modifica el pedido a la cantidad límite y genera tantos pedidos duplicados como sean necesarios, a modo de partición del pedido.
4. Cálculos de tiempos de setup entre pedidos y de tiempos de proceso entre pedidos y líneas. El horizonte temporal y todas las medidas de tiempo se expresan en minutos.
5. Ejecución del modelo preliminar. A partir de la solución obtenida, se almacena para cada pedido el conjunto de líneas de permitidas. Nótese que los pedidos iniciales o en curso y los de mantenimiento ya tienen su respectiva línea asignada.
6. Generación de grupos independientes de pedidos y líneas. Se construyen las intersecciones entre pedidos con líneas compartidas y se separan aquellos que no tengan coincidencias, que podrán ser resueltos por separado.
7. Ejecución del modelo de programación, iterando según los grupos generados. Finalmente se obtienen las secuencias para cada línea y una programación concreta de pedidos a 15 días vista. Se ha establecido un tiempo de ejecución de 30 minutos como máximo para encontrar el óptimo o la mejor solución hasta ese momento.
8. Exportación de resultados. La solución es exportada a una hoja de cálculo en formato de tabla, con el listado de pedidos, sus líneas asignadas y los instantes de inicio y fin. Además, se genera un gráfico Gantt donde se muestra los resultados.
9. Repetir desde el punto 5 para el producto B.

Para corroborar la validez de la herramienta y a fin de comprobar cómo se comporta el modelo ante variaciones, se han realizado testeos con distintas instancias y parámetros. Tras esta comprobación, podemos destacar algunos aspectos importantes para este problema.

Para el modelo preliminar podemos señalar el valor dado a  $lcm$ , o total de líneas permitidas para procesar un mismo color. Como se ha mencionado anteriormente es recomendable partir de  $lcm = 2$  e ir incrementando paulatinamente este valor si la programación resulta no factible. El incrementar hasta un número de líneas cercano al total de las líneas disponibles (12 en el caso del producto A) conlleva una menor congestión de pedidos en las líneas, pero aumenta el tiempo de resolución.

Otro aspecto a considerar es la variación de las velocidades de las líneas. La ligera variación de la tasa de salida entre las líneas es suficiente para que el modelo preliminar seleccione las más veloces en la primera etapa. No obstante, siempre se priorizará la selección de las líneas con producción en curso por la continuidad de la fabricación en éstas.

Pasando al modelo principal de programación, es importante observar y revisar los efectos que produce en el modelo, y como se traslada a la realidad, el fraccionado de los pedidos en función a un tamaño máximo establecido. En la ecuación que lo define,  $PedidoMax = p \cdot \frac{cap}{h}$ , hemos observado que el valor de  $p = 3$  es adecuado para las instancias reales que podemos encontrarnos. Un valor de  $p$  muy alto puede suponer la generación de pedidos con tiempos de proceso excesivamente grandes, lo que se traduce en:

- Ineficiencias en las programaciones de las paradas de mantenimiento. Estas paradas se definen dentro

de un margen o ventana temporal, pero si los pedidos son muy grandes podría no encontrarse un hueco para su comienzo, o bien, se podrían generar tiempos ociosos enormes previamente a la parada de mantenimiento. No obstante, este problema se puede mitigar aumentando el parámetro de tolerancia de las ventanas para las paradas de mantenimiento,  $tpm$ .

- Problema no factible ante fechas de entrega ajustadas. Estos pedidos son menos flexibles para la secuenciación, dificultando su asignación y la de los demás pedidos.

Por el contrario, un valor de  $p$  muy bajo puede provocar la generación de muchos pedidos fraccionados. Como se sabe, en problemas de programación y secuenciación de la producción, el aumento de pedidos o trabajos, incrementa exponencialmente la complejidad del problema, por lo que el tiempo de ejecución podría dispararse.

Para ciertos momentos críticos, es interesante tener en cuenta las repercusiones de las fechas de entrega en el comportamiento del modelo. En (Ho JC, 1995; Pereira Lopes, Manuel J. & de Carvalho, 2007) encontramos una propuesta para la generación de fechas de entrega, con una fórmula que establece como límite inferior el tiempo del trabajo con máximo tiempo de proceso y como superior un valor dado por  $\frac{2 \cdot h}{q}$ , donde  $h$  es el

makespan resultante de insertar cada trabajo en la máquina con el menor tiempo de terminación para ese trabajo, y  $q$  es un parámetro de congestión que al aumentar ajusta la variabilidad en las fechas de entrega. La organización trabaja con tiempos de entrega bastante amplios debido a una cadena de suministro correctamente planificada pero ante situaciones críticas en las que encontremos un problema no factible, podemos variar los parámetros ya comentados: el número de líneas asignables por color,  $lcm$ ; el tamaño de pedido máximo, a través de  $p$ ; y la tolerancia para las ventanas de asignación de las paradas de mantenimiento,  $tpm$ .

Por otro lado, relacionado con los tiempos de setup y de proceso hay que tener en cuenta que se debe satisfacer la llamada desigualdad triangular. A través de la inequación  $s_{ijl} \leq s_{ikl} + p_{kl} + s_{jkl}$ , se evita la situación donde el setup entre dos trabajos  $i$  y  $j$  es mayor que el que se produciría al intercalar otro trabajo  $k$  entre estos dos. Dada la naturaleza de nuestro problema, donde el tiempo de setup es muy corto comparado con los tiempos de proceso de los pedidos, esta desigualdad ya se cumple implícitamente.

Finalmente, cabe mencionar la dificultad debida a las paradas de mantenimiento, situación en la que se puede comprobar un incremento importante de la complejidad, ya que no solo se agregan pedidos nuevos a secuenciar, sino también se generan las variables de ventanas para el comienzo y fin de las paradas. En este tipo de situaciones se debe introducir el parámetro de tolerancia para las ventanas, en nuestro caso un valor suficiente es  $tpm = 60$ , expresado en minutos. Ya se ha expuesto como al aumentar el valor de este parámetro aumenta la flexibilidad en la programación, pero esto no significa necesariamente una mejora en la solución encontrada.

En el siguiente apartado, se expondrán los resultados obtenidos para una instancia real correspondiente al producto A.

### 5.3. Análisis de resultados

Antes de mostrar los resultados que se obtienen, es importante detallar el punto de partida previo al desarrollo de la herramienta.

En la situación real de partida, la programación y el control de la producción los lleva a cabo un planificador,

pero no sigue un procedimiento estándar, se basa en la experiencia y en la revisión exhaustiva de los consumos del stock, que hay que precisar que son bastante elevados. La principal tarea de programación se ejecuta los viernes, para fijar la producción de las líneas de cara a la semana siguiente. El desarrollo de una programación semanal le ocupa la práctica totalidad de la jornada, unas seis horas aproximadamente, más las revisiones y reajustes constantes durante la semana. Cuenta con SAP como generador de peticiones y propuestas de programación de la producción en órdenes de fabricación, pero en la realidad se trabaja con una hoja de cálculo de apoyo, que permite incluir los aspectos que SAP no logra captar. El planificador, ante estos pedidos, realiza los descuentos de almacén siempre que le sea posible y en base al stock disponible asigna las líneas que se van a encargar de producir estos productos, con el fin de reaprovisionar constantemente el almacén. A esto se le suma el control diario que requiere la programación de la producción y sus reajustes debido a pedidos urgentes y otras modificaciones que puedan surgir.

Como vemos la figura del planificador está siempre presente en todo el proceso, lo que convierte esta actividad en un proceso manual basado en la experiencia del empleado y que induce a fallos e ineficiencias. Estas ineficiencias quedan en parte absorbidas por la gran cantidad de líneas de fabricación disponibles y el gran colchón de inventarios para los productos de mayor rotación. Dicho esto, parece conveniente poner en marcha una metodología que permita mejorar la programación de la producción, aportando una herramienta robusta capaz de programar todos los pedidos en fecha y minimizando los tiempos y, por tanto, los costes de setup y el uso de las líneas.

En la siguiente figura se muestra un flujograma a modo de resumen del proceso que se estaría llevando a cabo actualmente, señalando en un marco rojo las actividades que plantea sustituir la herramienta que se ha desarrollado:

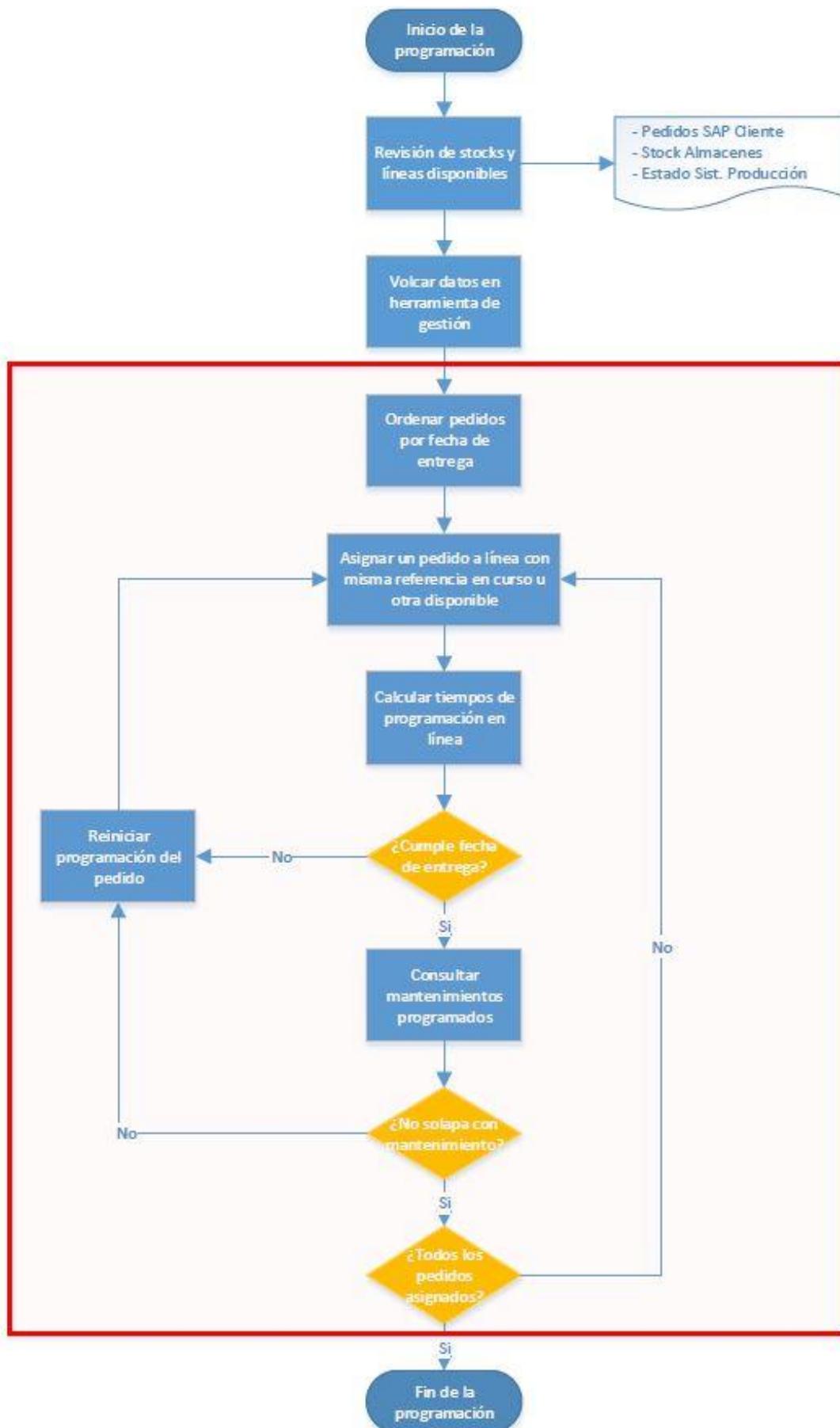


Figura 5.1 Flujograma del proceso de programación por el planificador

A partir de una instancia real se muestra los resultados obtenidos con la herramienta, para finalmente describir la mejora conseguida.

Se trata de un ejemplo real cuyas fechas han sido trasladadas al momento de la ejecución para generar una programación válida. Para empezar, los valores para los parámetros de configuración de los modelos son:

*Tabla 5.3 Parámetros de configuración inicial*

<b>Parámetro</b>	<b>Descripción</b>	<b>Valor</b>
<i>lcm</i>	nº líneas asignables por color	2 líneas
<i>cap</i>	Capacidad promedio de fabricación de línea	15114 uds/1000
<i>PedidoMax</i>	Tamaño del pedido máximo	2940 uds/1000
<i>tpm</i>	Tolerancia máxima de ventanas de mantenimiento	60 minutos
<i>tm</i>	Tardanza máxima permitida para cada pedido	0 minutos

A continuación, se detalla el estado de las líneas disponibles. Cada línea tiene las siguientes tasas de salida:

*Tabla 5.4 Tasas de salida (velocidades) de las líneas*

<b>Líneas</b>	<b>Pérdidas Rendim.</b>	<b>Pérdidas Calidad</b>	<b>Tasas de salida (uds/min)</b>
<b>Línea 1</b>	0,96	0,99	719,05
<b>Línea 2</b>	0,88	0,91	603,47
<b>Línea 3</b>	0,98	0,99	734,68
<b>Línea 4</b>	0,96	0,97	704,53
<b>Línea 5</b>	0,96	0,98	711,79
<b>Línea 6</b>	0,98	0,89	660,47
<b>Línea 7</b>	0,94	0,98	696,32
<b>Línea 8</b>	0,98	0,99	734,68
<b>Línea 9</b>	0,92	0,99	687,79
<b>Línea 10</b>	0,96	0,96	697,26
<b>Línea 11</b>	0,98	0,99	734,68
<b>Línea 12</b>	0,96	0,98	711,79

En la tabla se han reflejado en color verde las líneas más veloces, mientras que en color rojo se muestran las más lentas.

Los pedidos en curso para cada línea son los siguientes:

*Tabla 5.5 Pedidos en curso de las líneas*

<b>Líneas</b>	<b>Referencia en Curso</b>	<b>Color de Ref.</b>	<b>Cantidad Restante (uds/1000)</b>
<b>Línea 1</b>	A100000128	Green	100
<b>Línea 2</b>			
<b>Línea 3</b>	A100000529	Blue	50
<b>Línea 4</b>	A100000036	White	120
<b>Línea 5</b>			
<b>Línea 6</b>			
<b>Línea 7</b>			
<b>Línea 8</b>	A100000529	Blue	520
<b>Línea 9</b>			
<b>Línea 10</b>			
<b>Línea 11</b>	A100000036	White	320
<b>Línea 12</b>	A100000500	Yellow	210

Se han planificado las siguientes paradas de mantenimiento:

*Tabla 5.6 Paradas por mantenimiento preventivo planificado en cada línea*

<b>Líneas</b>	<b>Comienzo Mantenimiento (min.)</b>	<b>Finalización Mantenimiento (min.)</b>
<b>Línea 1</b>	8620	8860
<b>Línea 2</b>	1200	1440
<b>Línea 3</b>		
<b>Línea 4</b>	3800	4040
<b>Línea 5</b>		
<b>Línea 6</b>		
<b>Línea 7</b>		
<b>Línea 8</b>	10500	10740



PA001									
A	White	0128704085/50	A100000036	23 White PA001	MTS	09/11/2020	30/11/2020	1365	
A	Red	0128704088/10	A100000125	23 Red P1805C PA001	MTS	10/11/2020	26/11/2020	1260	
A	Blue	0128704091/20	A100000131	23 Blue P2945C PA001	MTS	10/11/2020	26/11/2020	1575	
A	White	0128704094/10	A100000241	23 White PROMO Plv RG PA001	MTO	10/11/2020	28/11/2020	1260	
A	Orange	0128704095/10	A100000246	27 mm Orange P165C PA001	MTO	10/11/2020	01/12/2020	945	
A	Blue	0128704096/10	A100000131	23 Blue P2945C PA001	MTS	11/11/2020	26/11/2020	1365	
A	Yellow	0128704097/10	A100000777	23 Yellow PROMO Rauch PA001	MTO	11/11/2020	27/11/2020	1365	
A	Black	0128704099/20	A100000874	23 Cap_Base Black PA001	MTO	12/11/2020	30/11/2020	420	
A	Green	0128704102/10	A100000128	23 Green P350C PA001	MTS	14/11/2020	02/12/2020	1050	
A	White	0128704103/20	A100000036	23 White PA001	MTS	15/11/2020	01/12/2020	4200	
A	Red	0128704105/10	A100000125	23 Red P1805C PA001	MTS	16/11/2020	01/12/2020	1575	

La tabla se compone de un conjunto de pedidos de diversas referencias, para solicitudes de distintos clientes o de reaprovisionamiento del almacén. Las fechas de entrega desde la creación del pedido cumplen la condición de superar los 15 días de antelación. Podemos identificar como el pedido 0128704103/20 tiene una cantidad superior al pedido máximo establecido, por lo que se generará un pedido adicional. Vemos también como hay ciertas referencias que son diferentes, aunque comparten un mismo color, como lo son las A100000777 y A100000500 de color amarillo.

En resumen, de cara a la resolución de los modelos y agregando los pedidos en curso y las paradas de mantenimiento, se dispone de un total de 43 pedidos (26 pendientes, 12 iniciales o en curso y 5 de mantenimientos) en 12 líneas.

Una vez plasmados los datos iniciales, procedemos a resolver el problema ejecutando el script con los modelos de optimización. El caso que se procede a resolver ha sido ejecutado en un PC con un procesador AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G 1,80 GHz y una RAM de 4 GB.

Comienza el modelo preliminar en dos fases que logra el óptimo rápidamente, como se muestra en la tabla de resultados:

Tabla 5.8 Resumen de solución obtenida para el modelo preliminar

Función Objetivo	Descripción	Solución óptima	Tiempo de resolución
$Z_{1.1} = \min V - U$	Equilibrado de líneas	$Z_{1.1} = 0$	1,33 segundos
$Z_{1.2} = \min V$	Tiempo de línea máximo	$Z_{1.2} = 8168 \text{ min.}$	0,78 segundos

Visualmente, el resultado obtenido se puede mostrar de la siguiente manera:

L1	Cant: 5873,32 (O24, O25, O30, O44)				Total: 5873 Uds/1000
L3	Cant: 2796 (O26, O37, O41, O46)	Cant: 2759 (O28, O33, O42)	Cant: 444,33 (O31, O34, O48)	Cant: 1,68 (O24, O25, O30, O44)	Total: 6001 Uds/1000
L4	Cant: 2314 (O27, O36, O39, O40, O43, O49)	Cant: 3440,67 (O31, O34, O48)			Total: 5755 Uds/1000
L8	Cant: 2641 (O28, O33, O42)	Cant: 3360 (O32, O35, O47)			Total: 6001 Uds/1000
L11	Cant: 6001 (O27, O36, O39, O40, O43, O49)				Total: 6001 Uds/1000
L12	Cant: 1299 (O26, O37, O41, O46)	Cant: 4515 (O29, O38, O45)			Total: 5814 Uds/1000

Figura 5.2 Asignaciones de colores en líneas

Como vemos, el uso de las líneas se reduce a la mitad, segregando cada color en dos líneas a lo sumo, pero manteniendo reparto equilibrado de la carga. Podemos observar también, cómo este modelo selecciona las líneas más veloces.

Para este caso no ha sido posible crear grupos independientes de pedidos y líneas. No obstante, este modelo ha aportado una reducción a un conjunto de 36 pedidos y 6 líneas. Además, no serán generadas las variables de precedencia  $X_{ij}$  de pedidos que no compartan al menos una línea.

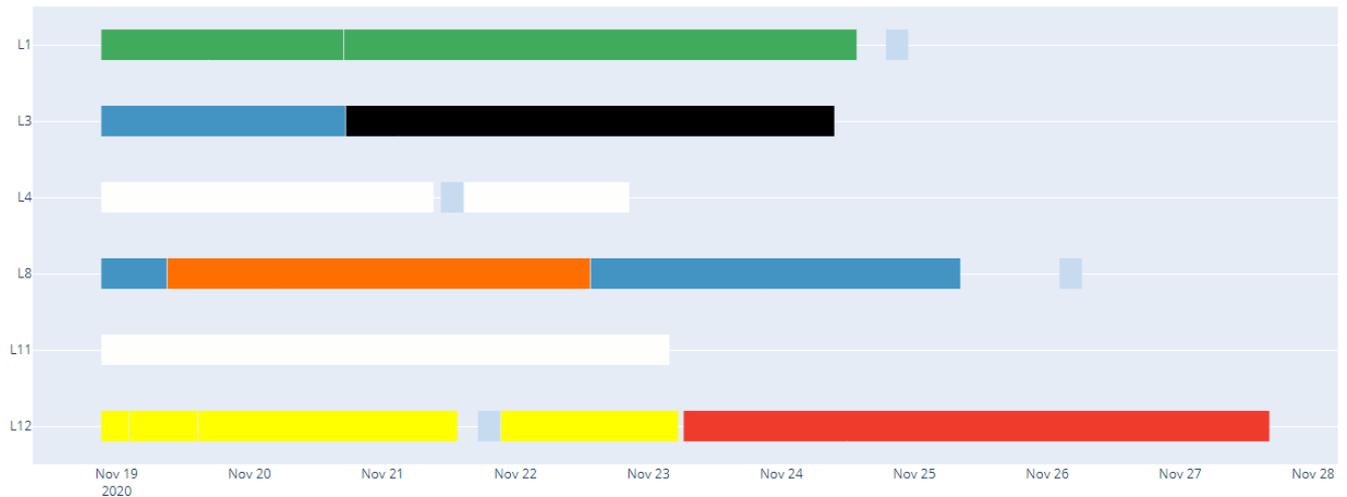
Obtenida esta solución del modelo preliminar, podemos calcular el valor de la ponderación de los tiempos de setup frente a la suma de tiempos de finalización. Tenemos que  $lp = 6$ ,  $\sum ct_i = 35445$  y  $\bar{ts} = 723,24$ ; entonces  $w = 294$ .

Finalmente, la resolución del modelo arroja el siguiente resultado:

Tabla 5.9 Resumen de solución obtenida para el modelo principal

Función Objetivo	Descripción	Solución óptima	Tiempo de resolución
$Z_2 = \min \sum F_i + w \cdot \sum \sum s_{ij} \cdot X_{ij}$	Suma de tiempos de finalización y setup ponderado	$Z_2 = 154407 + 294 \cdot 640 = 342602$	408,73 segundos

Utilizando la visualización de Gantt, común para este tipo de problemas, podemos reflejar la programación de la siguiente forma:



*Figura 5.3 Visualización Gantt de resultados de la programación de la producción*

En la gráfica se muestran las líneas activas L1, L3, L4, L8, L11 y L12 y los pedidos que se procesan en ellas según el color. Los tiempos de setup aparecerán como huecos en las secuencias de las líneas y las paradas de mantenimiento se reflejan con franjas de un color algo más azulado que el fondo. Esta gráfica tiene una interpretación muy visual pero no permite identificar el pedido concreto sobre los colores mostrados.

Para entrar en más detalle, la exportación a la hoja de cálculo desglosa la programación en formato tabular dando la información que aparece a continuación:

ID_Ped	Tipo	Descripcion	Color	Linea	Fecha Inicio	Fecha Fin	Fecha Entrega
O1	En Curso	N/A - A100000128	Green	L01	18/11/2020 21:15	18/11/2020 23:34	18/12/2020 21:15
O24	Regular	0125877785/10 - A100000128	Green	L01	18/11/2020 23:34	19/11/2020 16:36	21/11/2020 23:59
O30	Regular	0126068666/30 - A100000128	Green	L01	19/11/2020 16:36	20/11/2020 16:56	01/12/2020 23:59
O25	Regular	0138201793/10 - A100000503	Green	L01	20/11/2020 17:06	21/11/2020 17:27	22/11/2020 23:59
O44	Regular	0130215502/10 - A100000503	Green	L01	21/11/2020 17:27	24/11/2020 13:35	26/11/2020 23:59
O18	Mntto. Prev.	N/A - N/A	N/A	L01	24/11/2020 18:55	24/11/2020 22:55	18/12/2020 21:15
O3	En Curso	N/A - A100000529	Blue	L03	18/11/2020 21:15	18/11/2020 22:23	18/12/2020 21:15
O42	Regular	0129903036/10 - A100000529	Blue	L03	18/11/2020 22:23	20/11/2020 17:15	21/11/2020 23:59
O31	Regular	0129965550/10 - A100000874	Black	L03	20/11/2020 17:25	21/11/2020 02:57	29/11/2020 23:59
O48	Regular	0129945732/10 - A100000874	Black	L03	21/11/2020 02:57	21/11/2020 17:15	26/11/2020 23:59
O34	Regular	0129945732/10 - A100000874	Black	L03	21/11/2020 17:15	24/11/2020 09:33	29/11/2020 23:59
O4	En Curso	N/A - A100000036	White	L04	18/11/2020 21:15	19/11/2020 00:05	18/12/2020 21:15
O49	Regular	0125943000/10 - A100000036	White	L04	19/11/2020 00:05	19/11/2020 05:03	24/11/2020 23:59
O27	Regular	0125935033/10 - A100000036	White	L04	19/11/2020 05:03	20/11/2020 00:55	24/11/2020 23:59
O43	Regular	0125943000/10 - A100000036	White	L04	20/11/2020 00:55	21/11/2020 09:13	29/11/2020 23:59
O20	Mntto. Prev.	N/A - N/A	N/A	L04	21/11/2020 10:35	21/11/2020 14:35	18/12/2020 21:15
O36	Regular	0125879416/10 - A100000241	White	L04	21/11/2020 14:45	22/11/2020 20:33	27/11/2020 23:59
O8	En Curso	N/A - A100000529	Blue	L08	18/11/2020 21:15	19/11/2020 09:03	18/12/2020 21:15
O35	Regular	0125124538/20 - A100000246	Orange	L08	19/11/2020 09:13	20/11/2020 04:16	26/11/2020 23:59
O47	Regular	0125124538/20 - A100000246	Orange	L08	20/11/2020 04:16	21/11/2020 01:42	30/11/2020 23:59
O32	Regular	0125916581/20 - A100000246	Orange	L08	21/11/2020 01:42	22/11/2020 13:26	29/11/2020 23:59
O28	Regular	0129952017/10 - A100000131	Blue	L08	22/11/2020 13:36	23/11/2020 20:34	25/11/2020 23:59
O33	Regular	0127194782/10 - A100000131	Blue	L08	23/11/2020 20:34	25/11/2020 08:18	25/11/2020 23:59
O21	Mntto. Prev.	N/A - N/A	N/A	L08	26/11/2020 02:15	26/11/2020 06:15	18/12/2020 21:15
O11	En Curso	N/A - A100000036	White	L11	18/11/2020 21:15	19/11/2020 04:30	18/12/2020 21:15
O40	Regular	0130223523/10 - 2 - A100000036	White	L11	19/11/2020 04:30	20/11/2020 09:05	30/11/2020 23:59
O39	Regular	0130223523/10 - A100000036	White	L11	20/11/2020 09:05	23/11/2020 03:47	30/11/2020 23:59
O12	En Curso	N/A - A100000500	Yellow	L12	18/11/2020 21:15	19/11/2020 02:10	18/12/2020 21:15
O46	Regular	0128704050/10 - A100000777	Yellow	L12	19/11/2020 02:20	19/11/2020 14:37	19/11/2020 23:59
O37	Regular	0130230841/10 - A100000500	Yellow	L12	19/11/2020 14:47	20/11/2020 10:28	27/11/2020 23:59
O26	Regular	0125879440/10 - A100000500	Yellow	L12	20/11/2020 10:28	21/11/2020 13:30	23/11/2020 23:59
O22	Mntto. Prev.	N/A - N/A	N/A	L12	21/11/2020 17:15	21/11/2020 21:15	18/12/2020 21:15
O41	Regular	0128704050/10 - A100000777	Yellow	L12	21/11/2020 21:25	23/11/2020 05:23	26/11/2020 23:59
O29	Regular	0130209688/10 - A100000125	Red	L12	23/11/2020 06:23	24/11/2020 11:53	25/11/2020 23:59
O45	Regular	0130213293/10 - A100000125	Red	L12	24/11/2020 11:53	26/11/2020 03:13	26/11/2020 23:59
O38	Regular	0126068666/10 - A100000125	Red	L12	26/11/2020 03:13	27/11/2020 16:06	30/11/2020 23:59

Figura 5.4 Programación de la producción detallada en la hoja de cálculo

La tabla muestra los pedidos ordenados por líneas e instante de inicio en cada línea. En la primera columna se indica el acrónimo del pedido dado por el script. La segunda clasifica la procedencia de cada pedido, pedido en curso, de mantenimiento o pedido regular pendiente de programar. La descripción concatena el pedido del sistema con la referencia, si aplica. También aparecen las líneas a las que han sido asignados los pedidos, con los instantes de inicio, fin y la fecha de entrega comprometida.

En la gráfica de Gantt podemos observar, cómo los pedidos de las mismas referencias o colores similares se procesan en serie, de forma que se mantiene un flujo constante que minimiza los cambios de configuración de las líneas.

Parece que se produce una excepción de esta lógica en las líneas L3 y L8 al dividir el procesado de piezas de color azul, sin embargo, si nos fijamos en la tabla de detalles, vemos como pertenecen a distintas referencias, es decir, es otro tipo de azul y habría que realizar un setup de igual forma que se produce en los cambios a naranja o negro.

Se ha conseguido programar la totalidad de los pedidos evitando frenar la producción por las paradas de mantenimiento de las líneas L1 y L8. Por tanto, cabría esperar que esas líneas se paren al finalizar esos pedidos, a menos que surja algún pedido sobrevenido que puede intercalarse antes de los mantenimientos. Las planificaciones de todos los mantenimientos se han visto desplazadas por el modelo dentro del intervalo

máximo permitido.

Como muestran los resultados, se dispone de un sistema de producción sobredimensionado ya que no es necesario el uso de todas las líneas. Esto se debe a que el valor de la pieza fabricada es despreciable respecto del producto final, por lo que se tiene un alto nivel de stock y múltiples líneas para poder abastecer cualquier pico de producción que pueda llegar, tratando de esta forma de evitar fallos en el suministro a clientes.

Sin embargo, es evidente la mejora conseguida si tenemos en cuenta que el planificador invierte unas 6 horas en elaborar una propuesta de programación, en la cual no se garantiza una disposición óptima, frente a una herramienta que permite generar una programación en, apenas 7 minutos, obteniendo una solución óptima para los objetivos planteados. Esta mejora puede trasladarse también al control de la producción debido a las reprogramaciones que puedan surgir, en ocasiones de manera urgente, en cuyas situaciones esta herramienta sería un apoyo ideal a la toma de decisiones.

Por último, de cara a integrar esta herramienta a la actividad diaria, debemos realizar unas consideraciones finales sobre posibles carencias u otras excepciones que se pueden encontrar.

Existe una problemática que subyace en la elección de las líneas que se activan o desactivan en cada ejecución. Este planteamiento puede resultar válido si se trabaja con una demanda lo suficientemente estable. De no ser así, las altas fluctuaciones en los pedidos entrantes pueden provocar continuos desajustes en las líneas que se mantienen en funcionamiento, lo que no es admisible debido a la necesaria estabilidad y antelación que requiere la planificación del personal encargado de supervisar y realizar las operaciones en las líneas. Además, a esto se suma el alto coste de arranque de las líneas, aunque podría ser modelado como un setup penalizado en la función objetivo. Por tanto, se puede entender la elección de las líneas como una decisión a nivel táctico frente a la visión operativa que es la programación de los pedidos. Para este enfoque basta con ejecutar el modelo preliminar una única vez de manera trimestral en base a los volúmenes acordados con el cliente y obtener una disposición de las líneas fijas a medio plazo, quedando el modelo principal para las programaciones diarias.

Cuando surgen picos de demanda para ciertas referencias o colores puede ser necesario que tal referencia o color sea producido en varias líneas simultáneamente. Esto se soluciona aumentando el número de líneas permitidas,  $lcm$ , en el modelo preliminar. Sin embargo, este parámetro se aplica a todos los colores por igual, de forma que para el resto de colores se permitirían más asignaciones a líneas de las necesarias. Esto se puede solventar adaptando este parámetro para que sea específico para cada color, en función del volumen a procesar sobre el total:  $lcm_i$ .

Pese a la posible necesidad de estos últimos ajustes, podemos decir que se ha desarrollado una herramienta que aporta un método ágil para la programación de la producción enfocada a la reducción de los costes de setup, seleccionando el uso de las líneas que sean necesarias, y sin dejar de lado el compromiso de entrega de los pedidos. A su vez, considera las líneas que se encuentran en funcionamiento y las paradas de mantenimiento planificadas, para hacer que esta herramienta sea completamente funcional en el día a día.



## 6 CONCLUSIONES Y POSIBLES MEJORAS

---

**E**n el presente trabajo se ha planteado el problema de programación de máquinas paralelas uniformes con el objetivo de minimizar los tiempos de setup, garantizando el cumplimiento de las fechas de entrega, aplicado a un entorno real y con enfoque de resolución exacta. Se ha desarrollado una herramienta práctica, que asume que las líneas estén en funcionamiento con pedidos en curso en el momento de la resolución y las que considera paradas de mantenimiento planificadas, evitando que puedan frenar la producción. Esto permite que sea utilizada de forma ágil para realizar reprogramaciones del sistema de producción.

Se ha logrado resolver una instancia real de unos 43 pedidos, incluyendo pedidos en curso y mantenimientos, en 12 posibles líneas en un tiempo de tan sólo 7 minutos. Esto ha sido posible al abordar el problema en dos fases. Una primera fase en la que se resuelve un modelo de optimización para seleccionar las líneas necesarias y, en función del color, preasignar colores a líneas. Y una segunda fase de programación de la producción.

Podemos indicar dos líneas futuras de mejora, continuando con este enfoque o adoptando nuevas técnicas de resolución. Siguiendo la metodología planteada podemos mencionar los siguientes puntos:

- Profundizando en un modelo preliminar que seleccione las líneas y los colores que pueden procesarse en ellas, se podría elaborar un algoritmo exacto iterativo de enumeración completa de los pares de líneas a los que pueda asignarse cada color.
- A la hora de establecer y calcular un pedido de tamaño máximo, vimos que esto podría desembocar en la generación de varios pedidos, lo que podría complicar en exceso la resolución del modelo. Se podría plantear otra forma de realizar las particiones, orientada al número de pedidos máximos a obtener, de esta forma se podría mantener el controlado el tamaño del problema.
- Otro punto a contemplar podría consistir en resolver un problema más sofisticado y complejo, que tenga en cuenta otros recursos como pueden ser el personal, los materiales o las tareas que no puedan desarrollarse de forma simultánea.
- En esta línea podría ser interesante también considerar, de manera más explícita a la que se ha planteado, el gasto energético de las líneas, el consumo de materia prima o los costes y desperdicios en los setup.
- La optimización de la programación de la producción puede sacar a relucir ineficiencias en el sistema a otros niveles. Por ejemplo, estudios posteriores de reducción de los niveles de stocks y planificación del uso de líneas son interesantes.

Dejando a un lado la propuesta seguida en el presente trabajo, existen otras opciones a valorar a la hora de enfrentar un problema de este calado. Como, por ejemplo:

- Desarrollo de algoritmos exactos basados en las propiedades del problema. Por ejemplo, generando lotes de pedidos en función de los colores, adaptando el problema de la mochila para la secuenciación cuando existan paradas de mantenimiento, etc.
- Desarrollo de metaheurísticas. En la literatura se ha comprobado cómo el uso de metaheurísticas es la mejor elección cuando se buscan buenas soluciones en un corto plazo de tiempo en el caso de problemas de gran tamaño.
- La fiabilidad de los datos no siempre es exacta. Más aún cuando se trabaja en un entorno real en el que intervienen múltiples factores que pueden modificar las condiciones del sistema de producción. Por ejemplo, al ejecutar los modelos se suponen unos rendimientos en las líneas durante el periodo de la programación que podrían sufrir alteraciones. Los modelos estocásticos, son una opción a tener en cuenta para considerar este tipo de casuísticas.

## REFERENCIAS

- Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3), 985–1032.
- Avalos-Rosales, O., Angel-Bello, F., Álvarez, A., & Cardona-Valdés, Y. (2018). Including preventive maintenance activities in an unrelated parallel machine environment with dependent setup times. *Computers and Industrial Engineering*, 123, 364–377.
- Baez, S., Angel-Bello, F., & Alvarez, A. (2016). Time-dependent formulations for minimizing total completion time in a parallel machine scheduling problem with dependent setup times. *IFAC-PapersOnLine*, 49(12), 857–862.
- Báez, S., Angel-Bello, F., Alvarez, A., & Melián-Batista, B. (2019). A hybrid metaheuristic algorithm for a parallel machine scheduling problem with dependent setup times. *Computers and Industrial Engineering*, 131(March 2018), 295–305.
- Balakrishnan, N., Kanet, J. J., & Sridharan, “Sri” V. (1999). Early/tardy scheduling with sequence dependent setups on uniform parallel machines. *Computers and Operations Research*, 26(2), 127–141.
- Christ, Q., Dauzère-Pérès, S., & Lepelletier, G. (2019). An Iterated Min–Max procedure for practical workload balancing on non-identical parallel machines in manufacturing systems. *European Journal of Operational Research*, 279(2), 419–428.
- Chuzhoy, J., & Codenotti, P. (2009). Resource minimization job scheduling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5687 LNCS, 70–83.
- Fanjul-Peyro, L. (2020). Models and an exact method for the Unrelated Parallel Machine scheduling problem with setups and resources. *Expert Systems with Applications: X*, 5, 100022.
- Fanjul-Peyro, L., Ruiz, R., & Perea, F. (2018). Reformulations and an exact algorithm for unrelated parallel machine scheduling problems with setup times. *Computers and Operations Research*, 101, 173–182.
- Framinan, J. M., Leisten, R., & Ruiz García, R. (2014). Manufacturing Scheduling Systems. In *Manufacturing Scheduling Systems*. Springer. Springer-Verlag. London.
- Hartmut Stadtler, C. K. (2005). *Supply Chain Management and Advanced Planning*. Springer. Springer Science+Business Media. Berlin (Germany).
- Hillier, F. S. and Lieberman, G. J. (2010). *Introducción a la Investigación de Operaciones*. Pablo E. Roig V. McGRAW-HILL/INTERAMERICANA EDITORES, S.A. Mexico.

- Ho JC, C. Y.-L. (1995). *Minimizing the number of tardy jobs for m parallel machines*. *European Journal of Operational Research* (1995) 84(2) 343-355.
- Kim, D. W., Na, D. G., & Chen, F. F. (2003). Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. *Robotics and Computer-Integrated Manufacturing*, 19(1–2), 173–181.
- Lenstra, J., Rinnooy Kan, A., & Brucker, P. (1977). *Complexity of machine scheduling problems*. *Annals of Discrete Mathematics*. Volume 1, 1977, Pages 343-362.
- Manne, A. S. (1960). On the Job-Shop Scheduling Problem. *Operations Research*. Volume 8, Issue 2, Pages 159-294.
- Mazdeh, M. M., Zaerpour, F., Zareei, A., & Hajinezhad, A. (2010). Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs. *Applied Mathematical Modelling*, 34(6), 1498–1510.
- McCormick, S. T., & Pinedo, M. L. (1995). Scheduling n Independent Jobs on m Uniform Machines with both Flowtime and Makespan Objectives: A Parametric Analysis. *ORSA Journal on Computing*, 7(1), 63–77.
- Osorio-Valenzuela, L., Pereira, J., Quezada, F., & Vásquez, Ó. C. (2019). Minimizing the number of machines with limited workload capacity for scheduling jobs with interval constraints. *Applied Mathematical Modelling*, 74, 512–527.
- Pereira Lopes, Manuel J. & de Carvalho, J. M. V. (2007). *A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times*. *European Journal of Operational Research*, Elsevier, vol. 176(3), pages 1508-1527.
- Pinedo, M. L. (2005). *Planning and Scheduling in Manufacturing and Services*. Springer. Springer Science+Business Media. New York (USA).
- R.L. Graham, E.L. Lawler, J.K. Lenstra, A. H. G. R. K. (1979). *Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey*. *Annals of Discrete Mathematics*. Volume 5, pages 287-326.
- Rocha, P. L., Ravetti, M. G., Mateus, G. R., & Pardalos, P. M. (2008). Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers and Operations Research*, 35(4), 1250–1264.
- Vignier, A., Billaut, J. C., & Proust, C. (1999). Les problèmes d'ordonnancement de type flow-shop hybride: État de l'art. In *RAIRO - Operations Research*. Volume 33, Issue 2, pp. 117-183.
- Wang, S., & Liu, M. (2015). Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning. *Journal of Manufacturing Systems*, 37, 182–192.
- Xu, D. H., Sun, K. B., & Li, H. X. (2006). Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. *Computers and Operations Research*, 35(4), 1344–1349.
- Yeh, W. C., Chuang, M. C., & Lee, W. C. (2015). Uniform parallel machine scheduling with resource consumption constraint. *Applied Mathematical Modelling*, 39(8), 2131–2138.