UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIERÍA INDUSTRIAL VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

ANÁLISIS Y DISEÑO DE UN MODELO MATEMÁTICO PARA LA PLANIFICACIÓN DE LA PRODUCCIÓN CON CAPACIDADES Y FECHAS DE ENTREGA Y COMPARATIVA CON UN MRP CLÁSICO

AUTOR: Beatriz Jin Morales Azara

TUTOR: María Pilar Tormos Juan

COTUTOR: Rubén Ruiz García

Curso Académico: 2020-21

RESUMEN

El problema de la planificación de la producción es uno de los retos a los que se enfrentan diariamente las empresas. Para solucionar dicho problema existen diversas técnicas, como por ejemplo la Planificación de Requerimientos de Materiales (MRP). Esta técnica en muchos casos puede resultar ineficiente. Es por este motivo, que surgen otras técnicas como algoritmos heurísticos y modelos matemáticos, capaces de resolver el problema de la planificación de la producción con mayor eficiencia. El presente trabajo tiene como objetivo diseñar un modelo matemático para la planificación de la producción con capacidades y fechas de entrega. Este modelo aúna la misma funcionalidad que tiene el algoritmo heurístico de un MRP desarrollado por la empresa GeInfor y que está en estudio para su implantación en varias empresas finalistas. Para formular dicho modelo, se parte del modelo estándar de la planificación de la producción y sobre este se crearán sucesivos modelos añadiendo nuevas variables y restricciones. La incorporación de más y más variables y restricciones hace que sea necesario emplear distintas herramientas de modelización, debido al aumento de la dimensión del problema. La finalidad será la de estudiar el proceso de formulación del modelo matemático y analizar las diferencias existentes entre las soluciones óptimas obtenidas.

Palabras clave: optimización; modelo matemático; MRP; programación lineal entera mixta; demanda dependiente.

RESUM

El problema de la planificació de la producció és un dels reptes als que s'enfronten diàriament les empreses. Per a solucionar aquest problema existixen diverses tècniques, com per exemple, la Planificació de Requeriments de Materials (MRP). Aquesta tècnica en molts casos pot resultar ineficient. És per aquest motiu, que sorgixen altres tècniques com a algorismes heurístics i models matemàtics, capaços de resoldre el problema de la planificació de la producció amb major eficiència. El present treball té com a objectiu dissenyar un model matemàtic per a la planificació de la producció amb capacitats i dates de lliurament. Aquest model conjumina la mateixa funcionalitat que té l'algorisme heurístic d'un MRP desenvolupat per l'empresa GeInfor i que està en estudi per a la seua implantació en diverses empreses finalistes. Per a crear aquest model, es partix del model estàndard de la planificació de la producció i sobre aquest es crearan successius models afegint noves variables i restriccions. La incorporació de més i més variables, fa que siga precís emprar diferents eines de modelització. La finalitat serà la d'estudiar el procés de formulació del model matemàtic i analitzar les diferències existents entre les solucions òptimes obtingudes.

Paraules clau: optimització; model matemàtic; MRP; programació lineal entera mixta; demanda dependent.

ABSTRACT

The production planning problem is one of the challenges that companies face on a daily basis. In order to solve this problem, there are various techniques such as the Materials Requirements Planning (MRP). This technique can be inefficient in many cases. Therefore, other techniques such as heuristic algorithms and mathematical models have emerged. As they are able to solve production planning problems more efficiently. The aim of the current work is to design a mathematical model for production planning with capacities and delivery dates. This model combines the same functionality as the MRP heuristic algorithm developed by the company GeInfor and which is under study to implement in several final companies. To formulate this model, we will start from the standard production planning model. From this one, successive models will be created with the addition of new variables and constraints. The inclusion of more and more variables and constraints makes it necessary to use different modelling tools, due to the increase in size of the problem. The purpose will be to study the formulation process, as well as, to analyse the differences between the optimal solutions obtained.

Keywords: optimization; mathematical model; MRP; mixed-integer linear programming; dependent demand.

ÍNDICE GENERAL

DOCUMENTOS CONTENIDOS

- Memoria
- Anexos
- Presupuesto

I. ÍNDICE DE LA MEMORIA

II. ÍNDICE DE LOS ANEXOS

III. ÍNDICE DEL PRESUPUESTO

IV. ÍNDICE DE TABLAS

V. ÍNDICE DE FIGURAS

VI. ÍNDICE DE LISTADOS

MEMORIA

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

La Investigación Operativa (*Operations Research*, OR), tiene su origen en los inicios de la Segunda Guerra Mundial, así lo afirman Gass & Assad (2006). Durante la guerra, los recursos eran escasos y las operaciones militares debían de ser ejecutadas de manera eficiente, asignando recursos de forma óptima a las diversas actividades. Tras ver el éxito de la OR en el ámbito militar, este método científico se extendió para resolver problemas de otros ámbitos como la industria. Con el paso del tiempo, la complejidad dentro de las organizaciones creció e hizo que cada vez fuese más difícil asignar los recursos disponibles a las actividades, dando una mayor trascendencia a la OR.

Existen muchas definiciones asociadas a OR, según Taha (2012) es la aplicación del método científico en la toma de decisiones de problemas relacionados con el control de las organizaciones. Mediante el uso de técnicas cuantitativas la OR, trata de coordinar las distintas actividades u operaciones dentro de una organización para conseguir el mejor resultado posible.

Una de las áreas objeto de estudio de la OR son los modelos de la planificación de la producción. Estos modelos ayudan a las empresas dedicadas al sector de la producción a decidir, por ejemplo, en qué momento deben de fabricar y qué cantidades deben de producir. Estas decisiones, atienden siempre a las limitaciones de la empresa, como pueden ser de capacidad de fabricación o de almacenaje. Todo esto se hace sin perder de vista un objetivo, generalmente las empresas buscan satisfacer objetivos económicos como maximizar los beneficios o minimizar los costes.

1.2 Justificación

La necesidad de planificar la producción surge debido a la rápida evolución que sufren las empresas de manera constante y es por ello, que tratan de implementar sistemas de gestión que les permita ser al mismo tiempo competitivas y eficientes, para garantizar así su estabilidad y permanencia en el mercado.

Es por ello, que surgen empresas tecnológicas que buscan desarrollar softwares y programas para la planificación de la producción. Es el caso de la empresa GeInfor (https://geinfor.com/). Actualmente la empresa se encuentra desarrollando un algoritmo heurístico de la Planificación de Requerimientos de Materiales (*Material Requirement Planning*, MRP) dentro de su software “ERP Industria 4.0”. Este algoritmo se está diseñando con el objetivo de proveer a las empresas de información para decidir cuándo deben realizarse los lanzamientos de producción y compra de materiales, teniendo en cuenta las restricciones de capacidad, almacenamiento y recursos que puedan presentar.

De forma paralela a los algoritmos heurísticos, están los modelos matemáticos o modelos de optimización. Ambas técnicas, pueden aplicarse a la planificación de la producción, sin embargo, la diferencia fundamental entre un algoritmo heurístico y un modelo matemático es el tipo de solución que son capaces de ofrecer y el tiempo de cómputo que necesitan para llegar a la solución.

La heurística, por una parte, resuelve los problemas proporcionando una solución aceptable, que no tiene por qué ser la mejor (y raramente lo será), pero dentro de un tiempo de cómputo admisible. En contraste, los modelos matemáticos ofrecen siempre una solución óptima, pero a costa de un tiempo de cómputo superior. Por este motivo, en este Trabajo Fin de Grado (TFG) se estudia la formulación de un modelo matemático que facilite una solución óptima al problema de la planificación de la producción y que aúne la misma funcionalidad que el algoritmo MRP de GeInfor. De este modo, se podrán estudiar las diferencias entre la solución que proporciona el modelo matemático y el algoritmo heurístico.

Cabe destacar, que la originalidad de este trabajo reside en el hecho de que la gran mayoría de los modelos desarrollados para la planificación de la producción, pretenden satisfacer un objetivo económico como maximizar beneficios o minimizar costes. En cambio, en este TFG, el objetivo es minimizar la diferencia existente entre la fecha de entrega efectiva de un pedido y la fecha de entrega de demanda de este. En otras palabras, se trata de entregar los pedidos en el periodo más próximo posible al periodo de demanda, pero permitiendo flexibilidad en la entrega, algo que las técnicas de Planificación de Requerimientos de Materiales (MRP) no hacen, según indican Ptak & Smith (2011).

1.3 Objetivos

Por todo lo comentado con anterioridad, en este TFG se analizará y formulará un modelo matemático, que replique las condiciones y la finalidad que desempeña el algoritmo MRP de la empresa GeInfor, trabajando con fechas de entrega y capacidades.

Para diseñar este modelo, será necesario formular previamente varios modelos hasta dar con el que reúna todas las especificaciones buscadas. Cada uno de los modelos planteados, se implementará y posteriormente se estudiará la evolución de las distintas soluciones obtenidas, para distintos escenarios hipotéticos.

Dado que conforme se avance en el trabajo, el modelo crecerá de tamaño, pues aumentarán el número de restricciones y variables, será preciso emplear distintas herramientas de modelización. En un inicio se partirá con modelos simples fácilmente resolubles con el Solver de la herramienta de Microsoft Excel (Excel) y se finalizará viendo la implementación en el Solver profesional IBM ILOG CPLEX (CPLEX). Está segunda herramienta de optimización no se ha visto en el Grado de Tecnologías Industriales (GITI), y es la propia alumna quien aprenderá a programar en dicho lenguaje de modelización, para hacer uso de la herramienta en el trabajo.

1.4 Estructura del trabajo

Para concluir este primer capítulo, se describe la estructura del presente TFG.

- Capítulo 2 - ANTECENDENTES: se describirá en qué consiste el MRP, cuáles son las especificaciones del algoritmo en desarrollo por GeInfor a incluir en el modelo matemático y por tanto cuál es el problema a resolver.

- Capítulo 3 - MODELIZACIÓN MATEMÁTCA: se explicarán los pasos básicos para formular un modelo matemático genérico y brevemente se describirá la programación lineal (*Linear Programming*, LP) y la programación lineal entera mixta (*Mixed Integer Programming*, MIP). Así como una descripción de las herramientas de modelización que se van a emplear, Excel y CPLEX.

- Capítulo 4 - PROPUESTA DEL MODELO: se presentará la formulación de los modelos de manera cronológica para observar la evolución y cambios que se presenten de un modelo a otro. Se partirá, en un inicio del modelo estándar de la producción, y sobre este, se irá modificando la función objetivo y añadiendo nuevas restricciones y variables en cada caso. Llegando a un modelo final, que replique todas las condiciones que presenta el algoritmo de la empresa.

- Capítulo 5 - SOLUCIÓN DE LOS MODELOS: se implementará cada uno de los modelos en Excel o en CPLEX según el tamaño de este. Y a partir de unos casos de ejemplo se estudiarán las soluciones obtenidas. En el último apartado, se comparará la solución que provee el modelo y la que facilitaría el algoritmo MRP.

- Capítulo 6 y 7 - CONCLUSIONES y LÍNEAS DE FUTURO: se extraen las ideas principales que se han obtenido a partir de los resultados y se plantean posibles líneas de trabajo futuro.

2 ANTECEDENTES

En este segundo capítulo se explica qué es y cómo funciona la Planificación de Requerimiento de Material. Después, se ve el funcionamiento del algoritmo MRPc de la empresa GeInfor a partir del cual, se extraerá el planteamiento del problema a resolver. La solución al problema se presenta en el capítulo 4 con la propuesta de modelo matemático.

2.1 Material Requirement Planning

La Planificación de Requerimientos de Materiales (*Material Requirement Planning*, MRP) es un sistema de información que surgió en los años 70, cuya finalidad es transformar las necesidades de un Plan Maestro de Producción (*Master Production Schedule*, MPS) en la programación de órdenes de compra de materiales y órdenes de fabricación. En estas órdenes se indica las cantidades exactas y las fechas en las que se deben realizar (Ptak & Smith, 2011).

Según Pochet & Wolsey (2006) el MRP se emplea generalmente para horizontes de planificación de corto plazo y resulta interesante para resolver problemas en los que existe una estructura multinivel de los productos. Se entiende como estructura multinivel aquella en la que unos productos pueden interactuar con otros productos. Es decir, que un mismo producto puede ser resultado de un producto de un paso anterior, incluso ser necesario para la elaboración de un producto de un paso siguiente. Dentro de esta estructura multinivel, encontramos tres categorías de productos: materias primas, semielaborados y productos finales. Por tanto, el MRP identifica qué componentes y qué materiales son necesarios para producir los productos finales.

Figura 1. Entradas del MRP. Fuente: Elaboración propia a partir de Andrés (2020).

Los datos de entada necesarios para un MRP son los siguientes, véase Figura 1:

- El Plan Maestro de Producción (MPS): en él se determinan las cantidades de los productos finales necesarias para satisfacer la demanda independiente. Se denomina demanda independiente o externa, a aquella que proviene de demandas del mercado.

- La Lista de Materiales (*Bill of Materials*, BOM): es un diagrama que describe la relación entre los distintos tipos de productos (materias primas, semielaborados y productos finales) y por tanto indica qué componentes son necesarios para producir los productos finales y en qué cantidades. Esto se identifica como la demanda dependiente, puesto que, dada la relación subyacente entre los productos, la demanda de un producto está sujeta a la demanda de otro que se encuentre en un nivel de fabricación superior. El BOM presenta una estructura arborescente, donde existe una relación padre-hijo entre un nivel y el nivel inmediatamente inferior al que esté ligado. En el primer nivel (nivel superior) se encuentran los productos finales, en los niveles intermedios, los semielaborados y en el último nivel (nivel inferior) se encuentran las materias primas. Véase Figura 2.

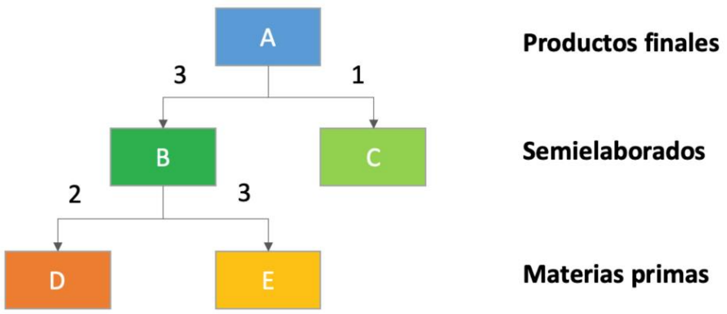

Figura 2. Ejemplo BOM. Fuente: Elaboración propia.

En este ejemplo aparecen 5 productos distintos (A, B, C, D y E), de los cuales, A es el producto final. Mientras que B y C son semielaborados y D y E son materias primas. Las cantidades especificadas en la relación de un producto a otro son las unidades necesarias para producir una unidad del producto del nivel superior. En este caso, para producir una unidad de A serían necesarias 3 unidades de B y 1 de C, y al mismo tiempo para producir una unidad de B se necesitarían 2 de D y 3 de E.

- El registro de inventario: se trata de una plantilla periodificada en la cual se almacena información sobre el estado del inventario en cada uno de los periodos. Se recopila información tal como las necesidades brutas de los productos, la cantidad disponible en almacén, cuándo se producen los lanzamientos de las órdenes, las necesidades netas y las órdenes en curso.

De forma simplificada, el MRP para cada uno de los productos de los distintos niveles parte de las necesidades brutas (NB). Estas vienen dadas por la demanda independiente del MPS para los productos finales y por la demanda dependiente para los semielaborados y materias primas. Después se calculan las necesidades netas (NN), esto es decir que se comprueba si la cantidad de stock existente

(IP) y las órdenes en curso lanzadas anteriormente (OC), son capaces de satisfacer las necesidades brutas. Y por último se determina en qué periodos se deben de lanzar las órdenes (PPL) y en qué cantidad ($RPPL$) para que los productos lleguen a tiempo para satisfacer las necesidades brutas. La cantidad de tiempo necesaria para disponer de un producto se denomina tiempo de suministro (Ts) o *lead-time*. Este tiempo suele variar según el producto, puesto que depende de diversos factores como, las operaciones de transporte que necesite, las tareas administrativas que conlleve o por el procesado específico que requiera un producto.

Este MRP se calcula aplicando simplemente las siguientes fórmulas (Andrés, 2020):

$$NN_t = NB_t - IP_t - OC_t \quad ,\forall t$$

$$Si\ NN_t < 0 \rightarrow NN_t = 0 \quad ,\forall \text{t}$$

$$IP_t = IP_{t-1} + OC_{t-1} + RPPL_{t-1} - NB_{t-1} \ ,\forall \text{t}$$

$$RPPL_t = Lotificación\ (NN_t; NN_{t+1}; \dots;\ NN_{n)} \quad ,\forall \text{t}$$

$$PPL_t = RPPL_{t+TS} \ ,\forall \text{t}$$

Donde:

- t: Periodos que va de 1 a n.
- Ts: Tiempo de suministro.
- NB: Necesidades brutas.
- NN: Necesidades netas.
- IP: Stock disponible.
- OC: Órdenes en curso.
- $RRPL$: Cantidad de producto que se planea recibir.
- PPL: Cantidad de producto que se debe lanzar.

La lotificación indica la técnica de dimensionamiento de lote que se emplee.

El MRP que se ha presentado se denomina MRP estándar o de capacidad infinita puesto que las órdenes de fabricación y compra no se ven limitadas a ninguna restricción de capacidad. Se observa que no hay ninguna ecuación que contenga ningún término relativo a las capacidades. Entiéndase por capacidad cualquier medida limitante, como puede ser financiera, de fabricación, de almacenaje, etc. Por lo que este MRP presenta capacidad ilimitada y, por tanto, es capaz de satisfacer todas las demandas.

En resumen, el sistema MRP estándar determina la cantidad y la fecha en la que se deben de producir los productos para satisfacer las necesidades impuestas por el MPS (demanda independiente) y atendiendo a la relación existente entre los productos detallada en el BOM (demanda dependiente) sin presentar ninguna restricción de capacidad, pero no es útil cuando en cada periodo existen restricciones de capacidad, algo que, por otra parte, es lo más habitual en las empresas.

2.2 Algoritmo MRP GeInfor

La empresa GeInfor, Gestión Empresarial Informatizada, (https://geinfor.com/) es una empresa valenciana del sector tecnológico, que ofrece software para optimizar la producción y aumentar la eficiencia de las empresas.

Tal y como se ha mencionado en el capítulo 1, actualmente, GeInfor se encuentra desarrollando un nuevo algoritmo MRP. Este nuevo algoritmo se puede considerar como una evolución del MRP estándar, ya que la novedad de este reside en incluir las restricciones de capacidad.

En adelante se denominará a este algoritmo como MRPc (c, dado que está sujeto a restricciones de capacidad). Por tanto, el MRPc calculará en qué momento y en qué cantidades se deberán realizar las compras de materiales y la fabricación de productos, teniendo en cuenta las restricciones de capacidad que puedan existir.

A continuación, se muestra un esquema simplificado del algoritmo del MRPc, para comprender de forma general cómo funciona.

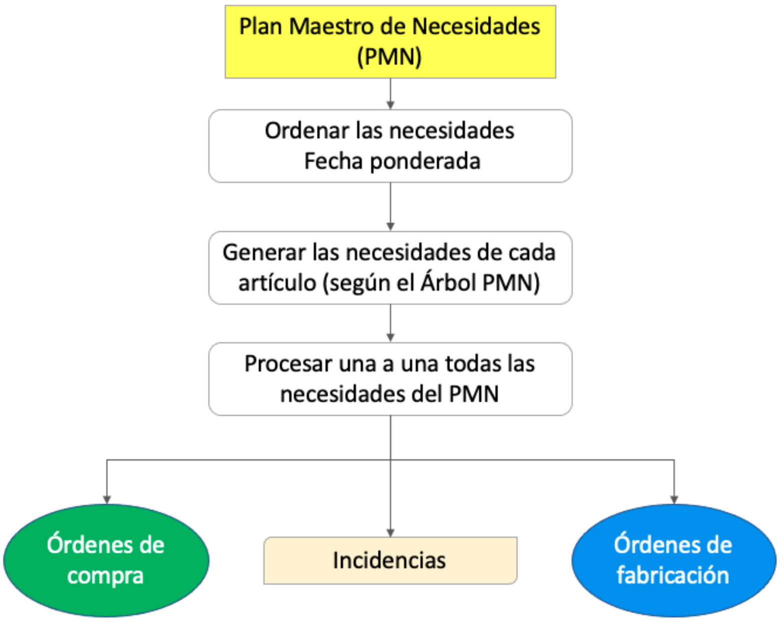

Figura 3. Esquema general del algoritmo MRPc. Fuente: Elaboración propia a partir del esquema proporcionado por GeInfor.

En primer lugar, se genera el Plan Maestro de Necesidades (PMN), en él se indican los pedidos demandados y los productos que se engloban dentro de cada uno de los pedidos. Estos pedidos y productos tendrán asociados unas necesidades que habrá que cumplir para poder satisfacer las demandas. Después, las necesidades se ordenan según el parámetro que denominan fecha ponderada. La fecha ponderada es un valor que tiene en cuenta la prioridad o importancia que se le otorga a cumplir con un pedido y la fecha en la que se debe entregar el mismo. Es decir, aquellos pedidos que se consideren más importantes y tengan la fecha de entrega más temprana, serán los pedidos que el algoritmo intentará satisfacer primero. Una vez estudiado el orden en el que se deben procesar los pedidos, es necesario estudiar las necesidades de los artículos o productos que componen los pedidos, a partir de lo que denominan Árbol PMN. El Árbol PMN es la estructura del algoritmo que indica la manera de procesar las necesidades, véase Figura 4 y que en esencia es lo mismo que la lista de materiales. Una vez procesadas cada una de las necesidades del PMN se lanzan las órdenes de compra, las órdenes de fabricación y si se han producido alguna incidencia durante el proceso.

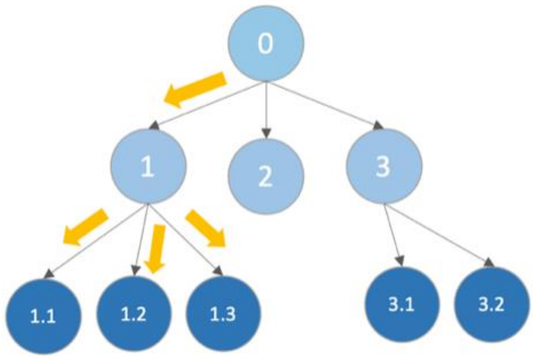

Figura 4. Ejemplo Árbol PMN. Fuente: Elaboración propia a partir del esquema proporcionado por GeInfor.

Lo que se muestra en la figura anterior, es un ejemplo de un Árbol PMN, y describe la manera de procesar las necesidades. En este ejemplo, la forma en la que se recorrería sería la siguiente: se comienza desde el nivel más alto (0) y se continúa por un nodo de un nivel inferior (1), después se recorre el siguiente nodo del nivel inferior a este y así sucesivamente hasta llegar al último nivel (1.1). Una vez recorrida la totalidad de la rama de árbol, se vuelve al nodo inmediatamente superior al último que se ha recorrido y que aún no haya sido procesado (1.2) y se recorre de la misma manera hasta llegar al último nivel. Así hasta recorrer cada uno de los nodos. Por lo que, la secuencia completa a seguir sería (0) - (1) - (1.1) - (1.2) - (1.3) - (2) - (3) - (3.1) - (3.2). Es decir, el algoritmo recorre de manera recursiva los nodos y los procesa uno a uno. Añadir, que en el caso en el que uno de los nodos no pueda satisfacer la necesidad, se dejan de recorrer los nodos inferiores unidos a este y cancela todos los nodos superiores que dependan del mismo. Esto quiere decir que, si alguno de los semielaborados o materias primas no está disponible, el producto final se cancela en su totalidad, y este al formar parte de un pedido, llevaría también a la cancelación del mismo.

Una vez visualizada la estructura general de algoritmo MRPc, se ahonda en cómo se calcula si es posible satisfacer o no una necesidad y cuándo deben lanzarse las órdenes. Para ello, se provee de una descripción numerada de los pasos que sigue el algoritmo y un diagrama de los mismos.

Descripción del algoritmo:

Paso 1. Se comprueba si todo el árbol ya ha sido planificado.

Paso 1.1. Si todo el árbol está planificado, se procede al lanzamiento de la compra de material y de producción. En caso contrario, paso 1.2.

Paso 1.2. Se selecciona el nivel más alto que no se haya planificado aún. Continúa en el paso 2.

Paso 2. Se verifica si hay stock disponible en el periodo indicado, para el producto de dicho nivel.

Paso 2.1. En caso afirmativo, y se disponga de stock, se procede a la reserva de material. Se vuelve al paso 1. En caso contrario, paso 2.2.

Paso 2.2. Si no se dispone de stock del artículo, se estudia si hay stock disponible del artículo de un nivel inferior. Continúa en el paso 3.

Paso 3. Se verifica si hay stock del artículo del nivel inferior siguiente.

Paso 3.1. Si existe dicho stock, se comprueba si hay capacidad para fabricar el artículo del nivel superior anterior. En caso contrario paso 3.2.

Paso 3.1.1 En el caso en el que, sí exista la capacidad necesaria, se reserva dicha capacidad. Y se procede a lanzar las órdenes de producción y las órdenes de compra según las reservas de capacidad y material que se hayan hecho. En caso contrario, paso 3.1.2.

Paso 3.1.2 Si no hay capacidad para poder fabricar el artículo, se busca capacidad en un periodo anterior.

Paso 3.1.2.1. En cuanto se encuentre un periodo anterior en el que haya capacidad para fabricar, se regresa al paso 3.1.1 en caso contrario, paso 3.1.2.2.

Paso 3.1.2.2. Si no existe ningún periodo anterior con capacidad de fabricación, se genera la incidencia de pedido y se cancelan todas las reservas y lanzamientos del mismo.

Paso 3.2. Si no hay stock disponible para el artículo, se comprueba el stock del siguiente inferior, hasta dar con algún artículo con stock o se llegue al final del árbol.

Paso 3.2.1. Si encuentra stock en uno de los artículos, se vuelve al paso 3.1. En caso contrario paso 3.2.2.

Paso 3.2.2. Si no se encuentra ningún artículo con stock, una vez recorrido todos los productos inferiores, se genera la incidencia de pedido y se cancelan todas las reservas y lanzamientos del pedido, que se relacionen con este producto.

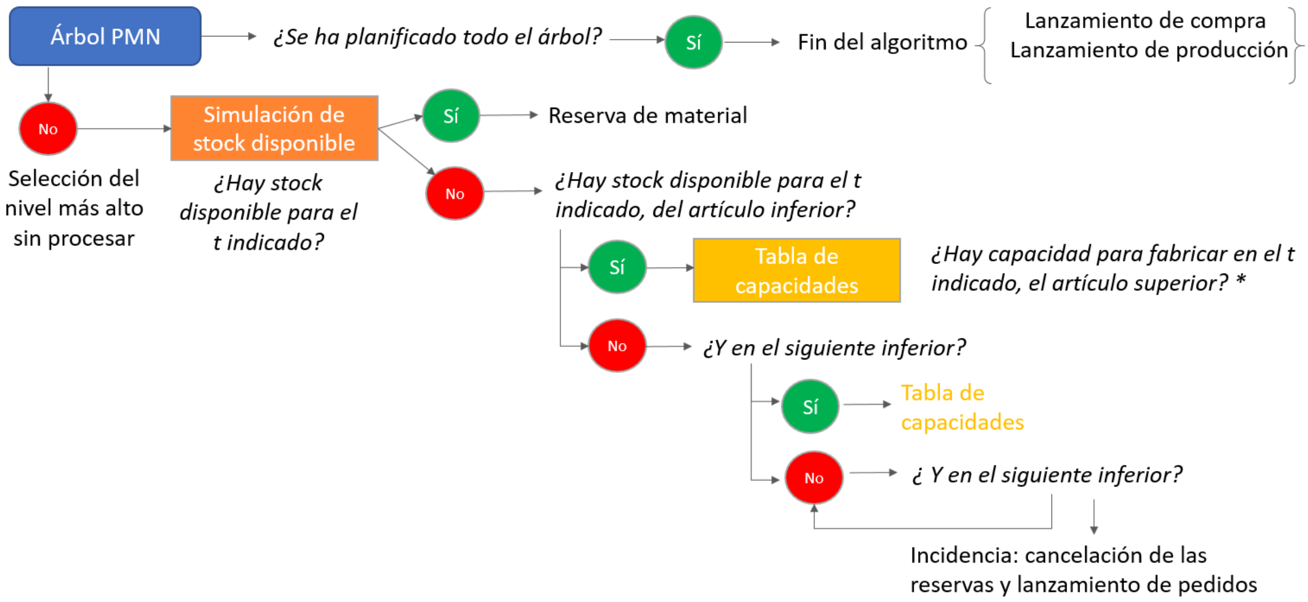

Figura 5. Esquema detallado del algoritmo MRPc. Fuente: Elaboración propia a partir de diagramas proporcionados por GeInfor.

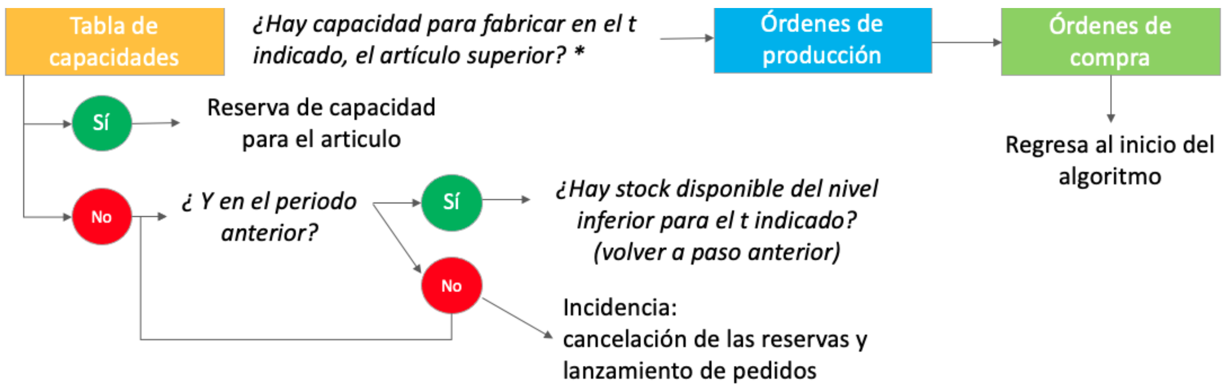

Figura 6. Continuación del esquema detallado MRPc. Fuente: Elaboración propia a partir de diagramas proporcionados por GeInfor.

Resumiendo, la idea principal del algoritmo es comprobar si se puede cumplir cada una de las necesidades de un pedido y de los productos que engloba, y en caso afirmativo hacer las reservas de material y capacidades pertinentes. Partiendo del nivel más alto que no haya sido comprobado, se comprueba si hay stock. Si no hay stock de un producto se verifica si hay stock del artículo del nivel inferior y se comprueba si hay capacidad para fabricar y lanzar las ordenes de fabricación y compra. Si no hay stock de un artículo de un nivel inferior, se comprueba el siguiente inferior y se hace de nuevo la misma comprobación de capacidad. En el caso de que no haya capacidad en el periodo para el que se demanda, se busca la capacidad en un periodo anterior. Tanto si no existe stock de ninguno de los artículos inferiores, cómo si no existe ningún periodo anterior a la fecha de entrega de la demanda con capacidad para fabricar, se deben cancelar todas las órdenes relacionadas con ese pedido.

2.3 Descripción del problema

Una vez descrito el funcionamiento del algoritmo MRPc, se plantea el problema a resolver. Dado que se pretende crear un modelo matemático que desempeñe la misma función que el MRPc, las especificaciones del algoritmo que se detallan a continuación serán los requisitos que deberá cumplir el modelo.

<u>Especificaciones del algoritmo y del modelo:</u>

- Se pretende planificar la producción a lo largo de un horizonte temporal, por lo que se trata de un problema multiperiodo.
- Existen restricciones de capacidad, por tanto, se habla de un problema de capacidad finita.
- Existe más de un tipo de producto, también denominado problema multiproducto.
- Se desagregan las demandas a pedidos. Es decir, que un pedido está compuesto por un conjunto de demandas. Y estas, indican qué cantidad de producto se quiere entregar y en qué fecha. Por lo que es posible que un mismo producto pueda darse en varios pedidos. Llámese pedido o lote.
- La técnica de lotificación empleada es de lote a lote, es decir que las cantidades a producir son las justas para satisfacer la demanda de manera exacta en cada uno de los periodos.
- En el caso en el que no se pueda satisfacer un producto correspondiente a un pedido, se deberá cancelar el propio pedido incluyendo todas las reservas de capacidad y stock que se hubiesen realizado asociadas a satisfacer dicho pedido.
- Es posible dar prioridad a un pedido, para dar mayor importancia a satisfacer un pedido frente a otros.
- Existe una relación entre las demandas de los distintos niveles de productos, por lo que hay una demanda dependiente a satisfacer. Se conoce como un problema multinivel con materias, primas, semielaborados y productos finales.
- No se tiene en cuenta stock de seguridad, aunque, como se verá es trivial considerarlo.
- No se da la posibilidad de hacer *splitting.* Entiéndase este término como la imposibilidad del fraccionamiento de entrega de una demanda y por tanto se obliga a que una demanda deba entregarse en un único periodo (y no en varios).
- La salida del algoritmo al igual que la solución del modelo matemático, determinará las cantidades y las fechas de las órdenes de compra y de fabricación.
- Se pretende maximizar el número de pedidos entregados.
- Se busca entregar las demandas lo más cerca posible a la fecha de entrega. Es decir, es posible adelantar las entregas, si no hubiera capacidad en el periodo de entrega.

Se concluye este capítulo habiendo visto la descripción del MRP y el funcionamiento del algoritmo MRPc. Así como los requisitos necesarios para realizar la formulación del modelo correspondiente en el capítulo 4. Estos requisitos serán incorporados en los modelos de forma escalonada, hasta dar con un modelo definitivo que cumpla con todos ellos.

3 MODELIZACIÓN MATEMÁTICA

Se inicia este capítulo haciendo una breve descripción de los pasos a seguir a la hora de resolver un problema mediante la modelización matemática. Y se introducirán algunas nociones acerca de la programación matemática, centrándose en la programación lineal y la programación entera mixta. En el último apartado, se conocerán las herramientas de optimización que se emplearán para resolver los modelos planteados.

3.1 Formulación generalizada de modelos

Formular un modelo consiste en representar un problema real, mediante expresiones matemáticas que caractericen de manera precisa la situación a modelizar. La modelización en muchos casos puede ser compleja y es necesario que previamente se entienda cual es el problema que se pretende resolver (2.3 Descripción del problema).

Las fases para llevar a cabo la formulación de un modelo son las siguientes, como indica Maroto et al. (2003):

- Paso 1: Identificación de las variables de decisión.
- Paso 2: Identificación de la función objetivo.
- Paso 3: Identificación de las restricciones.

<u>Paso 1: Identificación de las variables de decisión</u>

La identificación de las variables de decisión es el paso inicial para la construcción del modelo matemático. Las variables decisión se identifican como los elementos sobre los que se tiene control y sobre los que se toma una decisión. La correcta definición de estas es indispensable para poder proseguir con el modelado.

<u>Paso 2: Identificación de la función objetivo</u>

Como bien indica su nombre, la función objetivo (F.O.) es una expresión matemática que representa el objetivo del problema que se pretende resolver. Esta función depende de los valores que tomen las variables decisión y según del caso, tratará de maximizar o minimizar el valor objetivo.

<u>Paso 3: Identificación de las restricciones</u>

Las restricciones son expresiones matemáticas cuya función es limitar los valores que pueden tomar las variables decisión. Estas limitaciones pueden ser de tipo tecnológicas, económicas o incluso legales. En definitiva, la solución óptima deberá cumplir con todas condiciones restrictivas del problema.

Estas tres fases son las que se han aplicado a la hora de formular cada uno de los modelos que se proponen en el capítulo 4. Por tanto, un modelo matemático consta siempre de estos tres elementos principales: variables de decisión, función objetivo y restricciones. A estos elementos se le deben indicar las unidades que presentan. Cabe añadir un cuarto elemento: los parámetros, que son valores numéricos conocidos de antemano y que sirven para relacionar los tres elementos anteriores mencionados.

Para terminar este apartado, se muestra la forma genérica para la representación de cualquier modelo matemático. Incluye los 4 elementos que se han comentado en el párrafo anterior: parámetros, variables de decisión, función objetivo y restricciones. Por lo que respecta a la notación, generalmente, las variables de decisión se indican con letras mayúsculas mientras que los parámetros se escriben con letra minúscula.

Parámetros:

i: que va de 1 a m, donde m es el número de restricciones existentes.

j: que va de 1 a n, donde n es el número de variables existentes.

b_i: Disponibilidad del recurso. Puede ser una limitación o requerimiento.

$a_{i,j}$: Coeficientes técnicos.

C_j: Coeficientes de las variables en la función objetivo.

Variables decisión:

X_j: Variables de decisión.

Función Objetivo:

Z: Valor de la función objetivo.

$$Máx\ o\ Mín. \quad Z = \sum_{j=1}^{n} C_j \cdot X_j$$

Restricciones:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} \cdot X_{i,j} \leq b_i \text{ ó}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} \cdot X_{i,j} \geq b_i \text{ ó}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} \cdot X_{i,j} = b_i$$

3.2 Programación matemática

Una vez formulado un modelo matemático se necesita emplear técnicas de optimización, como la programación matemática para obtener una solución óptima del problema. Dentro de la programación matemática existen distintas técnicas que según el tipo de modelo que se pretende resolver, se emplean unas u otras.

Existen tres tipos de modelos, como afirman Lova y Tormos (2003) ordenados de menor a mayor según el coste computacional que conllevan, estos son:

1. Modelos lineales.
2. Modelos de programación lineal entera o mixta.
3. Modelos no lineales.

En los próximos apartados se explicarán las características básicas de los modelos lineales y los de programación lineal entera o mixta, puesto que son estos dos tipos de modelos los empleados en este TFG.

3.2.1 Programación lineal

Los modelos lineales son aquellos cuya función objetivo y restricciones son de tipo lineal[1] y dónde las variables de decisión son continuas[2]. Las características básicas o también denominadas hipótesis de la programación lineal son:

- Divisibilidad: $X_i \in \mathbb{R}$, las variables de decisión pueden tomar cualquier valor real.
- No negatividad: $X_i \geq 0$, las variables de decisión deben tomar valores no negativos.
- Linealidad: las expresiones función objetivo y restricciones deben ser lineales.
- Certidumbre: los coeficientes C_j , b_i, $a_{i,j}$ toman un valor numérico conocido de antemano que además no cambia.

La programación lineal (*Linear Programming*, LP), es una herramienta matemática en continuo auge, que se emplea diariamente para resolver problemas de todos los ámbitos imaginables. Tal y como afirman Hillier & Lieberman (2001), se trata de uno de los avances más importantes del siglo XX.

Según indica Nash (1990), el método algebraico más eficiente para encontrar la solución óptima de los problemas de programación lineal es el conocido Método Simplex. El algoritmo o método "original" de Simplex, tuvo sus inicios en 1947 y fue creado por George Dantizg. Previamente, hubo otros matemáticos que también publicaron artículos sobre el tema de la LP. Pero fue G. Dantzig quien propuso el uso de modelos basados en desigualdades lineales para la planificación de las actividades de la empresa.

La idea general del algoritmo Simplex es trabajar de manera iterativa, variando el valor de las variables de decisión hasta dar con que el valor máximo o mínimo (según el caso) de la función

[1] Función polinómica de primer grado.
[2] Puede tomar un número infinito de valores dentro de un intervalo.

objetivo. Basándose en la resolución de ecuaciones lineales aplicando la técnica de Gauss-Jordan (Lova & Tormos, 2003).

3.2.2 Programación lineal entera pura o mixta

Por otro lado, los modelos de programación lineal entera pura o mixta son aquellos cuya función objetivo y restricciones son de tipo lineal. Pero dónde las variables de decisión son discretas[3]. Por tanto, de las hipótesis de la PL, se satisfacen todas a excepción de la primera, dada la condición de no divisibilidad de las variables.

La diferencia entre la programación entera (*Integer Programming*, IP) y la programación entera mixta (*Mixed Integer* Programming), es que en la primera únicamente se trabaja con variables enteras, mientras que en la segunda se trabaja tanto variables enteras como continuas.

Tanto para problemas de IP como de MIP, existen distintos algoritmos de optimización, el Algoritmo de Bifurcación y Acotación también denominado Algoritmo *Branch & Bound* (B&B) es el más habitual. El algoritmo B&B es una técnica de optimización que se basa en la idea de obtener la mejor combinación de valores de las variables, pero sin tener que explorar todo el conjunto de posibilidades.

Tal y como explican Land & Doig (1960), el algoritmo inicialmente resuelve el problema como si se tratase de un LP (es decir, se relaja la condición de enteras de las variables) y si obtiene una solución entera (sin haber impuesto dicha condición), termina el proceso, y esa es la solución requerida. En el caso en el que no encuentre una solución entera en el primer intento, el algoritmo realiza la operación de bifurcación o ramificación que consiste en escoger una de las variables que aún no sea entera y crear dos nodos descendientes, imponiendo que esta variable tome los dos valores enteros más próximos. Por ejemplo, sea la variable continua $Xs = a, b$, donde (a es la cifra de las unidades y b la parte decimal), los dos nodos descendientes son $Xs \geq a + 1$ y $X \leq a$. De cada uno de los nodos creados, se plantea un subproblema que es el problema PL inicial adicionando una de las restricciones anteriores en cada subproblema. De este modo, al resolverlo de nuevo, se impondrá que la variable Xs tome un valor entero. Los criterios empleados para no continuar explorando una rama son: que se obtenga una solución con carácter entero, que no exista solución para ese subproblema o que se obtenga una solución continua, con un valor de la función objetivo peor.

Este proceso se realiza de manera iterativa y finaliza la búsqueda de solución cuando no quedan partes del árbol que explorar. De este modo, se consigue un conjunto de posibles soluciones que cumplan con la condición de enteras y de entre estas soluciones candidatas, la solución óptima será aquella que presente el mejor valor de la función objetivo. El algoritmo B&B tiene como principal desventaja que es exponencial en su tiempo de ejecución como función del número de variables enteras y/o binarias y para grandes tamaños de problema, no es factible su utilización práctica.

[3] Puede tomar un número finito de valores dentro de un intervalo, valores de tipo entero.

3.3 Herramientas de modelización

En este apartado se verán las dos herramientas de modelización que se utilizan para desarrollar este TFG. Las herramientas de modelización se emplean para calcular la solución del problema una vez ya se ha estudiado su formulación.

3.3.1 Microsoft Excel

Excel es una herramienta sencilla que hoy en día se encuentra integrada prácticamente en todas las empresas. Desde sus primeras versiones, Excel incluye de manera gratuita el optimizador Solver, que realmente es una versión gratuita del Solver ofrecido por la Empresa Frontline Systems (https://www.solver.com/). Este Solver emplea el algoritmo Simplex para problemas LP y para problemas MIP emplea el algoritmo B&B.

La ventaja principal que presenta el Solver de Excel es que implementar modelos simples en una hoja de cálculo es muy cómodo y sencillo, y de manera rápida se obtienen soluciones al problema reflejadas de forma muy visual y fácilmente interpretables.

Para facilitar la interpretación y lectura de la hoja de cálculo se siguen unas convenciones de estilo para identificar los distintos elementos del LP. La convención de colores que se muestra a continuación es la empleada en las hojas de Excel en el capítulo 5.

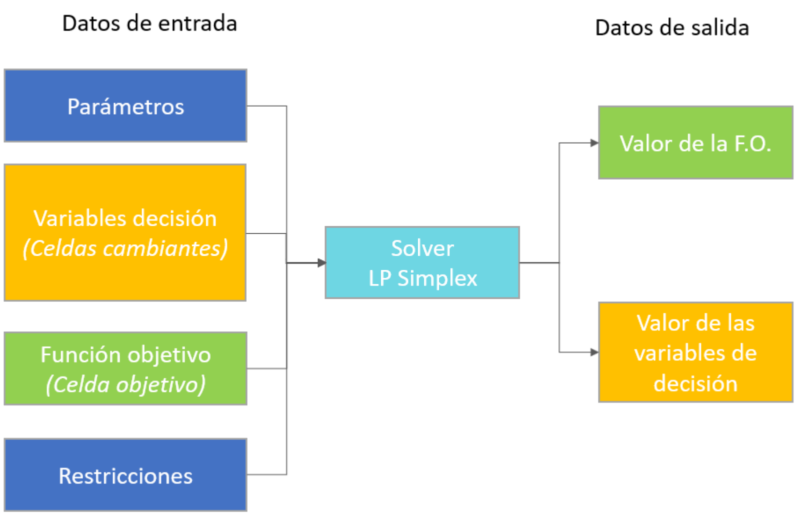

Figura 7. Elementos en una hoja de cálculo en un LP. Fuente: Elaboración propia a partir de Lova & Tormos (2003).

A pesar de que las prestaciones de Excel van mejorando con las sucesivas versiones, este software gratuito es limitado para modelos de gran dimensión. La dimensión de un modelo viene dada por el número de variables de decisión (enteras y binarias) y restricciones. A partir de un número considerable e incluso dependiendo de la construcción de la función objetivo, la implementación de un modelo en una hoja de cálculo puede resultar ser muy costosa. Además, llega un momento que la propia flexibilidad de las hojas de cálculo va en contra de la escalabilidad del modelo con lo que es fácil cometer errores y muy difícil encontrarlos.

3.3.2 IBM ILOG CPLEX

Como se ha comentado en el subapartado anterior, a partir de un cierto volumen de datos, restricciones y variables, emplear un Solver profesional es mucho más eficiente. Hablando tanto de tiempo computacional como del tiempo necesario para implementar el propio modelo en el programa.

IBM ILOG CPLEX Optimizer (https://www.ibm.com/es-es/products/ilog-cplex-optimization-studio) es un software de optimización de pago que con potentes algoritmos es capaz de resolver problemas matemáticos con millones de restricciones y variables (a diferencia de Excel que está limitado a 200 variables y 100 restricciones en su versión gratuita). Además, CPLEX no solo ofrece los valores que toman de las variables de decisión y la función objetivo y holgura de las restricciones, sino también información relacionada con el motor de búsqueda. Por ejemplo, el número de iteraciones que ha realizado, el tiempo computacional o el GAP[4] de la solución. Incluso presenta funcionalidades como establecer un tiempo límite computacional, rango de tolerancia aceptable y un % de GAP máximo. Aunque algunas de estas funciones también están soportadas por el Solver de EXCEL, es en CPLEX donde se permite un mayor control y capacidad.

En resumen, se trata de un Solver Profesional estado del arte, dirigido a la resolución de problemas matemáticos, capaz de resolver problemas reales de grandes dimensiones.

[4] Diferencia entre la mejor solución entera conocida y la solución calculada con la relajación de restricciones de integralidad. Una solución óptima es aquella que tiene un GAP de 0%.

4 PROPUESTA DEL MODELO

En este cuarto capítulo se estudia la evolución de los modelos, desde el modelo estándar de la planificación de la producción hasta concluir con la propuesta del modelo final. El objetivo es formular un modelo que aúne las especificaciones y funcionalidad del algoritmo MRP de GeInfor. Aunque las especificaciones del algoritmo se hayan visto descritas en el apartado 2.3 Descripción del problema, se provee de un cuadro resumen, Tabla 1. Características del problema a incorporar en el modelo con las características a incorporar.

- Problema multiperiodo.
- Problema de capacidad finita.
- Problema multiproducto.
- Problema multinivel.
- Desagregado de demandas a pedidos.
- Técnica de lotificación de lote a lote.
- Cancelación completa de un pedido en el caso en el que no se puedan satisfacer todas las demandas del mismo.
- Posibilidad a dar prioridad a un pedido.
- Sin stock de seguridad.
- Sin división de lotes y pedidos (*splitting*).
- Cálculo de las cantidades y las fechas de las órdenes de compra y de fabricación.
- Maximizar el número de pedidos entregados y servir las demandas lo más cerca posible a la fecha de entrega.

Tabla 1. Características del problema a incorporar en el modelo.

Los modelos se han desarrollado de manera secuencial, es decir, que un modelo se formulará partiendo de la base del modelo anterior. La adición o cambio de variables, restricciones o función objetivo, vendrá dada según las características que se vayan añadiendo. En la Figura 8, se muestra un esquema con los pasos seguidos hasta llegar a la formulación del último modelo. De esta manera se puede ir observando cómo se van incorporando las características del algoritmo en los modelos.

La estructura que presenta cada uno de los modelos es la siguiente:

- Descripción de los objetivos a conseguir del modelo.
- Especificaciones del algoritmo MRPc que se incorporan.
- Índices, parámetros, variables de decisión, función objetivo y restricciones.
- Explicación de la construcción del modelo.
- Diferencias respecto del modelo anterior.

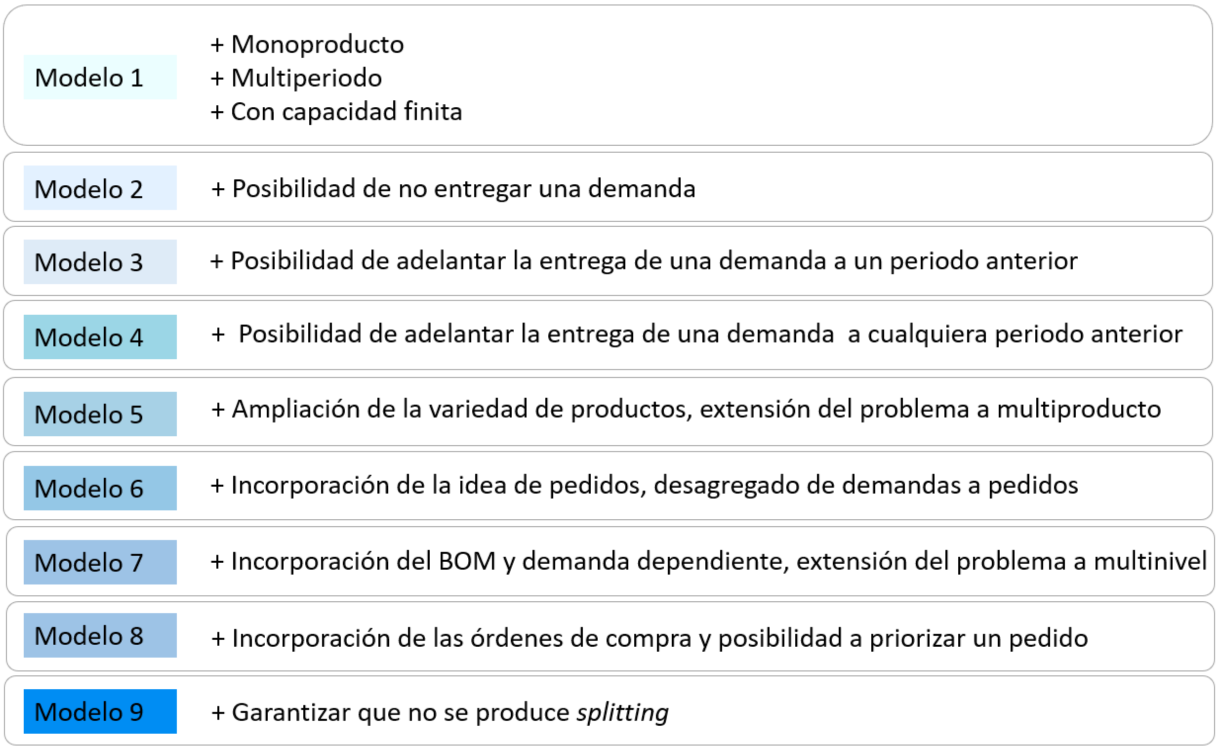

Figura 8. Esquema sobre las propuestas de modelos. Fuente: Elaboración propia.

4.1 Modelo 1. Modelo estándar de la planificación de la producción

En este modelo de la planificación de la producción, se parte del modelo más simple posible, conocido como modelo estándar de la producción, según Pochet & Wolsey (2006). Se trata de un modelo monoproducto ya que únicamente tiene en consideración un tipo de producto. Además, es multiperiodo dado que existe un horizonte de planificación temporal y con capacidad finita dado que la solución se verá restringida por las capacidades del sistema. Es importante incidir en que este modelo ya considera capacidades, con lo que es una mejora sustancial con respecto a los algoritmos clásicos MRP.

El objetivo de este modelo es minimizar los costes debidos a la fabricación de productos y al almacenamiento de stock.

▪ Monoproducto ▪ Multiperiodo ▪ Capacidad finita

Tabla 2. Características del modelo 1.

Índices

t: Periodos que va 1 a n.

Parámetros

c_t: Capacidad máxima de fabricación del producto en el periodo t, (unidades producidas por periodo) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento del producto en el periodo t, (unidades almacenadas por periodo) t = 1, …, n.

p_t: Coste de fabricar una unidad del producto en el periodo t, (unidades monetarias por unidad de producto) t = 1, …, n.

h_t: Coste de almacenar una unidad del producto en el periodo t, (unidades monetarias por unidad de producto) t = 1, …, n.

d_t: Demanda del producto en el periodo t, (unidades por periodo) t = 1, …, n.

s_0: Stock inicial de producto (unidades).

Variables

X_t: Unidades del producto fabricadas en el periodo t, (unidades por periodo) t = 1, …, n.

S_t: Unidades del producto almacenadas en el periodo t (unidades por periodo) t = 1, …, n.

Función Objetivo

Minimizar el coste total de fabricación y almacenamiento.

$$\text{mín } Z = \sum_{\mathrm{t=1}}^{\mathrm{n}} (p_t \cdot X_t + h_t \cdot S_t) \tag{1.1}$$

Restricciones

[Capacidad de producción]

$$X_t \leq c_t \quad t = 1, \dots, n \tag{1.2}$$

[Capacidad de almacenamiento]

$$S_t \leq a_t \quad t = 1, \dots, n \tag{1.3}$$

[Restricción de equilibrio]

$$S_{t-1} + X_t = S_t + d_t \quad t = 1, \dots, n \tag{1.4}$$

[Condición no negatividad]

$$X_t, S_t \geq 0 \quad t = 1, \ldots, n \tag{1.5}$$

[Variables enteras]

$$X_t, S_t \; enteras \quad t = 1, \ldots, n \tag{1.6}$$

Tal y como se ha mencionado, la función objetivo de este modelo es obtener los valores de las unidades de producto fabricadas y unidades de stock para cada periodo que minimice los costes asociados a la fabricación y al almacenamiento. El coste total se expresa como la suma del coste asociado a la fabricación por las unidades fabricadas y el coste de almacenamiento por las unidades de stock, a lo largo de todos los periodos, ecuación (1.1).

Respecto a las restricciones, las restricciones de capacidad y almacenamiento garantizan que las unidades producidas y las unidades de stock no rebasen en ningún momento la capacidad máxima asociada a fabricación y almacenamiento, respectivamente (1.2) y (1.3).

Sin embargo, la restricción fundamental característica de la planificación de la producción es la restricción de equilibrio (1.4). Esta ecuación garantiza que se produzca la conservación de flujo, "la cantidad que entra de producto" debe ser igual a "la cantidad que sale de producto". Entiéndase como entradas o *inputs,* a la cantidad de producto almacenado del periodo anterior y las nuevas unidades producidas, mientras que las salidas u *outputs,* a la demanda a satisfacer y las unidades que se almacenen en este periodo puesto que no son necesarias para cubrir la demanda. Obsérvese la figura siguiente.

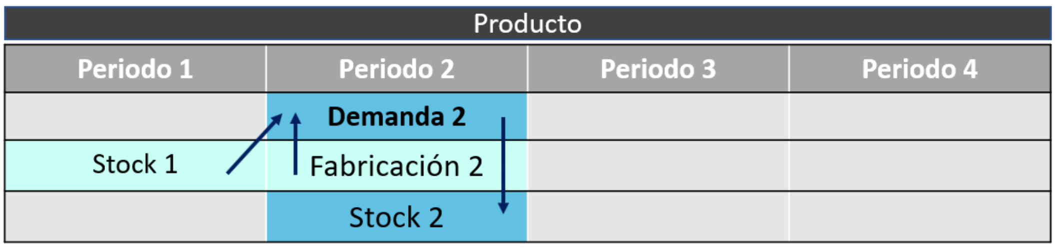

Figura 9. Ejemplo de la restricción de equilibrio. Fuente: Elaboración propia.

En este ejemplo, para el periodo t=2, los *inputs* son el almacenamiento del periodo anterior ("stock 1") más la fabricación de nuevas unidades ("fabricación 2") y los *outputs* son la demanda por cubrir en ese periodo ("demanda 2") y las unidades restantes se almacenan para satisfacer la demanda del siguiente periodo ("stock 2").

Por último, hay que garantizar la condición de no negatividad de las variables decisión (1.5) y de enteras, dado que se habla de unidades de producto (1.6).

4.2 Modelo 2. Modelo con posibilidad de no satisfacer una demanda

Este segundo modelo, parte del modelo estándar de producción multiperiodo anterior. La diferencia respecto de éste es que se añade la posibilidad de no satisfacer una demanda. Si se retrocede a la restricción de equilibrio (1.4) del modelo anterior, se puede observar que la demanda (d_t) es una constante. Esto quiere decir que d_t tomará un valor fijo en la restricción y que por tanto para poder obtener una solución del problema, tendrá que satisfacer siempre la demanda dada la restricción de igualdad. Esto es un gran inconveniente, puesto que, si se da el caso en el que no se puede satisfacer una demanda de un periodo cualquiera, por ejemplo, debido a las restricciones de capacidad, no se obtendría solución al problema (solución no factible[5]). Por tanto, la ventaja de este modelo es que se da la posibilidad de satisfacer o no una demanda. De esta manera, a pesar de que no se puedan satisfacer todas las demandas, si se pueda obtener una solución.

Este cambio, que puede parecer trivial, no lo es tanto. Por ejemplo, si se permite no satisfacer demandas y al mismo tiempo se mantiene el objetivo de minimización de costes, el Solver proporcionará una solución óptima trivial de coste cero donde no se satisface ninguna demanda. Por lo tanto, es necesario cambiar el objetivo, que ahora deber incluir la maximización del número de demandas. Se dice que una demanda es satisfecha o servida, cuando se consigue entregar la cantidad de producto que se requiere.

▪ Monoproducto ▪ Multiperiodo ▪ Capacidad finita ▪ **Con posibilidad de no servir una demanda**

Tabla 3. Características del modelo 2.

Índices

t: Periodos que va de 1 a n.

Parámetros

c_t: Capacidad máxima de fabricación del producto en el periodo t, (unidades producidas por periodo) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento del producto en el periodo t, (unidades almacenadas por periodo) t = 1, …, n.

[5] No existe ninguna solución que cumpla todas las restricciones impuestas en el problema a la vez. En ese caso, el Solver proporciona una solución no factible, es decir, da unos valores a las variables de decisión, pero hay una o varias restricciones que no se cumplen.

d_t: Demanda del producto en el periodo t, (unidades por periodo) t = 1, …, n.

s_0: Stock inicial del producto (unidades).

Variables

X_t: Unidades del producto fabricadas en el periodo t, (unidades por periodo) t = 1, …, n.

S_t: Unidades del producto almacenadas en el periodo t, (unidades por periodo) t = 1, …, n.

YE_t *(0,1)*: Variable binaria, 1 si se entrega la demanda del periodo t, 0 en caso contrario, t = 1, …, n.

Función Objetivo

Maximizar el número demandas satisfechas.

$$\text{máx } Z = \sum_{t=1}^{n} YE_t \tag{2.1}$$

Restricciones

[Capacidad de producción]

$$X_t \leq c_t \quad t = 1, \ldots, n \tag{2.2}$$

[Capacidad de almacenamiento]

$$S_t \leq a_t \quad t = 1, \ldots, n \tag{2.3}$$

[Uso de la variable binaria YE_t]

$$X_t \leq d_t \cdot YE_t \quad t = 1, \ldots, n \tag{2.4}$$

$$YE_t \leq d_t \quad t = 1, \ldots, n \tag{2.5}$$

[Restricción de equilibrio]

$$S_{t-1} + X_t = S_t + d_t \cdot YE_t \qquad t = 1, \ldots, n \tag{2.6}$$

[Condición no negatividad]

$$X_t, S_t \geq 0 \quad t = 1, \ldots, n \tag{2.7}$$

[Variable binaria]

$$YE_t \ binarias \quad t = 1, \ldots, n \tag{2.8}$$

[Variables enteras]

$$X_t, S_t \ \ enteras \ \ t = 1, \dots, n \qquad (2.9)$$

En primer lugar, para poder decidir si se satisface o no la demanda correspondiente de un periodo, se debe de introducir una variable binaria que permita controlar el valor que pueda tomar la demanda. A esta variable se le ha denominado YE_t y habrá una asociada por cada una de las demandas d_t. Ya que es necesario indicar para cada uno de los periodos si se satisface o no la demanda de ese periodo.

Según la definición de la variable YE_t, tomará valor 1 si se entrega la demanda del periodo t (se entrega la cantidad de producto exigida por demanda) y 0 en caso contrario, se trata de una variable binaria, ecuación (2.8). Para garantizar estas condiciones se emplea la ecuación (2.4). Dónde se expresa que la cantidad de producto producida debe de ser menor o igual a la demanda multiplicada por su variable binaria. De esta manera se impone que si $YE_t = 1$, la cantidad a producir será como máximo la demanda y en caso de que $YE_t = 0$, se obliga a que no se produzca puesto que la demanda no se va a satisfacer. Además, se puede observar que para el caso en el que la demanda sea 0, es necesario limitar el valor de la binaria. Dado que si $d_t = 0$, no se debe considerar como una demanda a satisfacer y se impone que su YE_t a la que está asociada, tome de manera inmediata $YE_t = 0$ ecuación (2.5).

Por otro lado, en la ecuación (2.6) la restricción de equilibrio se observa que el término de la derecha de la igualdad se ha modificado. Se tiene la demanda multiplicada por su binaria de entrega asociada, de este modo el término asociado a la demanda es libre de tomar valor 0 (si no se satisface, $YE_t = 0$) o el propio valor de la d_t (si se satisface, $YE_t = 1$).

Por último, la función objetivo del modelo busca maximizar el número de demandas satisfechas, se escribe como la suma de todas las demandas del horizonte de planificación. Sabiendo lo que representa la variable binaria YE_t, la F.O. (2.1) trata de ponerlas todas a valor 1.

4.3 Modelo 3. Modelo con posibilidad de adelantar la entrega de una demanda un solo periodo

En este tercer modelo se quiere añadir la posibilidad de que se pueda adelantar la entrega del producto, pero con un adelanto máximo de un periodo. Permite entregar la demanda correspondiente a un periodo en su mismo periodo o en su periodo anterior. Además, se permite entregar la demanda en dos tiempos, por lo que puede servir tanto en el periodo anterior como en el mismo periodo (división de la demanda o *splitting*).

El objetivo de este modelo es maximizar el número de demandas satisfechas y el número de entregas efectivas realizadas en la fecha de entrega de la demanda, frente al número de entregas efectuadas con un periodo de antelación.

Para ello, es necesario crear una nueva variable que indique la cantidad de producto que se sirve en un periodo y a qué demanda corresponde. Además, para determinar en qué periodos se efectúa una entrega será necesario adicionar una variable binaria que indique en qué periodos se ha efectuado una entrega, y correspondiente a qué demanda.

▪ Monoproducto ▪ Multiperiodo ▪ Capacidad finita ▪ Con posibilidad de no servir una demanda ▪ **Con posibilidad de adelantar un solo periodo la entrega de una demanda**

Tabla 4. Características del modelo 3.

Índices

t , t_1, t_2: Periodos que van de 1 a *n.*

Parámetros

c_t: Capacidad máxima de fabricación del producto en el periodo t, (unidades producidas por periodo) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento del producto en el periodo t, (unidades almacenadas por periodo) t = 1, …, n.

d_t: Demanda del producto en el periodo t, (unidades por periodo) t = 1, …, n.

s_0: Stock inicial del producto (unidades).

Variables

X_t Unidades del producto fabricadas en el periodo t, (unidades por periodo) t = 1, …, n.

S_t: Unidades del producto almacenadas en el periodo t, (unidades por periodo) t = 1, …, n.

E_{t_1,t_2}: Unidades de producto que se sirven en el periodo t_1, correspondientes a la demanda del periodo t_2, t_1, t_2=1, …, n, $t_1 \leq t_2$.

YE_t *(0,1)*: Variable binaria, 1 si se entrega la demanda del periodo t, 0 en caso contrario, t = 1, …, n.

W_{t_1,t_2} *(0,1)*: Variable binaria, 1 si se sirve producto en el periodo t_1 correspondiente a la demanda del periodo t_2, 0 en caso contrario, t_1, t_2=1, …, n, $t_1 \leq t_2$.

Función Objetivo

Maximizar el número total de demandas satisfechas y número de entregas efectivas en el mismo periodo que la demanda y minimizar el número de entregas efectuadas con un periodo de antelación.

$$\text{máx } Z = \sum_{t=1}^{n} YE_t + \sum_{t=1}^{n} W_{t,t} - \sum_{t=2}^{n} W_{t-1,t} \quad (3.1)$$

Restricciones

[Capacidad de producción]

$$X_t \leq c_t \quad t = 1, \dots, n \quad (3.2)$$

[Capacidad de almacenamiento]

$$S_t \leq a_t \quad t = 1, \dots, n \quad (3.3)$$

[Uso de la variable binaria YE_t]

$$E_{t-1,t} + E_{t,t} = d_t \cdot YE_t \quad t = 1, \dots, n \quad (3.4)$$

$$YE_t \leq d_t \quad t = 1, \dots, n \quad (3.5)$$

[Restricción de equilibrio]

$$S_{t-1} + X_t = S_t + E_{t,t} + E_{t,t+1} \quad t = 1, \dots, n \quad (3.6)$$

[Uso de la variable binaria W_{t_1,t_2}]

$$\varepsilon \cdot W_{t,t} \leq E_{t,t} \leq d_t \cdot W_{t,t} \quad t = 1, \dots, n \quad (3.7)$$

$$\varepsilon \cdot W_{t-1,t} \leq E_{t-1,t} \leq d_t \cdot W_{t-1,t} \quad t = 2, \dots, n \quad (3.8)$$

Siendo ε un número muy pequeño positivo.

[Condición de no negatividad]

$$X_t, S_t, E_{t_1,t_2} \geq 0 \quad t, t_1, t_2 = 1, \dots, n, \; t_1 \leq t_2 \quad (3.9)$$

[Variables binarias]

$$YE_t, E_{t_1,t_2} \; binarias \quad t, t_1, t_2 = 1, \dots, n, \; t_1 \leq t_2 \quad (3.10)$$

[Variables enteras]

$$X_t, S_t, E_{t_1,t_2} \; enteras \quad t, t_1, t_2 = 1, \dots, n, \; t_1 \leq t_2 \quad (3.11)$$

En primer lugar, se explica el uso de la ecuación (3.4), uso de la variable YE_t. Se puede observar que respecto al modelo anterior esta ecuación ha sufrido una modificación. Puesto que ahora se permite entregar parte o la demanda entera en un periodo anterior, la demanda ya no se limita con la cantidad de unidades que se fabrican en ese periodo. Sino con la cantidad de producto que se entrega en el periodo (t) $E_{t,t}$ y en el periodo anterior (t-1) $E_{t-1,t}$. Por lo que la ecuación indica que la cantidad entregada en un periodo anterior más la cantidad entregada en el periodo de la demanda, tiene que ser exactamente igual a la cantidad de producto demandada. Obsérvese la figura siguiente, dónde se expone un sencillo ejemplo para comprender la restricción.

Producto			
Periodo 1	**Periodo 2**	**Periodo 3**	**Periodo 4**
		Demanda 3	
	Entrega 2,3	Entrega 3,3	

Figura 10. Ejemplo de la restricción (3.4). Fuente: Elaboración propia.

En este ejemplo, se supone que existe una demanda de producto en el periodo 3 (“demanda 3”). Esta demanda con fecha de entrega en *t*=3, puede ser satisfecha mediante la entrega en el periodo anterior, periodo 2 (“entrega 2,3”) o con la entrega en el mismo periodo (“entrega 3,3”). La suma de las dos entregas debe de ser igual a la cantidad demanda en el periodo 3.

Con relación al uso de la variable E_{t_1,t_2}, también debe de modificarse la restricción de equilibrio (3.6). Dado que se permite el adelanto de entrega en un periodo, para un periodo dado, la cantidad que se produce y que había almacenada, tiene que ser igual a la cantidad que se sirve en dicho periodo para satisfacer tanto la demanda de ese periodo como la del periodo siguiente.

En el ejemplo de la Figura 11. Ejemplo restricción , para el periodo t=2, las unidades almacenadas en el periodo anterior (“stock 1”) más las unidades fabricadas en el periodo (“fabricación 2”) debe de ser igual a la cantidad que se entrega en el periodo para satisfacer la demanda de dicho periodo (“entrega 2,2”) y la que se entrega para satisfacer la demanda del periodo siguiente (“entrega 2,3”).

Producto			
Periodo 1	**Periodo 2**	**Periodo 3**	**Periodo 4**
	Demanda 2	**Demanda 3**	
Stock 1	Fabricación 2		
	Stock 2		
	Entrega 2,2		
	Entrega 2,3		

Figura 11. Ejemplo restricción (3.6). Fuente: Elaboración propia.

Respecto al uso de la variable binaria $W_{t1,t2}$, indica si en el periodo $t_1$se ha realizado una entrega correspondiente a la demanda del periodo t_2, toma el valor 1 en caso de afirmativo y 0 en caso contrario. Para ello se emplea la restricción (3.7) y (3.8) para indicar que si E_{t_1,t_2}es distinto de 0 (se hace una entrega de producto) entonces $W_{t1,t2}$ toma el valor 1.

En cuanto a las condiciones de variables binarias y enteras (3.9) y (3.10), se adicionan las nuevas variables $W_{t1,t2}$ con carácter binario y E_{t_1,t_2} con carácter entero.

Finalmente, la función objetivo parte de la F.O. anterior (2.1) pero añadiendo el término aditivo relativo a las entregas efectuadas en la fecha de entrega y un término substractivo relativo a las entregas efectuadas con un periodo de antelación. Dado el carácter de maximización, al incluir el signo menos delante del tercer término de la ecuación, tratará de hacer ese término 0, o lo que es lo mismo intentar que no se hagan entregas con un periodo de antelación. No obstante, siempre convendrá más hacer una entrega fraccionada en dos periodos que no realizar la entrega, dado que YE_t y $W_{t,t}$ están sumando en la función objetivo e interesará hacer esas variables positivas, que sumarán 2, aunque $W_{t-1,t}$ sume 1.

4.4 Modelo 4. Modelo con posibilidad de adelantar la entrega de una demanda en cualquier periodo anterior

A diferencia del modelo anterior, en este se da la posibilidad de servir una demanda con adelanto, en cualquier periodo anterior a su fecha de entrega. Por lo tanto, se trata de extender el problema anterior que permite un solo periodo de adelanto, hasta otro que permita adelantar la entrega un número indefinido de periodos.

El objetivo del problema es el mismo, trata de satisfacer el máximo número de demandas e intenta servir las demandas lo más próximo posible a su fecha de entrega.

- Monoproducto
- Multiperiodo
- Capacidad finita
- Con posibilidad de no servir una demanda
- **Con posibilidad de entregar una demanda con antelación en cualquier periodo anterior**

Tabla 5. Características del modelo 4

Índices

t , t_1, t_2: Periodos que van de 1 a n.

g: Periodos que va de 1 a n-1.

Parámetros

c_t: Capacidad máxima de fabricación del producto en el periodo t, (unidades producidas por periodo) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento del producto en el periodo t, (unidades almacenadas por periodo) t = 1, …, n.

d_t: Demanda del producto en el periodo t, (unidades por periodo) t = 1, …, n.

s_0: Stock inicial del producto (unidades).

b: Bonificación por satisfacer una demanda.

f_g: Penalización por servir una unidad de producto con adelanto en el periodo g, g= 1, …, n-1.

Variables

X_t Unidades del producto fabricadas en el periodo t, (unidades por periodo) t = 1, …, n.

S_t: Unidades del producto almacenadas en el periodo t, (unidades por periodo) t = 1, …, n.

E_{t_1,t_2}: Unidades de producto que se sirven en el periodo t_1, correspondientes a la demanda del periodo t_2, t_1, t_2=1, …, n, $t_1 \leq t_2$.

YE_t*(0,1)*: Variable binaria, 1 si se entrega la demanda del periodo t, 0 en caso contrario, t = 1, …, n.

W_{t_1,t_2}*(0,1)*: Variable binaria, 1 si se entrega producto en el periodo t_1 correspondiente a la demanda del periodo t_2, 0 en caso contrario, t_1, t_2=1, …, n, $t_1 \leq t_2$.

Función Objetivo

Maximizar el número total de demandas entregadas y el número de demandas que se sirven en la propia fecha de entrega. Y minimizar el número de entregas que se sirven con adelanto. Dentro de las entregas que se sirven con antelación, serán más perjudiciales aquellas que se alejan más de la fecha de entrega correspondiente.

$$\text{máx } Z = b\sum_{\mathrm{t}=1}^{\mathrm{n}} YE_t + \sum_{\mathrm{t}=1}^{\mathrm{n}} W_{t,t} - \sum_{k=1}^{t-1}\sum_{t=2}^{n} f_{t-k} \cdot W_{t-k,t} \tag{4.1}$$

Restricciones

[Capacidad de producción]

$$X_t \leq c_t \quad t = 1, \dots, n \tag{4.2}$$

[Capacidad de almacenamiento]

$$S_t \leq a_t \quad t = 1, \dots, n \tag{4.3}$$

[Uso de la variable binaria YE_t]

$$\sum_{k=1}^{t} E_{k,t} = d_t \cdot YE_t \qquad t = 1, \dots, n \tag{4.4}$$

$$YE_t \leq d_t \qquad t = 1, \dots, n \tag{4.5}$$

[Restricción de equilibrio]

$$S_{t-1} + X_t = S_t + \sum_{k=0}^{n-t} E_{t,t+k} \qquad t = 1, \dots, n \tag{4.6}$$

[Uso de la variable binaria $W_{t1,t2}$]

$$\varepsilon \cdot W_{t1,t2} \leq E_{t_1,t_2} \leq d_t \cdot W_{t_1,t_2} \qquad t, t_1, t_2 = 1, \dots, n,\ t_1 \leq t_2 \tag{4.7}$$

Siendo ε un número muy pequeño positivo.

[Condición de no negatividad]

$$X_t, S_t, E_{t_1,t_2} \geq 0 \quad t, t_1, t_2 = 1, \dots, n,\ t_1 \leq t_2 \tag{4.8}$$

[Variables binarias]

$$YE_t, W_{t_1,t_2}\ binarias \quad t, t_1, t_2 = 1, \dots, n,\ t_1 \leq t_2 \tag{4.9}$$

[Variables enteras]

$$X_t, S_t, E_{t_1,t_2}\ enteras \quad t, t_1, t_2 = 1, \dots, n,\ t_1 \leq t_2 \tag{4.10}$$

El uso de la variable binaria YE_t (4.4), tiene la misma función que la presentada en el modelo anterior. La diferencia se encuentra al lado izquierdo de la igualdad donde, en lugar de incluir únicamente las entregas del periodo de demanda y el periodo anterior, se incluyen todas las entregas realizadas desde el periodo de inicio $t = 1$ hasta la fecha de entrega máxima de la demanda.

Producto			
Periodo 1	**Periodo 2**	**Periodo 3**	**Periodo 4**
		Demanda 3	
Entrega 1,3	Entrega 2,3	Entrega 3,3	

Figura 12. Ejemplo restricción (4.4). Fuente: Elaboración propia.

Como se puede observar en la Figura 12, lo que se indica en esta ecuación es que todas las cantidades de producto que se entreguen desde el periodo inicial (“periodo 1”) hasta la fecha de entrega (“periodo 3”), deberá ser igual a la cantidad demandada (“demanda 3”).

Así pues, en la restricción de equilibrio (4.6) también debe modificarse el término relativo a las entregas, a la derecha de la igualdad. Puesto que el stock disponible y las unidades que se fabriquen podrán emplearse para satisfacer cualquier demanda (de los periodos siguientes). Véase Figura 13.

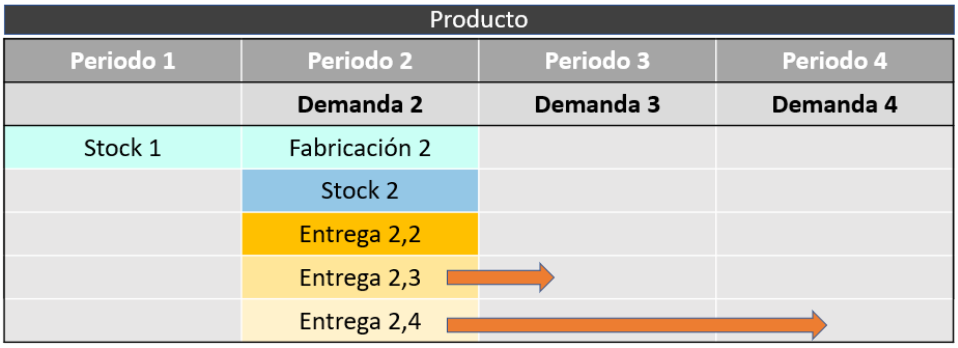

Figura 13. Ejemplo restricción (4.6). Fuente: Elaboración propia.

En este ejemplo para el $t = 2$, el stock disponible (“stock 1”) y las unidades que se fabrican (“fabricación 2”), debe de ser igual a la cantidad de producto que se entrega en el periodo destinado a satisfacer la propia demanda del periodo y cualquier demanda futura (“demanda 3” y “demanda 4”).

Por otro lado, la restricción (4.7) es la ampliación de las restricciones (3.7) y (3.8) a todos los periodos siendo $t_1 \leq t_2$, dado que es posible servir el pedido o parte del pedido, en cualquier periodo anterior a la fecha de entrega, pero no se permite en ningún caso atrasar la entrega de demanda.

En último lugar, queda explicar la F.O. (4.1). Como se menciona se busca entregar el máximo número de demandas posibles. Y tratando de entregar en la propia fecha de entrega. Para ello, se añade la constante b (un número grande positivo) y dado que el objetivo es maximizar, hará que las YE_t tomen valor 1. Mientras que, dado el signo negativo delante del tercer término, se va a minimizar el número de entregas que se sirven con adelanto. Pero, dentro de las entregas que se sirven con antelación, deben ser más negativas aquellas que se alejen más de la fecha de entrega. Para ello se incluye la constante f_{t-k}, que tomará valores más grandes cuanto más se aleje de la fecha de entrega (mayor valor de k) y más pequeños cuanto más próximo este (menor valor de k). Aunque, en comparación, el valor de la bonificación es más grande que el de las penalizaciones, ya que es más favorable entregar una demanda (aunque sea con antelación) que no entregarla. Estos valores los fija el usuario decisor.

El problema existente en este modelo es que, dado que en la F.O. no aparece ningún término relativo a los costes que pueda llevar asociados una entrega, el modelo siempre a va a entregar una demanda, independientemente de los costes que pueda conllevar. Sin embargo, en la realidad, el hecho de entregar una demanda en la fecha de entrega deseada puede llevar costes de almacenamiento o de producción que en algunos no compense entregarla en dicha fecha, y sea mejor adelantarla. Esto afecta a este último modelo, pero también a los anterior. Este problema se aborda en el siguiente modelo.

4.5 Modelo 5. Modelo multiproducto

Como se ha comentado en el modelo anterior, el problema reside en determinar hasta qué punto es mejor entregar una demanda en la fecha de entrega, con todos los costes que puede conllevar, o adelantar la fecha de entrega efectiva acarreando un efecto negativo por no cumplir con la fecha de entrega marcada por el cliente. Para ello, será necesario transformar la F.O. anterior en unidades monetarias, de esta manera, tendrá en consideración los costes derivados por satisfacer una demanda.

Además, de resolver dicho problema, se extiende a ser un modelo multiproducto.

▪ Monoproducto ▪ Multiperiodo ▪ Capacidad finita ▪ Con posibilidad de no servir una demanda ▪ Con posibilidad de entregar una demanda con antelación en cualquier periodo anterior ▪ **Multiproducto**

Tabla 6. Características del modelo 5.

Índices

t, t_1, t_2: Periodos que van de 1 a n.

g: Periodos que va de 1 a n-1.

i: Productos que va de 1 a m.

Parámetros

c_t: Capacidad máxima de fabricación en el periodo t, (unidades producidas por periodo) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento en el periodo t, (unidades almacenadas por periodo) t = 1, …, n.

d_t^i: Demanda del producto i en el periodo t, (unidades por periodo) i = 1, …, m, t = 1, …, n.

S_0^i: Stock inicial del producto i (unidades) i = 1, …, m.

p_t^i: Coste de fabricar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

h_t^i: Coste de almacenar una unidad del producto *i* en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

q_t^i: Coste fijo de producción del producto i en el periodo t, (unidades monetarias) i = 1, …, m,

t = 1, …, n.

b: Bonificación económica por la entrega de una demanda, (unidades monetarias por demanda).

f_g: Penalización por servir una unidad de producto con adelanto en el periodo g, g= 1, …, n-1.

Variables

X_t^i: Unidades del producto *i* fabricadas en el periodo t, (unidades por periodo) i = 1, …, m,

t = 1, …, n.

S_t^i: Unidades del producto i almacenadas en el periodo t, (unidades por periodo) i = 1, …, m,

t = 1, …, n.

E_{t_1,t_2}^i: Unidades de producto i que se sirven en el periodo t_1, correspondientes a la demanda del producto *i* del periodo t_2 , i = 1, …, m, t_1, t_2=1, …, n, $t_1 \leq t_2$.

YE_t^i *(0,1)*: Variable binaria, 1 si se entrega la demanda del producto i del periodo t, 0 en caso contrario, i = 1, …, m t = 1, …, n.

Y_t^i *(0,1)*: Variable binaria, 1 si se fabrica alguna unidad del producto *i* en el periodo t, i = 1, …, m,

t = 1, …, n.

Función Objetivo

Maximizar el número total de demandas satisfechas y cantidad de producto servido en la fecha de entrega y al mismo tiempo minimizar la cantidad de producto servido con antelación y los costes, tanto fijos como variables.

$$\begin{aligned} \text{máx } Z = b\sum_{i=1}^{m}\sum_{t=1}^{n} YE_t^i + \sum_{i=1}^{m}\sum_{t=1}^{n} E_{t,t}^i - \sum_{i=1}^{m}\sum_{t=2}^{n}\sum_{k=1}^{t-1} f_{t-k}\cdot E_{k,t}^i \\ - \sum_{i=1}^{m}\sum_{t=1}^{n}(p_t^i\cdot X_t^i + h_t^i\cdot S_t^i + q_t^i\cdot Y_t^i) \end{aligned} \qquad (5.1)$$

Restricciones

[Capacidad de producción]

$$\sum_{i=1}^{m} X_t^i \leq c_t \quad t = 1, \dots, n \qquad (5.2)$$

[Capacidad de almacenamiento]

$$\sum_{i=1}^{m} S_t^i \leq a_t \qquad t = 1, \dots, n \tag{5.3}$$

[Uso de la variable binaria YE_t^i]

$$\sum_{k=1}^{t} E_{k,t}^i = d_t^i \cdot YE_t^i \qquad i = 1, \dots, m, t = 1, \dots, n \tag{5.4}$$

$$YE_t^i \leq d_t^i \qquad i = 1, \dots, m, t = 1, \dots, n \tag{5.5}$$

[Restricción de equilibrio]

$$S_{t-1}^i + X_t^i = S_t^i + \sum_{k=0}^{n-t} E_{t,t+k}^i \qquad i = 1, \dots, m, t = 1, \dots, n \tag{5.6}$$

[Uso de la variable binaria Y_t^i]

$$\varepsilon \cdot Y_t^i \leq X_t^i \leq c_t \cdot Y_t^i \qquad i = 1, \dots, m, t = 1, \dots, n \tag{5.7}$$

Siendo ε un número muy pequeño positivo.

[Condición de no negatividad]

$$X_t^i, S_t^i, E_{t_1,t_2}^i \geq 0 \quad i = 1, \dots, m, t, t_1, t_2 = 1, \dots, n,\ t_1 \leq\ t_2 \tag{5.8}$$

[Variables binarias]

$$Y_t^i, YE_t^i \ \ binarias \quad i = 1, \dots, m,\ t = 1, \dots, n \tag{5.9}$$

[Variables enteras]

$$X_t^i, S_t^i, E_{t_1,t_2}^i \ enteras \quad i = 1, \dots, m,\ t, t_1, t_2 = 1, \dots, n,\ t_1 \leq\ t_2 \tag{5.10}$$

En primer lugar, por lo que respecta a la función objetivo se puede observar que está expresada en unidades monetarias. Se parte de la F.O. anterior adicionando un cuarto término substractivo, relativo a los costes (costes fabricación, costes de almacenamiento y costes fijos), para tener en consideración los gastos que conlleva realizar las entregas en función de sus costes fijos y variables de producción y almacenamiento.

Por lo que respecta a las restricciones de capacidad (5.2) y (5.3), dado que se habla de capacidad total de fabricación y almacenamiento, se deberá tener en cuenta todos los productos, indistintamente de qué tipo de producto *i* sea. El resto de las ecuaciones son iguales a las mostradas en el modelo 4, extendiendo cada uno de los términos a los distintos tipos de producto *i* existentes.

En último lugar, se incluye una nueva variable binaria Y_t^i, para controlar los gastos fijos. Mediante la ecuación (5.7), la Y_t^i, toma valor 1 en el caso en el que se fabrique alguna unidad ($X_t^i \neq 0$), de este modo en la F.O., los costes fijos actúan o no en la función según si se decide producir o no un producto en un periodo.

4.6 Modelo 6. Modelo con desagregado de demanda a pedidos

La próxima especificación del algoritmo a incluir en este modelo es el concepto de pedido. Pueden existir diversos pedidos y cada pedido engloba un conjunto de demandas de distintos productos y cada uno con su fecha de entrega. Por lo tanto, es posible que exista una misma demanda en distintos pedidos a la vez. Tal y como se ilustra en la

Figura 14. Desagregado de demanda a pedidos, dónde existen: dos pedidos (pedido 1 y pedido 2), 3 productos distintos (A, B y C) y 16 demandas en total, 8 correspondientes a cada pedido. Y se puede observar que tanto en el pedido 1 como en el 2, se repite el producto A y existe una misma demanda en ambos.

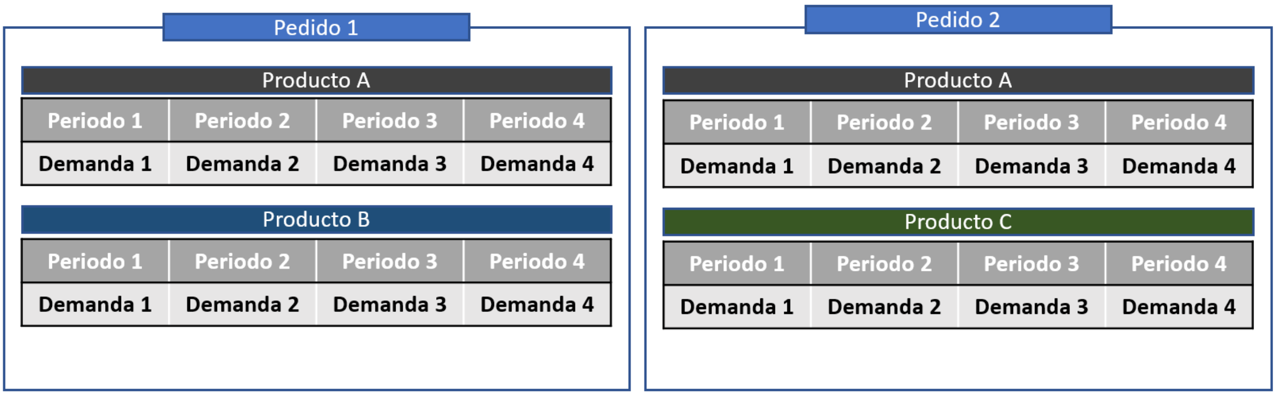

Figura 14. Desagregado de demanda a pedidos. Fuente: Elaboración propia.

Por este motivo será necesario identificar qué unidades se entregan para satisfacer una demanda, y a qué pedido pertenece. También será necesario, cumplir con la especificación de que únicamente se puede servir un pedido completo (se satisfacen todas las demandas del pedido) y no se permite entregar un pedido a medias si no están todos los productos del mismo servidos.

El objetivo de este modelo trata de servir o no pedidos completos. Y al igual que el algoritmo, en el caso en el que no se pueda servir un pedido al completo (todos los productos dentro del pedido), el pedido no se sirve, es decir que se sirven todos los productos del pedido, o ninguno).

- Monoproducto
- Multiperiodo
- Capacidad finita
- Con posibilidad de no servir una demanda
- Con posibilidad de entregar una demanda con antelación en cualquier periodo anterior
- Multiproducto
- **Desagregado de demanda a pedidos**

Tabla 7. Características del modelo 6.

Índices

t , t_1, t_2: Periodos que van de 1 a n.

g: Periodos que va de 1 a n-1.

i: Productos que va de 1 a m.

p: Pedidos que va de 1 a l.

Parámetros

c_t: Capacidad máxima de fabricación en el periodo t, (unidades producidas por periodo) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento en el periodo t, (unidades almacenadas por periodo) t = 1, …, n.

$d_t^{p,i}$: Demanda del producto i en el periodo t, que pertenece al pedido p, (unidades por periodo) p=1, …, l, i = 1, …, m, t = 1, …, n.

nd^p: Número de demandas distintas de cero, para todos los productos y periodos del pedido p, p=1, …, l.

S_0^i: Stock inicial del producto i (unidades) i = 1, …, m.

p_t^i: Coste de fabricar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

h_t^i: Coste de almacenar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

q_t^i: Coste fijo de producción del producto i en el periodo t, (unidades monetarias) i = 1, …, m, t = 1, …, n.

b^p: Bonificación económica por la entrega del pedido p, (unidades monetarias por pedido) p=1, …, l.

f_g: Penalización por servir una unidad de producto con adelanto en el periodo g, g= 1, …, n-1.

Variables

X_t^i: Unidades del producto *i* fabricadas en el periodo t, (unidades por periodo) *i* = 1, …, m, t = 1, …, n.

S_t^i: Unidades del producto *i* almacenadas en el periodo t, (unidades por periodo) *i* = 1, …, m, t = 1, …, n.

$E_{t_1,t_2}^{p,i}$: Unidades de producto i que corresponden al pedido p, que se sirven en el periodo t_1, correspondientes a la demanda con entrega en el periodo t_2 (unidades por periodo) p=1, …, *l*, *i* = 1, …, m, $t_1, t_2 = 1, \dots, n, t_1 \leq t_2$.

$YE_t^{p,i}$*(0,1)*: Variable binaria, 1 si se entrega la demanda del producto i correspondiente al pedido p en el periodo de t, 0 en caso contrario, p =1, …, *l*, *i* = 1, …, m, $t = 1, \dots, n$.

Y_t^i *(0,1)*: Variable binaria, 1 si se fabrica alguna unidad de producto i en el periodo t, 0 en caso contrario, *i* = 1, …, m, t = 1, …, n.

YP^p*(0,1)*: Variable binaria, 1 si se entrega el pedido p, 0 en caso contrario, p=1, …, l.

Función Objetivo

Maximizar el número pedidos entregados, tratando de entregar las demandas lo más cerca posible a la fecha de entrega y minimizar los costes.

$$\begin{aligned} \text{máx } Z = \sum_{p=1}^{l} b^p \cdot YP^p + \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=1}^{n} E_{t,t}^{p,i} - \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=2}^{n}\sum_{k=1}^{t-1} f_{t-k} \cdot E_{k,t}^{p,i} \\ - \sum_{i=1}^{m} \sum_{t=1}^{n} (p_t^i \cdot X_t^i + h_t^i \cdot S_t^i + q_t^i \cdot Y_t^i) \end{aligned} \tag{6.1}$$

Restricciones

[Capacidad de producción]

$$\sum_{i=1}^{m} X_t^i \leq c_t \quad t = 1, \dots, n \tag{6.2}$$

[Capacidad de almacenamiento]

$$\sum_{i=1}^{m} S_t^i \leq a_t \quad t = 1, \dots, n \tag{6.3}$$

[Uso de la variable binaria $YE_t^{p,i}$]

$$\sum_{k=1}^{t} E_{k,t}^{p,i} = d_t^{p,i} \cdot YE_t^{p,i} \qquad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{6.4}$$

$$YE_t^{p,i} \leq d_t^{p,i} \qquad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{6.5}$$

[Restricción de equilibrio]

$$S_{t-1}^i + X_t^i = S_t^i + \sum_{p=1}^{l} \sum_{k=0}^{n-t} E_{t,t+k}^{p,i} \qquad i = 1, \dots, m, t = 1, \dots, n \tag{6.6}$$

[Uso de la variable binaria Y_t^i]

$$\varepsilon \cdot Y_t^i \leq X_t^i \leq c_t \cdot Y_t^i \qquad i = 1, \dots, m, t = 1, \dots, n \tag{6.7}$$

Siendo ε un número muy pequeño positivo.

[Uso de la variable binaria YP^p]

$$nd_p \cdot YP^p = \sum_{i=1}^{m} \sum_{t=1}^{n} YE_t^{p,i} \qquad p = 1, \dots, l \tag{6.8}$$

[Condición de no negatividad]

$$X_t^i, S_t^i \geq 0 \quad i = 1, \dots, m, \ t = 1, \dots, n \tag{6.9}$$

$$E_{t_1,t_2}^{p,i} \geq 0 \qquad p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ t_1 \leq \ t_2 \tag{6.10}$$

[Variables binarias]

$$Y_t^i, \ binarias \ \ i = 1, \dots, m, \ \ t = 1, \dots, n \tag{6.11}$$

$$YE_t^{p,i} \ binarias \ \ p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{6.12}$$

$$YP^p \ binarias \ \ p = 1, \dots, l \tag{6.13}$$

[Variables enteras]

$$X_t^i, S_t^i \ enteras \quad i = 1, \dots, m, \ \ t, t_1, t_2 = 1, \dots, n, \ \ t_1 \leq \ t_2 \tag{6.14}$$

$$E_{t_1,t_2}^{p,i} \ enteras \qquad p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ \ t_1 \leq \ t_2 \tag{6.15}$$

Partiendo del modelo anterior, el primer paso es extender las variables relativas a entregas y demanda, a pedidos, añadiendo el superíndice p ($YE_t^{p,i}, E_{t_1,t_2}^{p,i}, d_t^{p,i}$). De este modo se puede identificar a qué pedido corresponden estas.

Para garantizar, que se entregue un pedido únicamente si se entregan todos los pedidos que incluye, se emplea la restricción (6.8). Mediante esta ecuación se impone que $YP^p = 1$ (se entrega un pedido), se dé solamente en el caso de que el número de demandas satisfechas sea igual al número de demandas existentes nd_p, para un pedido p. En el caso en el que no sean iguales, $YP^p = 0$, no se entrega el pedido y, por lo tanto, no entrega ninguna de las demandas del pedido (no se sirve ninguno de los productos del pedido), pues todas las $YE_t^{p,i}$, tomarían valor 0.

La última modificación por realizar está en la función objetivo. Dado que ahora se habla de entregar o no un pedido, lo que se busca es maximizar el número de pedidos entregados. Este es el motivo por el que, se sustituye el término asociado a la entrega de una demanda por la entrega de un pedido YP^p y el beneficio b^p asociado por entrega.

4.7 Modelo 7. Modelo multinivel

En este séptimo modelo se añade la demanda dependiente, tratándose ahora de un problema multinivel. Esto implica que la cantidad de producto a fabricar tendrá que satisfacer tanto la demanda independiente como la dependiente. La demanda dependiente viene dada por la Lista de Materiales o *Bill of Materials (BOM).* Esta indica qué cantidad y qué productos son necesarios para poder fabricar otro producto de nivel superior, así como los tiempos de suministro de cada uno.

- Monoproducto
- Multiperiodo
- Capacidad finita
- Con posibilidad de no servir una demanda
- Con posibilidad de entregar una demanda con antelación en cualquier periodo anterior
- Multiproducto
- Desagregado de demanda a pedidos
- **Multinivel**

Tabla 8. Características del modelo 7.

Índices

t , t_1, t_2: Periodos que van de 1 a *n.*

g: Periodos que va de 1 a n-1.

i, j: Productos que van de 1 a *m.*

p: Pedidos que va de 1 a l.

Parámetros

c_t: Capacidad máxima de fabricación en el periodo t, (unidades producidas por periodos)

t = 1, …, n.

a_t: Capacidad máxima de almacenamiento en el periodo t, (unidades almacenadas por periodos)

t = 1, …, n.

$d_t^{p,i}$: Demanda del producto i en el periodo t, que pertenece al pedido p, (unidades por periodo) p=1, …, l, i = 1, …, m, t = 1, …, n.

nd^p: Número de demandas distintas de cero, para todos los productos y periodos del pedido p, p=1, …, l.

S_0^i: Stock inicial del producto i (unidades) i = 1, …, m.

p_t^i: Coste de fabricar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

h_t^i: Coste de almacenar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

q_t^i: Coste fijo de producción del producto i en el periodo t, (unidades monetarias) i = 1, …, m,

t = 1, …, n.

b^p: Bonificación económica por la entrega del pedido p, (unidades monetarias por pedido)

p=1, …, l.

f_g: Penalización por servir una unidad de producto con adelanto en el periodo g, g= 1, …, n-1.

$r^{i,j}$: Cantidad de producto i requerida para producir una unidad de producto j, (unidades)

i = 1, …, m, j = 1, …, m, $i \neq j$.

γ^i: *Lead-time*, tiempo requerido para producir una unidad de producto i (periodos).

$D(i)$: Conjunto de productos que demandan al producto i, i = 1, …, m.

Variables

X_t^i: Unidades del producto i fabricadas en el periodo t, que estarán disponibles en el periodo

$t + \gamma_i$ (unidades por periodo) i = 1, …, m, t = 1, …, n.

S_t^i: Unidades del producto i almacenadas en el periodo t, (unidades por periodo) i = 1, …, m,

t = 1, …, n.

$E_{t_1,t_2}^{p,i}$: Unidades de producto i que corresponden al pedido p, que se sirven en el periodo t_1, correspondientes a la demanda con entrega en el periodo t_2 (unidades por periodo) p=1, …, l,

i = 1, …, m, $t_1, t_2 = 1, \dots, n, t_1 \leq t_2$.

$YE_t^{p,i}$*(0,1)*: Variable binaria, 1 si se entrega la demanda del producto i correspondiente al pedido p en el periodo t, 0 en caso contrario, p =1, …, l, i = 1, …, m, $t = 1, \dots, n$.

Y_t^i *(0,1)*: Variable binaria, 1 si se fabrica alguna unidad de producto i en el periodo t, 0 en caso contrario, i = 1, …, m, t = 1, …, n.

YP^p*(0,1)*: Variable binaria, 1 si se entrega el pedido p, 0 en caso contrario, p=1, …, l.

Función Objetivo

Maximizar el número pedidos entregados, tratando de entregar las demandas lo más cerca posible a la fecha de entrega y minimizar los costes.

$$\text{máx } Z = \sum_{p=1}^{l} b^p \cdot YP^p + \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=1}^{n} E_{t,t}^{p,i} - \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=2}^{n}\sum_{k=1}^{t-1} f_{t-k} \cdot E_{k,t}^{p,i} - \sum_{i=1}^{m}\sum_{t=1}^{n}(p_t^i \cdot X_t^i + h_t^i \cdot S_t^i + q_t^i \cdot Y_t^i) \quad (7.1)$$

Restricciones

[Capacidad de producción]

$$\sum_{i=1}^{m} X_t^i \le c_t \quad t = 1, \dots, n \quad (7.2)$$

[Capacidad de almacenamiento]

$$\sum_{i=1}^{m} S_t^i \le a_t \quad t = 1, \dots, n \quad (7.3)$$

[Uso de la variable binaria $YE_t^{p,i}$]

$$\sum_{k=1}^{t} E_{k,t}^{p,i} = d_t^{p,i} \cdot YE_t^{p,i} \quad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \quad (7.4)$$

$$YE_t^{p,i} \le d_t^{p,i} \quad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \quad (7.5)$$

[Restricción de equilibrio]

$$S_{t-1}^i + X_{t-\gamma_i}^i = S_t^i + \sum_{p=1}^{l}\sum_{k=0}^{n-t} E_{t,t+k}^{p,i} + \sum_{j \in D(i)} r^{i,j} \cdot X_t^j \quad i, j = 1, \dots, m, t = 1, \dots, n \quad (7.6)$$

[Uso de la variable binaria Y_t^i]

$$\varepsilon \cdot Y_t^i \leq X_t^i \leq c_t \cdot Y_t^i \qquad i = 1, \dots, m, t = 1, \dots, n \tag{7.7}$$

Siendo ε un número muy pequeño positivo.

[Uso de la variable binaria YP^p]

$$nd_p \cdot YP^p = \sum_{i=1}^{m} \sum_{t=1}^{n} YE_t^{p,i} \qquad p = 1, \dots, l \tag{7.8}$$

[Condición de no negatividad]

$$X_t^i, S_t^i \geq 0 \qquad i, j = 1, \dots, m, \ t = 1, \dots, n \tag{7.9}$$

$$E_{t_1,t_2}^{p,i} \geq 0 \qquad p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ t_1 \leq \ t_2 \tag{7.10}$$

[Variables binarias]

$$Y_t^i \ \ binarias \ \ i = 1, \dots, m, \ \ t = 1, \dots, n \tag{7.11}$$

$$YE_t^{p,i} \ binarias \ \ p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{7.12}$$

$$YP^p \ binarias \ \ p = 1, \dots, l \tag{7.13}$$

[Variables enteras]

$$X_t^i, X_t^j, S_t^i \ enteras \quad i = 1, \dots, m, \ \ t, t_1, t_2 = 1, \dots, n, \ \ t_1 \leq \ t_2 \tag{7.14}$$

$$E_{t_1,t_2}^{p,i} \ enteras \quad p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ \ t_1 \leq \ t_2 \tag{7.15}$$

Por una parte, la demanda independiente viene dada por $d_t^{p,i}$, y al igual que en los modelos anteriores se satisface haciendo uso de la ecuación (7.4). Dónde se impone que las cantidades de producto que se entreguen deban satisfacer exactamente la demanda.

Por otro lado, para poder integrar la demanda dependiente hay que realizar dos cambios importantes en la restricción de equilibrio (7.6). El primer cambio a realizar es modificar la variable X_t^i por $X_{t-\gamma_i}^i$. Dado que ahora, para disponer de una unidad fabricada debe transcurrir un tiempo determinado, denominado tiempo de suministro o *lead-time* (γ_i). Por tanto, la equivalencia de $X_{t-\gamma_i}^i$, es la cantidad de producto lanzada a fabricación en el periodo $t - \gamma_i$, y que trascurridos γ_i periodos, serán unidades que estarán disponibles en el periodo t. El segundo cambio a realizar es la adición del término $\sum_{j \in D(i)} r^{i,j} \cdot X_t^j$, mediante el cual se representa la estructura del BOM, ya que indica la cantidad de producto necesaria debido a la fabricación de un producto de un nivel superior. El parámetro $r^{i,j}$, indica la cantidad de producto i necesaria para fabricar una unidad de j. Donde j son los productos de nivel superior cuyos sucesores directos son un conjunto de productos $i, D(i)$. Además, se entiende que $i \neq j$, puesto que un producto no puede requerir de si mismo para fabricarse.

Cabe añadir que, para los problemas multinivel, es necesario planificar con un horizonte temporal suficientemente largo que sea al menos igual de largo que el tiempo máximo necesario para fabricar cualquier producto final desde su nivel más bajo. Obsérvese Figura 15. En este ejemplo, considerando los tiempos de suministro indicados en la figura, la duración máxima posible (el camino de mayor duración) que conllevaría fabricar el producto final A sería de 6 periodos ($\gamma^E = 2, \gamma^B = 3 \ y \ \gamma^A = 1$). Por lo tanto, el número de periodos del horizonte de planificación n en este caso debería ser $n \geq 6$. Alternativamente, para conectar una planificación con otra, si este no fuera el caso, entonces debería haber lanzado a producción las materias primeras o semielaborados de los niveles más bajos, so pena de no poder nunca entregar producto.

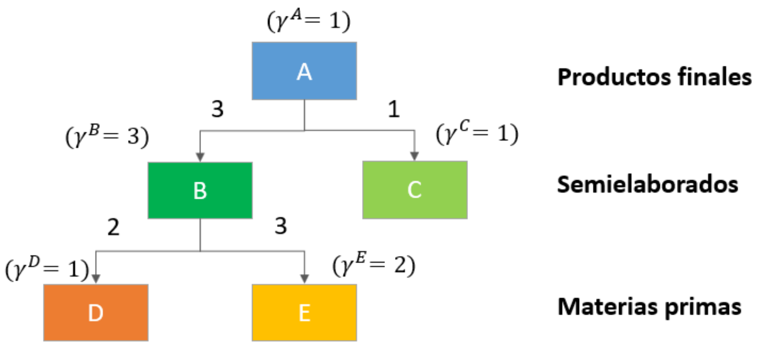

Figura 15. Ejemplo BOM y lead-times. Fuente: Elaboración propia.

El resto de las restricciones e incluso la F.O. se mantienen igual, ya que no se ven afectadas por la incorporación de la demanda dependiente de productos.

4.8 Modelo 8. Modelo con órdenes de compra y priorización de un pedido

El modelo anterior únicamente contemplaba órdenes de fabricación, ya que solo indicaba en qué momento se debía realizar la fabricación de pedidos. En cambio, en este modelo se van a incluir las órdenes de compra, que dan la posibilidad de realizar la compra de productos que por ejemplo no se puedan fabricar por falta de capacidad de producción o almacenamiento. Para poder incluir tanto las órdenes de fabricación como de compra, será necesario diferenciar que unidades provienen de la fabricación y cuáles de la compra.

La segunda característica que incluir en el modelo es la priorización de un pedido, en el que se posibilita dar una mayor importancia a un pedido con el objetivo de intentar entregarlo a toda costa, es decir de forma irrevocable.

▪ Monoproducto ▪ Multiperiodo ▪ Capacidad finita ▪ Con posibilidad de no servir una demanda ▪ Con posibilidad de entregar una demanda con antelación en cualquier periodo anterior ▪ Multiproducto ▪ Desagregado de demanda a pedidos ▪ Multinivel ▪ **Órdenes de compra** ▪ **Priorización de entrega de un pedido**

Tabla 9. Características del modelo 8.

Índices

t , t_1, t_2: Periodos que van de 1 a n.

g: Periodos que va de 1 a n-1.

i, j: Productos que van de 1 a m.

p: Pedidos que va de 1 a l.

Parámetros

c_t: Capacidad máxima de fabricación en el periodo t, (unidades producidas por periodos)

t = 1, …, n.

a_t: Capacidad máxima de almacenamiento en el periodo t, (unidades almacenadas por periodos)

t = 1, …, n.

$d_t^{p,i}$: Demanda del producto i en el periodo t, que pertenece al pedido p, (unidades por periodo) p=1, …, l, i = 1, …, m, t = 1, …, n.

nd^p: Número de demandas distintas de cero, para todos los productos y periodos del pedido p, p=1, …, l.

S_0^i: Stock inicial del producto i (unidades) i = 1, …, m.

p_t^i: Coste de fabricar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

pc_t^i: Coste de comprar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

h_t^i: Coste de almacenar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

q_t^i: Coste fijo de producción del producto i en el periodo t, (unidades monetarias) i = 1, …, m,

t = 1, …, n.

qc_t^i: Coste fijo de compra del producto i en el periodo t, (unidades monetarias) i = 1, …, m,

t = 1, …, n.

b^p: Bonificación económica por la entrega del pedido p, (unidades monetarias por pedido)

p=1, …, l.

v^p: Bonificación económica por la entrega prioritaria del pedido p, (unidades monetarias por pedido) p=1, …, l.

f_g: Penalización por servir alguna unidad de producto con adelanto en el periodo g, g= 1, …, n.

$r^{i,j}$: Cantidad de producto i requerida para producir una unidad de producto j, (unidades)

i = 1, …, m, j = 1, …, m, $i \neq j$.

$D(i)$: Conjunto de productos que demandan al producto i, i = 1, …, m.

γ^i: *Lead-time*, tiempo requerido para producir una unidad de producto i (periodos).

φ^i: *Lead-time*, tiempo requerido para recepcionar una compra de una unidad de producto i (periodos).

Variables

X_t^i: Unidades del producto i fabricadas en el periodo t, que estarán disponibles en el periodo

$t + \gamma_i$ (unidades por periodo) i = 1, …, m, t = 1, …, n.

XC_t^i: Unidades del producto i compradas en el periodo t, que estarán disponibles en el periodo

$t + \varphi^i$ (unidades por periodo) i = 1, …, m, t = 1, …, n.

S_t^i: Unidades del producto i almacenadas en el periodo t, (unidades por periodo) i = 1, …, m,

t = 1, …, n.

$E_{t_1,t_2}^{p,i}$: Unidades de producto i que corresponden al pedido p, que se sirven en el periodo t_1, correspondientes a la demanda con entrega en el periodo t_2 (unidades por periodo) p=1, …, l,

i = 1, …, m, $t_1, t_2 = 1, \dots, n$, $t_1 \leq t_2$.

$YE_t^{p,i}$*(0,1)*: Variable binaria, 1 si se entrega la demanda del producto i correspondiente al pedido p en el periodo t , 0 en caso contrario, p =1, …, l, i = 1, …, m, $t = 1, \dots, n$.

Y_t^i *(0,1)*: Variable binaria, 1 si se fabrica alguna unidad de producto i en el periodo t, 0 en caso contrario, i = 1, …, m, t = 1, …, n.

YC_t^i *(0,1)*: Variable binaria, 1 si se compra alguna unidad de producto i en el periodo t, 0 en caso contrario, i = 1, …, m, t = 1, …, n.

YP^p*(0,1)*: Variable binaria, 1 si se entrega el pedido p, 0 en caso contrario, p=1, …, l.

Función Objetivo

Maximizar el número de pedidos entregados, teniendo en cuenta qué pedidos tienen mayor prioridad. Tratando de entregar las demandas lo más cerca posible a la fecha de entrega. Y minimizar los costes.

$$\begin{aligned} \text{máx } Z = \sum_{p=1}^{l} v^p \cdot b^p \cdot YP^p + \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=1}^{n} E_{t,t}^{p,i} - \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=2}^{n}\sum_{k=1}^{t-1} f_{t-k} \cdot E_{k,t}^{p,i} \\ - \sum_{i=1}^{m}\sum_{t=1}^{n}(p_t^i \cdot X_t^i + h_t^i \cdot S_t^i + q_t^i \cdot Y_t^i + pc_t^i \cdot XC_t^i + qc_t^i \cdot YC_t^i) \end{aligned} \tag{8.1}$$

Restricciones

[Capacidad de producción]

$$\sum_{i=1}^{m} X_t^i \leq c_t \quad t = 1, \dots, n \tag{8.2}$$

[Capacidad de almacenamiento]

$$\sum_{i=1}^{m} S_t^i \leq a_t \quad t = 1, \dots, n \tag{8.3}$$

[Uso de la variable binaria $YE_t^{p,i}$]

$$\sum_{k=1}^{t} E_{k,t}^{p,i} = d_t^{p,i} \cdot YE_t^{p,i} \qquad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{8.4}$$

$$YE_t^{p,i} \leq d_t^{p,i} \qquad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{8.5}$$

[Restricción de equilibrio]

$$\begin{gathered} S_{t-1}^i + X_{t-\gamma_i}^i + XC_{t-\varphi_i}^i = S_t^i + \sum_{p=1}^{l}\sum_{k=0}^{n-t} E_{t,t+k}^{p,i} + \sum_{j \in D(i)} r^{i,j} \cdot (X_t^j + XC_t^j) \\ i = 1, \dots, m, t = 1, \dots, n \end{gathered} \tag{8.6}$$

[Uso de la variable binaria Y_t^i]

$$\varepsilon \cdot Y_t^i \leq X_t^i \leq c_t \cdot Y_t^i \quad i = 1, \dots, m, t = 1, \dots, n \tag{8.7}$$

Siendo ε un número muy pequeño positivo.

[Uso de la variable binaria YC_t^i]

$$\varepsilon \cdot YC_t^i \leq XC_t^i \leq \alpha \cdot YC_t^i \quad i = 1, \dots, m, t = 1, \dots, n \tag{8.8}$$

Siendo ε un número muy pequeño positivo y α un número muy grande positivo.

[Uso de la variable binaria YP^p]

$$nd_p \cdot YP^p = \sum_{i=1}^{m} \sum_{t=1}^{n} YE_t^{p,i} \quad p = 1, \dots, l \tag{8.9}$$

[Condición de no negatividad]

$$X_t^i, XC_t^i, S_t^i \geq 0 \quad i = 1, \dots, m, t = 1, \dots, n \tag{8.10}$$

$$E_{t_1,t_2}^{p,i} \geq 0 \quad p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ t_1 \leq t_2 \tag{8.11}$$

[Variables binarias]

$$Y_t^i, YC_t^i \ binarias \ i = 1, \dots, m, \ t = 1, \dots, n \tag{8.12}$$

$$YE_t^{p,i} \ binarias \ p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{8.13}$$

$$YP^p \ binarias \ p = 1, \dots, l \tag{8.14}$$

[Variables enteras]

$$X_t^i, S_t^i \ enteras \ i = 1, \dots, m, \ t, t_1, t_2 = 1, \dots, n, \ t_1 \leq t_2 \tag{8.15}$$

$$E_{t_1,t_2}^{p,i} \ enteras \ p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ t_1 \leq t_2 \tag{8.16}$$

Tal y como se ha mencionado al principio, el primer paso es duplicar las variables relativas a la fabricación X_t^i y Y_t^i y se crean XC_t^i y YC_t^i. Al igual que los parámetros γ^i, p_t^i y q_t^i que conlleva φ^i, pc_t^i y qc_t^i.

Estas nuevas variables, aparecen en la función objetivo, ya que no solo habrá costes asociados a la fabricación sino también a la compra, como costes por la compra de productos y costes fijos de compra. Estos gastos se representan con los dos últimos sumandos de la F.O. (8.1).

Otro cambio que realizar en la F.O., es la adición de la constante v^p delante de la bonificación por la entrega de un pedido (b^p), de esta manera se permite incrementar el valor b^p según lo urgente que sea la entrega un pedido. Contra mayor sea el valor de v^p, más prioritario será el pedido al que va asociada la constante.

Por lo que respecta a las restricciones, la que sufre más modificaciones es la restricción de equilibrio (8.6). Al igual que en el modelo anterior, para incorporar la demanda dependiente de los productos fabricados, se debe incluir ahora la demanda dependiente de productos comprados. Se añade el sumando $XC_{t-\varphi_i}^i$, como unidades compradas en el periodo $t - \varphi_i$, y que estarán disponibles en el periodo t. Y para representar el BOM tanto de los productos fabricados como comprados, se modifica la parte derecha de la igualdad por $\sum_{j \in D(i)} r^{i,j} \cdot \left(X_t^j + XC_t^j\right)$. De esta manera se satisface tanto la demanda de un artículo fabricado de un nivel superior, como la demanda de un artículo de compra de nivel superior. Además, dado que en muchos casos los tiempos de suministros son distintos en la compra y la fabricación, se han incluido dos parámetros distintos para cada uno de ellos, respectivamente, φ^iy γ^i.

Y por último al igual que existen costes fijos asociados a la fabricación, también lo hay debidos a la compra. Por este motivo se introduce la restricción (8.8), donde según si se realiza la compra de algún producto, la YC_t^i toma valor 1, en la F.O. implicaría considerar los costes fijos en ese periodo y de ese producto.

4.9 Modelo 9. Modelo sin posibilidad de *splitting*

Se concluye este cuarto capítulo, con la propuesta final del modelo que replica todas las especificaciones del algoritmo MRPc de GeInfor y presenta la misma funcionalidad.

En los modelos planteados del 1 al 8 se permitía servir una demanda en varios periodos, esto se conoce como *splitting*. Sin embargo, el algoritmo tiene una especificación que indica que solamente permite servir una demanda en un único periodo, por lo que no permite el *splitting*. En otras palabras, indica que una demanda debe servirse en su totalidad y en una sola vez. Se puede adelantar la demanda, pero solo se puede entregar en un periodo (si se entrega). Para satisfacer esta condición, se realizarán una serie de modificaciones a partir del modelo anterior.

- Monoproducto
- Multiperiodo
- Capacidad finita
- Con posibilidad de no servir una demanda
- Con posibilidad de entregar una demanda con antelación en cualquier periodo anterior
- Multiproducto
- Desagregado de demanda a pedidos
- Multinivel
- Órdenes de compra
- Priorización de entrega de un pedido
- **Sin *splitting***

Tabla 10. Características del modelo 9.

Índices

t , t_1, t_2: Periodos que van de 1 a n.

g: Periodos que va de 1 a n-1.

i, j: Productos que van de 1 a m.

p: Pedidos que va de 1 a l.

Parámetros

c_t: Capacidad máxima de fabricación en el periodo t, (unidades producidas por periodos) t = 1, …, n.

a_t: Capacidad máxima de almacenamiento en el periodo t, (unidades almacenadas por periodos) t = 1, …, n.

$d_t^{p,i}$: Demanda del producto i en el periodo t, que pertenece al pedido p, (unidades por periodo) p=1, …, l, i = 1, …, m, t = 1, …, n.

nd^p: Número de demandas distintas de cero, para todos los productos y periodos del pedido p, p=1, …, l.

S_0^i: Stock inicial del producto i (unidades) i = 1, …, m.

p_t^i: Coste de fabricar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

pc_t^i: Coste de comprar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

h_t^i: Coste de almacenar una unidad del producto i en el periodo t, (unidades monetarias por unidad de producto) i = 1, …, m, t = 1, …, n.

q_t^i: Coste fijo de producción del producto i en el periodo t, (unidades monetarias) i = 1, …, m, t = 1, …, n.

qc_t^i: Coste fijo de compra del producto i en el periodo t, (unidades monetarias) i = 1, …, m, t = 1, …, n.

b^p: Bonificación económica por la entrega del pedido p, (unidades monetarias por pedido) p=1, …, l.

v^p: Bonificación económica por la entrega prioritaria del pedido p, (unidades monetarias por pedido) p=1, …, l.

f_g: Penalización por servir una unidad de producto con adelanto en el periodo g, g = 1, …, n-1.

$r^{i,j}$: Cantidad de producto i requerida para producir una unidad de producto j, (unidades) i = 1, …, m, j = 1, …, m, $i \neq j$.

$D(i)$: Conjunto de productos que demandan al producto i, i = 1, …, m.

γ^i: *Lead-time*, tiempo requerido para producir una unidad de producto *i* (periodos).

φ^i: *Lead-time*, tiempo requerido para recepcionar una compra de una unidad de producto *i* (periodos).

Variables

X_t^i: Unidades del producto *i* fabricadas en el periodo t, que estarán disponibles en el periodo $t+\gamma_i$ (unidades por periodo) *i* = 1, …, m, *t* = 1, …, n.

XC_t^i: Unidades del producto *i* compradas en el periodo t, que estarán disponibles en el periodo $t+\varphi^i$ (unidades por periodo) *i* = 1, …, m, *t* = 1, …, n.

S_t^i: Unidades del producto *i* almacenadas en el periodo t, (unidades por periodo) *i* = 1, …, m, *t* = 1, …, n.

$E_{t_1,t_2}^{p,i}$: Unidades de producto i que corresponden al pedido p, que se sirven en el periodo t_1, correspondientes a la demanda con entrega en el periodo t_2 (unidades por periodo) p=1, …, *l*, *i* = 1, …, m, $t_1, t_2 = 1, \dots, n, t_1 \leq t_2$.

$W_{t_1,t_2}^{p,i}$*(0,1)*: Variable binaria, toma el valor 1 si se entrega producto i en el periodo t_1 correspondiente a la demanda del periodo t_2, perteneciente al pedido p, 0 en caso contrario, p=1, …, *l*, *i* = 1, …, m, $t_1, t_2 = 1, \dots, n, t_1 \leq t_2$.

Y_t^i *(0,1)*: Variable binaria, 1 si se fabrica alguna unidad de producto i en el periodo t, 0 en caso contrario, *i* = 1, …, *m*, *t* = 1, …, *n*.

YC_t^i *(0,1)*: Variable binaria, 1 si se compra alguna unidad de producto i en el periodo t, 0 en caso contrario, *i* = 1, …, *m*, *t* = 1, …, *n*.

YP^p*(0,1)*: Variable binaria, 1 si se entrega el pedido p, 0 en caso contrario, p=1, …, l.

Función Objetivo

Maximizar el de número pedidos entregados, teniendo en cuenta qué pedidos tienen mayor prioridad. Entregando las demandas lo más cerca posible a la fecha de entrega. Y minimizar los costes.

$$\begin{aligned} \text{máx } Z = \sum_{p=1}^{l} v^p \cdot b^p \cdot YP^p + \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=1}^{n} E_{t,t}^{p,i} - \sum_{p=1}^{l}\sum_{i=1}^{m}\sum_{t=2}^{n}\sum_{k=1}^{t-1} f_{t-k} \cdot E_{k,t}^{p,i} \\ - \sum_{i=1}^{m}\sum_{t=1}^{n}(p_t^i \cdot X_t^i + h_t^i \cdot S_t^i + q_t^i \cdot Y_t^i + pc_t^i \cdot XC_t^i + qc_t^i \cdot YC_t^i) \end{aligned} \tag{9.1}$$

Restricciones

[Capacidad de producción]

$$\sum_{i=1}^{m} X_t^i \leq c_t \quad t = 1, \dots, n \tag{9.2}$$

[Capacidad de almacenamiento]

$$\sum_{i=1}^{m} S_t^i \leq a_t \quad t = 1, \dots, n \tag{9.3}$$

[Uso de la variable binaria $W_{k,t}^{p,i}$]

$$\varepsilon \cdot W_{t_1,t_2}^{p,i} \leq E_{t_1,t_2}^{p,i} \leq d_t^{p,i} \cdot W_{t_1,t_2}^{p,i} \quad t, t_1, t_2 = 1, \dots, n, \; t_1 \leq t_2 \tag{9.4}$$

Siendo ε un número muy pequeño positivo.

$$\sum_{k=1}^{t} E_{k,t}^{p,i} = d_t^{p,i} \cdot \sum_{k=1}^{t} W_{k,t}^{p,i} \quad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{9.5}$$

$$\sum_{k=1}^{t} W_{k,t}^{p,i} \leq d_t^{p,i} \quad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{9.6}$$

[Restricción de equilibrio]

$$S_{t-1}^i + X_{t-\gamma_i}^i + XC_{t-\varphi_i}^i = S_t^i + \sum_{p=1}^{l} \sum_{k=0}^{n-t} E_{t,t+k}^{p,i} + \sum_{j \in D(i)} r^{i,j} \cdot (X_t^j + XC_t^j) \qquad i = 1, \dots, m, t = 1, \dots, n \tag{9.7}$$

[Uso de la variable binaria Y_t^i]

$$\varepsilon \cdot Y_t^i \leq X_t^i \leq c_t \cdot Y_t^i \quad i = 1, \dots, m, t = 1, \dots, n \tag{9.8}$$

Siendo ε un número muy pequeño positivo.

[Uso de la variable binaria YC_t^i]

$$\varepsilon \cdot YC_t^i \leq XC_t^i \leq \alpha \cdot YC_t^i \quad i = 1, \dots, m, t = 1, \dots, n \tag{9.9}$$

Siendo ε un número muy pequeño positivo y α un número muy grande positivo.

[Uso de la variable binaria YP^p]

$$nd_p \cdot YP^p = \sum_{i=1}^{m} \sum_{k=1}^{t} \sum_{t=1}^{n} W_{k,t}^{p,i} \quad p = 1, \dots, l \tag{9.10}$$

[Sin *Splitting*]

$$\sum_{k=1}^{t} W_{k,t}^{p,i} \leq 1 \qquad p = 1, \dots, l, i = 1, \dots, m, t = 1, \dots, n \tag{9.11}$$

[Condición de no negatividad]

$$X_t^i, S_t^i \geq 0 \quad i, j = 1, \dots, m, t = 1, \dots, n \tag{9.12}$$

$$E_{t_1,t_2}^{p,i} \geq 0 \quad p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ t_1 \leq \ t_2 \tag{9.13}$$

[Variables binarias]

$$Y_t^i, YC_t^i \ \ binarias \ \ i = 1, \dots, m, \ \ t = 1, \dots, n \tag{9.14}$$

$$W_{t_1,t_2}^{p,i} \ \ binarias \ \ p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ t_1 \leq \ t_2 \tag{9.15}$$

$$YP^p \ binarias \ \ p = 1, \dots, l \tag{9.16}$$

[Variables enteras]

$$X_t^i, S_t^i \ enteras \ \ i = 1, \dots, m, \ \ t, t_1, t_2 = 1, \dots, n, \ \ t_1 \leq \ t_2 \tag{9.17}$$

$$E_{t_1,t_2}^{p,i} \ enteras \ \ p = 1, \dots, l, i = 1, \dots, m, t_1, t_2 = 1, \dots, n, \ \ t_1 \leq \ t_2 \tag{9.18}$$

Como se puede observar la F.O. es exactamente igual a la del modelo anterior, ya que no se ve afectada por el hecho de que no se permita hacer *splitting* en la entrega.

Para cumplir con la condición de *"no splitting"*, vemos que la variable definida en el modelo anterior $YE_t^{p,i}$, se ha eliminado, ya que resulta redundante al incorporar la nueva variable $W_{t_1,t_2}^{p,i}$. Por tanto, en aquellas restricciones en las que apareciese la variable $YE_t^{p,i}$, deben modificarse, empleando una expresión equivalente con la nueva variable $W_{t_1,t_2}^{p,i}$. Como en la ecuación (9.5) y (9.10), dónde se ha sustituido $YE_t^{p,i}$ por $\sum_{k=1}^{t} W_{k,t}^{p,i}$, puesto que satisfacer una demanda ($YE_t^{p,i} = 1$) es lo mismo que decir que existe un periodo en el que se sirve la demanda ($\sum_{k=1}^{t} W_{k,t}^{p,i} = 1$). El hecho de que este sumatorio, tome valor 1 como máximo, se debe a la restricción (9.11) que se comenta en el siguiente párrafo.

La última característica que falta incorporar al modelo es garantizar que no se produzca *splitting*. Por esta razón se añade la restricción (9.11), donde la variable $W_{t_1,t_2}^{p,i}$ indica si se ha entregado o no, el producto correspondiente a la demanda de un pedido. Y, al imponer, que la suma de todas ellas a lo largo de todos los periodos sea menor o igual a 1, se consigue, que solo una de las variables tome valor 1. Por tanto, se realiza únicamente una entrega de la demanda y no se produce *splitting*.

Resumiendo, a lo largo de este capítulo 4 se han visto las distintas fases por las que ha pasado el modelo y qué cambios se han tenido que realizar, para ir incorporado de manera acumulada las características del algoritmo MRPc. Comenzando desde el modelo estándar de la producción (modelo 1), hasta este último modelo (modelo 9), donde se propone el modelo matemático final que recoge todas las especificaciones necesarias para replicar el funcionamiento del algoritmo MRP de la empresa GeInfor.

5 SOLUCIÓN DE LOS MODELOS

El paso siguiente a la formulación de los modelos matemáticos es la resolución de los mismos, mediante herramientas de optimización. El hecho de resolver un modelo permite comprobar si la solución obtenida es acorde a la esperada. Por este motivo, para verificar, que un modelo está correctamente formulado, es necesario plantear distintos escenarios y ver si la solución calculada en cada caso tiene sentido y por tanto es la que cabría esperar.

En este capítulo, se proponen algunos casos de ejemplo para resolver. Los modelos que van del 1 al 4 se resolverán con el Solver integrado en la herramienta Excel, mientras que los modelo del 5 al 9 se resolverán mediante el Solver profesional CPLEX. El cambio de emplear una herramienta u otra se debe a la dificultad de resolver el modelo en una hoja de cálculo a partir de un determinado número de restricciones y variables. Además, como se verá más adelante, el hecho de programar un modelo en CPLEX ofrece muchas más ventajas como la flexibilidad y la sencillez de modelar un problema. Ya que CPLEX emplea un lenguaje de programación orientado a modelar problemas matemáticos.

5.1 Modelo 1. Solución de Excel

Modelo estándar de la producción

En el apartado 4.1 Modelo 1. Modelo estándar de la planificación de la producción, se ha visto la formulación correspondiente a dicho modelo. En este apartado, se plantea un ejemplo muy sencillo con únicamente un producto y solo tres periodos de planificación, que busca minimizar los costes. Este problema se resuelve mediante la herramienta Excel. Dado que es el primer ejemplo a resolver con dicha herramienta, se explicarán los pasos necesarios para poder calcular la solución.

Los datos del problema son los siguientes:

Parámetro	$t1$	$t2$	$t3$
d_t	10	20	5
p_t	2	1	3
h_t	5	2	4
c_t	15	15	15
a_t	10	10	10

Parámetro	
s_0	5

Tabla 11. Datos del modelo 1, ejemplo 1.

El primer paso, es introducir los datos de entrada (los parámetros) y establecer las celdas cambiantes (variables de decisión) y la celda objetivo (función objetivo). También hay que añadir las condiciones a las que está sujeto el problema. En la figura siguiente se muestra la hoja de cálculo empleada para resolver dicho problema. Empleando el código de colores correspondiente: azul (parámetros), naranja (variables decisión) y verde (valor de la F.O.).

Modelo 1. Planificación estándar de la producción

Demanda (dt)	T1	T2	T3
Producto	10	20	5

Coste (Z)	0

Stock inicial (S0)	
Producto	5

Coste de fabricación (pt)	T1	T2	T3
Producto	5	2	4

Coste de almacenamiento (ht)	T1	T2	T3
Producto	2	1	3

Fabricación (Xt)	T1	T2	T3
Producto			

Almacenamiento (St)	T1	T2	T3
Producto			

Restricción de equilibrio			
T1	5	=	10
T2	0	=	20
T3	0	=	5

Restricciones de capacidad			
Capacidad de fabricar (ct)			
Producto	T1	T2	T3
Disponible	0	0	0
	<=	<=	<=
Máxima	15	15	15
Capacidad de almacenar (at)			
Producto	T1	T2	T3
Disponible	0	0	0
	<=	<=	<=
Máxima	10	10	10

Figura 16. Hoja de cálculo modelo 1. Fuente: Elaboración propia.

Una vez descrito el problema en la hoja de cálculo, es necesario introducir los parámetros de Solver para que calcule la solución.

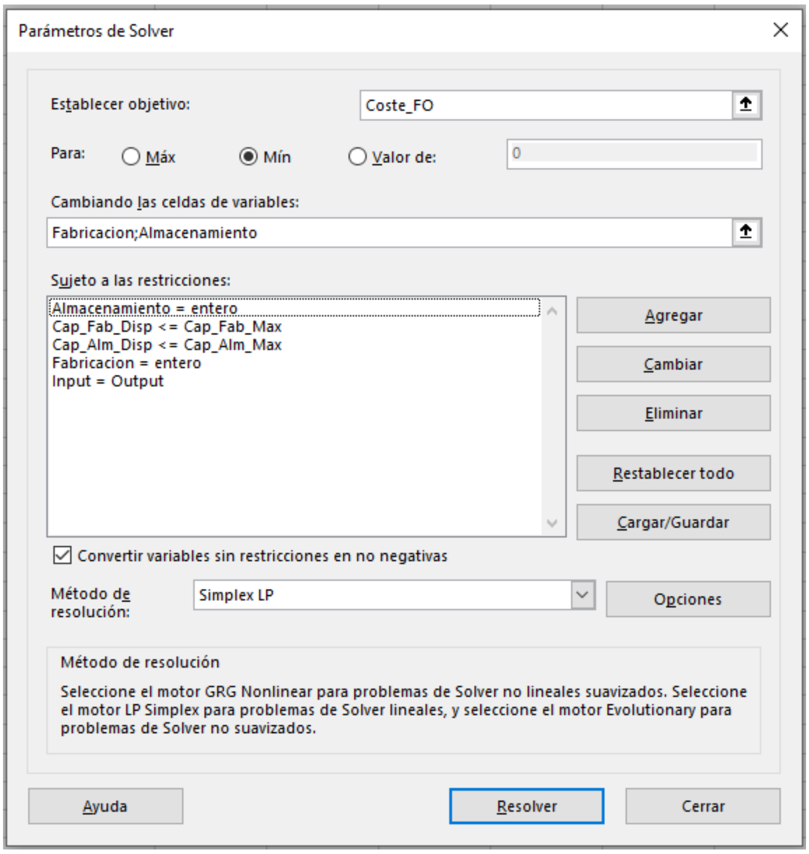

Figura 17. Interfaz Solver Excel

Como se observa en la figura, se estable como F.O. la celda correspondiente al coste y se marca el objetivo, en este caso, minimizar. Junto con las celdas variables que son “fabricación” y “almacenamiento”, que se corresponden a las variables decisión X_t y S_t. Posteriormente, queda agregar las restricciones: la restricción de capacidad de fabricación y almacenamiento y la restricción de equilibrio (“input”=”output”). Dada la condición de enteras de las unidades de stock y de fabricación, también debe indicarse como un tipo de restricción. Por último, queda seleccionar la casilla “Convertir las variables sin restricciones en no negativas” para imponer la condición de no negatividad de las variables y seleccionar el algoritmo que se quiere emplear, “Simplex LP”.

Dado que el modelo presenta un tamaño muy reducido, Excel proporciona la solución óptima en apenas un segundo.

Solución	$t1$	$t2$	$t3$
$X_t\ (uds.)$	10	15	5
$S_t(uds.)$	5	0	0

F.O.	
Z (€)	110

Tabla 12. Solución modelo 1, ejemplo 1.

Modelo 1. Planificación estándar de la producción

Demanda (dt)	T1	T2	T3
Producto	10	20	5

Coste (Z)	110

Stock inicial (S0)	
Producto	5

Coste de fabricación (pt)	T1	T2	T3
Producto	5	2	4

Coste de almacenamiento (ht)	T1	T2	T3
Producto	2	1	3

Fabricación (Xt)	T1	T2	T3
Producto	10	15	5

Almacenamiento (St)	T1	T2	T3
Producto	5	0	0

Restricción de equilibrio			
T1	15	=	15
T2	20	=	20
T3	5	=	5

Restricciones de capacidad			
Capacidad de fabricar (ct)			
Producto	T1	T2	T3
Disponible	10	15	5
	<=	<=	<=
Máxima	15	15	15
Capacidad de almacenar (at)			
Producto	T1	T2	T3
Disponible	5	0	0
	<=	<=	<=
Máxima	10	10	10

Figura 18. Solución modelo 1 en Excel. Fuente: Elaboración propia.

Una de las ventajas principales de emplear la herramienta Excel, es que es sencilla de emplear y se puede leer con facilidad las soluciones y cómo se satisfacen las restricciones. Como, por ejemplo, en la restricción de equilibrio, donde se ve la restricción de igualdad o en las restricciones de capacidad, donde se puede ver en qué periodos se emplea toda la capacidad y en cuáles existe holgura (sobra capacidad).

Sin embargo, la desventaja principal de emplear este modelo es que la demanda aparece como un valor fijo en la restricción de equilibrio, y dada la condición de igualdad, el problema impone que se tenga que satisfacer la demanda de manera obligatoria (apartado 4.1). Esto supone un problema, en aquellos casos en los que no exista capacidad suficiente para fabricar o almacenar las unidades necesarias para cumplir con todas las demandas. Puesto que no existirá ninguna solución capaz de cumplir todas las restricciones del problema.

Ahora se va a resolver el mismo problema planteado anteriormente, pero reduciendo las capacidades de fabricación. Los nuevos datos del problema son:

Parámetro	$t1$	$t2$	$t3$
d_t	10	20	5
p_t	2	1	3
h_t	5	2	4
c_t	10	10	10
a_t	10	10	10

Parámetro	
s_0	5

Tabla 13. Datos del modelo 1, ejemplo 2.

Fíjese que en la solución del modelo anterior ya indicaba que en el periodo 2, en la restricción de capacidad de fabricación, no había holgura y que empleaba toda la capacidad del periodo. Por tanto, al reducir la capacidad de 15 a 10 unidades, está claro que no se va a poder satisfacer la demanda del periodo 2, que pedía 15 unidades. Y al lanzar de nuevo a resolver el problema, Excel muestra el siguiente mensaje.

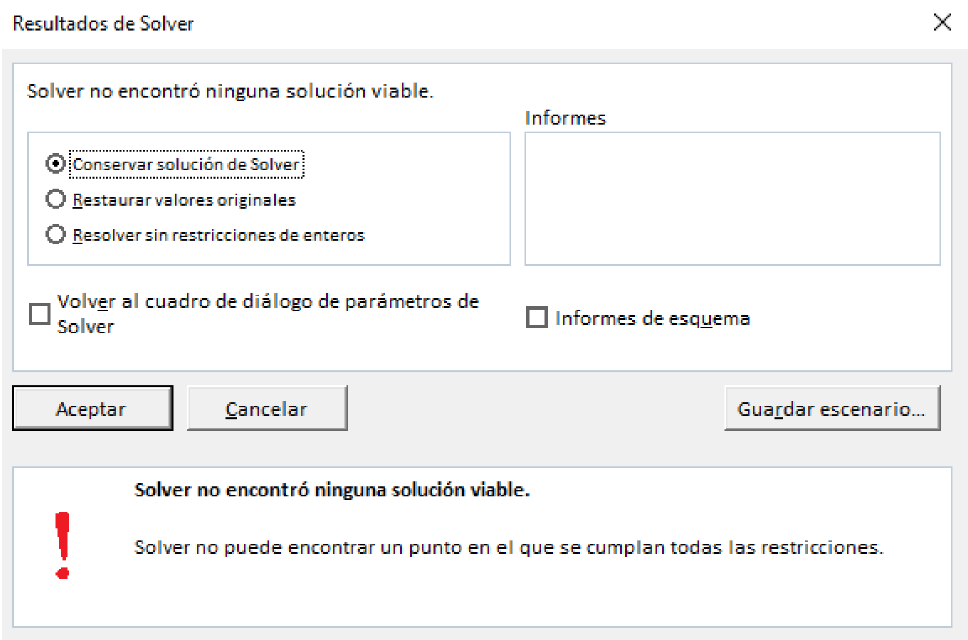

Figura 19. Solución no factible Excel.

Como se había predicho, no es posible resolver el problema porque no existe ninguna combinación de valores de las variables de decisión, capaz de cumplir todas las restricciones.

Este problema en el que se obliga a satisfacer una demanda se resuelve en el siguiente modelo (modelo 2). Donde se deja el término de la demanda libre de tomar el propio valor o el valor 0, según sea posible o no satisfacerla.

5.2 Modelo 2. Solución de Excel

Modelo con posibilidad de no satisfacer una demanda

Tal y como se ha mencionado en el párrafo anterior, en este modelo se da la posibilidad de decidir si se satisface o no una demanda, correspondiente al apartado 4.2. De este modo, aunque no se pueda cumplir con alguna demanda, sí que se pueda obtener una solución factible. Como se muestra en el siguiente ejemplo.

En este caso, se amplía el horizonte de planificación a 6 periodos, para poder observar mejor los cambios que va a ir sufriendo la solución a medida que se avance con los modelos.

Parámetro	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
d_t	15	10	20	15	10	8
c_t	15	5	5	10	5	5
a_t	5	5	5	5	5	5

Parámetro	
s_0	1

Tabla 14. Datos modelo 2.

A simple vista, se puede apreciar que, a excepción del primer periodo, las capacidades de fabricación son muy inferiores a la cantidad de producto que se demanda, por lo que, cabe esperar que no se puedan satisfacer. A pesar de ello, gracias a la nueva formulación del problema, se puede obtener una solución. Y se remarca que, sin las modificaciones pertinentes no se hubiese podido obtener ninguna solución.

Al resolver este problema con Excel, la solución obtenida es la siguiente:

Solución	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
$X_t\ (uds.)$	14	0	0	0	0	0
$S_t(uds.)$	0	0	0	0	0	0
YE_t	1	0	0	0	0	0

F.O.	
Z	1
Demandas satisfechas	1

Tabla 15. Solución modelo 2.

Los resultados obtenidos son los esperados, ya que únicamente existe capacidad suficiente para satisfacer únicamente una demanda, la del periodo 1. Para ver los resultados de la hoja de cálculo, recurrir al anexo, 1.1.

5.3 Modelo 3. Solución de Excel

Modelo con posibilidad de no satisfacer una demanda y de entregar una demanda con un periodo de adelanto

A diferencia del modelo anterior, en este se da la posibilidad de entregar una demanda con un periodo de adelanto. Con los mismos datos del problema anterior, se resuelve este problema y se analiza el efecto que tiene sobre la solución.

Parámetro	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
d_t	15	10	20	15	10	8
c_t	15	5	5	10	5	5
a_t	5	5	5	5	5	5

Parámetro	
s_0	1

Tabla 16. Datos del modelo 3.

Al dar la posibilidad de entregar con un periodo de antelación una demanda, conlleva que se puedan entregar muchas más demandas, puesto que en la restricción de equilibrio se indicará que la cantidad que se fabrique en un periodo pueda emplearse para cubrir la demanda del periodo siguiente.

Las soluciones obtenidas para este problema son:

Solución	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
$X_t\ (uds.)$	5	5	5	10	5	3
$S_t(uds.)$	5	0	5	0	5	0
YE_t	0	1	0	1	0	1

F.O.	
Z	6
Demandas satisfechas	3

Tabla 17. Solución modelo 3.

Solución		$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
E_{t_1,t_2} $(uds.)$	$t1$	0	0				
	$t2$		10	0			
	$t3$			0	0		
	$t4$				15	0	
	$t5$					0	0
	$t6$						8

Solución		$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
W_{t_1,t_2}	$t1$	0	0				
	$t2$		1	0			
	$t3$			0	0		
	$t4$				1	0	
	$t5$					0	0
	$t6$						1

Tabla 18. Continuación solución modelo 3.

A partir de la solución, se puede observar que se ha logrado pasar de satisfacer únicamente 1 demanda a satisfacer 3. En este caso, se están satisfaciendo las demandas del periodo 2,3 y 4 según indica la variable, YE_t. Al mismo tiempo, se consigue que los productos se entreguen en la fecha de entrega de la demanda y no con un periodo de adelanto. Como en el caso de la demanda 2, donde $W_{2,2}$ toma valor 1 (se sirve en la fecha de entrega) y $W_{1,2}$ toma valor 0 (no se sirve con un periodo de adelanto). Para leer las soluciones de la tabla para las variables con dos subíndices, el primero indica la fila y el segundo la columna.

Por tanto, permitir que se entregue un periodo antes de su fecha de entrega una demanda, hace que se incrementen las posibilidades de satisfacer más demandas. Para ver la hoja de cálculo asociada a este modelo, recurrir al anexo 1.2.

5.4 Modelo 4. Solución de Excel

Modelo con posibilidad de no satisfacer una demanda y de entregar una demanda en cualquier periodo anterior a su fecha de entrega

Como se ha visto en el apartado 4.4, este modelo es una extensión del modelo anterior, dando la posibilidad de entregar una demanda en un periodo cualquiera anterior a la fecha de entrega.

Para ver el cambio que produce en la solución, se resuelve este problema con los mismos datos empleados en el apartado 5.2 y 5.3. Pero, hay que añadir los nuevos parámetros definidos en la formulación del modelo como, la constante asociada a la bonificación que conlleva una demanda satisfecha y las constantes asociadas a la penalización por servir producto antes de la fecha de entrega.

Parámetro	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
d_t	15	10	20	15	10	8
c_t	15	5	5	10	5	5
a_t	5	5	5	5	5	5

Parámetro	
s_0	1

Parámetro	
f_{t-1}	1
f_{t-2}	3
f_{t-3}	5
f_{t-4}	7
f_{t-5}	9

Parámetro	
b	10

Tabla 19. Datos del modelo 4.

Dado que se pretende entregar los productos lo más cerca posible a la fecha de entrega, cuanto mayor sea la diferencia entre el periodo en el que se hace efectiva una entrega y la fecha de entrega, más negativo tiene que ser ese término en la F.O (4.1). Por este motivo las constantes asignadas van de menor a mayor (1,3,5,7 y 9), siendo el primer valor asociado a 1 periodo de adelanto y el último valor para 5 periodos de adelanto. Cabe destacar que, es preferible entregar una demanda antes de la fecha prevista, que no entregarla. Este es el motivo por el que, la constante asociada a satisfacer una demanda lleva un valor superior a la penalización debida a una entrega con 5 periodos de antelación.

Las soluciones obtenidas para este problema son:

Solución	$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
$X_t\ (uds.)$	13	5	5	9	5	5
$S_t(uds.)$	5	0	5	5	3	0
YE_t	0	1	0	1	1	1

F.O.	
Z	39
Demandas satisfechas	4

Tabla 20. Solución modelo 4.

Solución		$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
E_{t_1,t_2} $(uds.)$	$t1$	0	9	0	0	0	0
	$t2$		1	0	9	0	0
	$t3$			0	0	0	0
	$t4$				6	8	0
	$t5$					2	0
	$t6$						8

Solución		$t1$	$t2$	$t3$	$t4$	$t5$	$t6$
W_{t_1,t_2}	$t1$	0	1	0	0	0	0
	$t2$		1	0	1	0	0
	$t3$			0	0	0	0
	$t4$				1	1	0
	$t5$					1	0
	$t6$						1

Tabla 21. Continuación solución modelo 4.

A partir de la solución proporcionada por Excel, se puede observar que el número de demandas satisfechas ha aumentado respecto del problema anterior, pasa de 3 a 4 demandas satisfechas. Esto era de esperar, dado que se permite el fraccionamiento de una demanda en todos los periodos anteriores a la fecha de entrega. Esto permite ajustar mejor la capacidad disponible en cada periodo teniendo en cuenta no solo la demanda del mismo periodo, sino también la de los siguientes.

Finalmente, en al anexo 1.3, se puede apreciar, que a pesar de que no hay un gran volumen de datos, Excel ya tarda varios segundos en calcular la solución y sobre todo la hoja de cálculo empieza a ser difícil de manejar. Como se comentó en secciones anteriores, Excel permite un rápido modelado de ejemplos sencillos, al no requerir un lenguaje de programación, pero esta ventaja se convierte en un gran inconveniente conforme aumenta el tamaño y la complejidad del problema. Por este motivo, a partir del siguiente apartado los modelos se resolverán mediante la herramienta de optimización CPLEX.

5.5 Modelo 5. Solución de CPLEX

Modelo multiproducto, con posibilidad de no satisfacer una demanda y de entregar una demanda en cualquier periodo anterior a su fecha de entrega

El pasar de un problema monoproducto a uno multiproducto implica, que las variables y parámetros que habían asociadas a un único producto ahora deben extenderse a todos los tipos de productos existentes. Esto da como resultado un modelo de mayor dimensión, ya que se han multiplicado el número de variables, parámetros y restricciones por el número de productos existentes. Es decir que, si hubiese m productos distintos, y antes había 60 variables, ahora el problema pasaría a tener $60xm$ variables, lo que computacionalmente supone un coste mayor.

Para resolver este problema en CPLEX, se ha tenido que programar previamente el modelo matemático. El lenguaje de programación (estrictamente un lenguaje interpretado de scripting) es el

OPL (*Optimization Programming* Language) y que se ha tenido que trabajar durante este trabajo, ya que no se ve en el grado. Se ha empleado *User's Manual for CPLEX*, IBM y "CPLEX OPL Tutorial 01".

El código de programación es el siguiente:

```
/*INDICES*/
int T=...; // número de periodos
int I=...; //tipos de productos
range NbT =1..T;
range NbF=1..T-1;
range NbI=1..I;

/*PARAMETROS*/
int CapProd[NbT]=...;
int CapAlm[NbT]=...;
int Demanda[NbI][NbT]=...;
int Sinicial[NbI]=...;
float CosteProd[NbI][NbT]=...;
float CosteAlm[NbI][NbT]=...;
float CosteFijo[NbI][NbT]=...;
float Bonificacion=...;
float Penalizacion[NbF]=...;
int epsilon=1;

/*VARIABLES*/
dvar int+ X[NbI][NbT];
dvar int+ S[NbI][NbT];
dvar int+ E[NbI][NbT][NbT];
dvar boolean YE[NbI][NbT];
dvar boolean Y[NbI][NbT];

dexpr float IngresoEntregas=
Bonificacion*sum(i in NbI,t in NbT)YE[i][t];
dexpr int Entregas=
sum(i in NbI,t in NbT)E[i][t][t];
dexpr float AdelantoEntregas=
sum(i in NbI, t in 2..T,k in 1..t-1)(Penalizacion[t-k]*E[i][k][t]);
dexpr float CostesTotales=
sum(i in NbI,t in
NbT)(CosteProd[i][t]*X[i][t]+CosteAlm[i][t]*S[i][t]+CosteFijo[i][t]*Y[i][t
]);

/*FUNCION OBJETIVO*/
maximize
IngresoEntregas+Entregas-AdelantoEntregas-CostesTotales;

/*RESTRICCIONES*/
 subject to {
```

```
//Capacidad de produccion
 forall (t in NbT)
      Produccion: sum (i in NbI) X[i][t]<=CapProd[t];

// Capacidad de almacenamiento
 forall (t in NbT)
      Almacenamiento:sum (i in NbI)S[i][t]<=CapAlm[t];

//Continuidad par t =1
forall (i in NbI)
 Continuidad_0: Sinicial[i]+X[i][1]==S[i][1]+sum(k in 0..T-1)E[i][1][1+k];

//Continuidad para el resto de t menos t=T
 forall (i in NbI, t in NbT: 1<t<T)
   Continuidad_t: S[i][t-1]+X[i][t]==S[i][t]+sum(k in 0..T-t)E[i][t][t+k];

//Continuidad para t=T
forall (i in NbI)
      Continuidad_T: S[i][T-1]+X[i][T]==S[i][T]+E[i][T][T];

//Uso binaria para t=1
forall (i in NbI)
Binaria_YE1: E[i][1][1]==Demanda[i][1]*YE[i][1];

//Uso binaria YE para el resto de t
 forall (i in NbI,t in NbT: t>1)
 Binaria_YEt: sum(k in 1..t)E[i][k][t]==Demanda[i][t]*YE[i][t];

//Corner cases
 forall (i in NbI,t in NbT)
   YE[i][t]<=Demanda[i][t];

//Uso de la binaria Y
 forall(i in NbI, t in NbT)
   {
     epsilon*Y[i][t]<=X[i][t];
     X[i][t]<=CapProd[t]*Y[i][t];
   }
 }
```

Listado 1. Código de programación modelo 5.

Terminada la programación del modelo, se introducen unos datos de ejemplo y se calcula la solución. En este caso, se han considerado 6 periodos y 2 productos.

```
T=6;
I=2;
Demanda=[[10 15 4 5 3 0]
         [8 5 8 2 1 4]];
Sinicial=[7 6];
CosteProd=[[4 1 8 2 4 5]
           [12 7 8 2 4 1]];
CosteAlm=[[4 7 9 2 4 6]
          [8 4 7 1 2 3]];
CosteFijo=[[10 6 7 3 2 1]
           [3 7 8 2 5 7]];
CapProd=[6 9 4 3 7 5];
CapAlm=[7 2 5 2 1 4];
Bonificacion=250;
Penalizacion=[5 10 15 20 25];
```

Listado 2. Datos modelo 5.

Esta es la solución que proporciona Solver. Se trata de una solución óptima, con valor objetivo 2063€. Se consigue satisfacer 9 de las 11 demandas totales (distintas de cero) y solo hay una demanda que sirve parte de los productos con antelación.

```
// solution (optimal) with objective 2063
// Quality Incumbent solution:

YE = [[1 0 1 1 1 0]
             [1 1 0 1 1 1]];
E = [[[10 0 0 0 0 0]
                 [0 0 0 4 0 0]
                 [0 0 4 0 0 0]
                 [0 0 0 1 0 0]
                 [0 0 0 0 3 0]
                 [0 0 0 0 0 0]]
             [[8 0 0 0 0 0]
                 [0 5 0 0 0 0]
                 [0 0 0 0 0 0]
                 [0 0 0 2 0 0]
                 [0 0 0 0 1 0]
                 [0 0 0 0 0 4]]];
X = [[3 4 4 1 3 0]
             [2 5 0 2 1 4]];
S = [[0 0 0 0 0 0]
             [0 0 0 0 0 0]];
Y = [[1 1 1 1 1 0]
             [1 1 0 1 1 1]];
```

Listado 3. Solución modelo.

La ventaja que presenta CPLEX frente a Excel es la comodidad de programar el modelo y la facilidad existente para realizar modificaciones. Además de la cantidad de informacion que proporciona acerca del problema y del proceso de búsqueda de la solución.

Por ejemplo en esta imagen, CPLEX indica la cantidad de restricciones y variables existentes en el problema que se ha plantedo.

Estadística	Valor
Cplex	solution (optimal) with objective 2063
Constraints	102
Variables	120
Binary	24
Integer	96
Non-zero coefficients	243

Listado 4. Salida CPLEX estadísticas.

Y en la siguiente, muestra las iteraciones que ha realizado el algoritmo B&B hasta encontrar la solución óptima (GAP 0%) y el tiempo de cómputo, en este caso ha sido 0,05 segundos.

```
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                       1283,0000     2809,0000            118,94%
*     0+    0                       1298,0000     2809,0000            116,41%
      0     0     2101,5111    12    1298,0000     2101,5111       34    61,90%
*     0+    0                       2029,0000     2101,5111              3,57%
      0     0     2072,6667     1    2029,0000      Cuts: 31       44     2,15%
*     0     0      integral     0    2063,0000     2063,0000       46     0,00%
Elapsed time = 0,05 sec. (1,47 ticks, tree = 0,01 MB, solutions = 4)
```

Listado 5. Salida CPLEX registro del motor.

5.6 Modelo 6. Solución de CPLEX

Modelo con demanda desagregada a pedidos, multiproducto, con posibilidad de no satisfacer una demanda y de entregar una demanda en cualquier periodo anterior a su fecha de entrega

En este sexto modelo, se desagregaba la demanda a pedidos y esto hace que sea necesario identificar a qué pedido pertenece cada demanda, dado que puede haber una misma demanda de producto en varios pedidos al mismo tiempo, como se ha explicado en el apartado 4.6. El hecho de añadir ahora la idea de pedidos conlleva seguir ampliando la dimensión del modelo.

Se ha resuelto el siguiente caso en el que existen 2 pedidos, 2 producto y 3 periodos.

```
T=3;
I=2;
L=2;

CapProd=[7 9 4];
CapAlm=[5 2 5];
Demanda=[[[1 8 1][2 2 3]]
            [[8 9 1][1 5 10]]];
Sinicial=[4 3];

CosteProd=[[1 2 4]
               [2 7 5]];
CosteAlm=[[1 7 2]
              [2 4 3]];
CosteFijo=[[4 6 5]
               [2 7 5]];
Bonificacion=[200 250];
Penalizacion=[10 20];
```

Listado 6. Datos modelo 6.

Y la solución obtenida es la siguiente, con un valor de la F.O de 161 €. Podemos observar que, en este caso, el objetivo era que la variable YP_p tomase valor 1 en ambos pedidos, pero debido a las restricciones de capacidad, no es posible satisfacer todas las demandas. Únicamente se puede entregar el pedido uno. Dado que servir un pedido conlleva que se sirvan todas las demandas de este, se puede observar que todas las YE_p asociadas al pedido 1, toman valor 1. Y como el pedido 2 no se puede entregar al completo, se cancela y no se entrega ningún producto. Por lo tanto, todas las YE_p del pedido 2 toman valor 0.

```
// solution (optimal) with objective 161
YP = [1 0];
E = [[[[1 0 0]
                     [0 8 0]
                     [0 0 1]]
                 [[2 0 0]
                     [0 2 0]
                     [0 0 3]]]
             [[[0 0 0]
                     [0 0 0]
                     [0 0 0]]
                 [[0 0 0]
                     [0 0 0]
                     [0 0 0]]]];
```

Listado 7. Resultados modelo 6.

```
X = [[0 5 1]
             [1 0 3]];
S = [[3 0 0]
             [2 0 0]];

Y = [[0 1 1]
             [1 0 1]];
YE = [[[1 1 1]
                 [1 1 1]]
             [[0 0 0]
                    [0 0 0]]];
```

Listado 8. Continuación resultados modelo 6.

Para ver el listado con el modelo de CPLEX recurrir al anexo 2.1.

5.7 Modelo 7. Solución de CPLEX

Modelo multinivel, con demanda desagregada a pedidos, multiproducto, con posibilidad de no satisfacer una demanda y de entregar una demanda en cualquier periodo anterior a su fecha de entrega

Al tratarse de un problema multinivel, es necesario tener un BOM que indique la relación entre los distintos productos. En este ejemplo, existen 3 periodos, 2 productos y 2 pedidos. Y el BOM que presenta sería el que se muestra en Figura 20. Ejemplo BOM modelo . En él, se indica que para fabricar una unidad de "producto 1" serán necesarias dos unidades de "producto 2". Fabricar el "producto 1" como el "producto 2", conlleva 1 periodo cada uno. Por lo que fabricar el "producto 1" desde el nivel más bajo, tardaría 2 periodos.

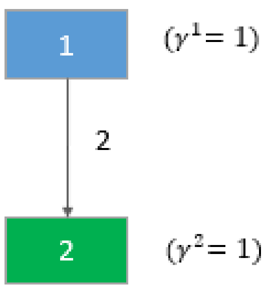

Figura 20. Ejemplo BOM modelo 7. Fuente: Elaboración propia.

Los datos del problema son los siguientes:

```
T=3;
TLead=1;
I=2;
L=2;

CapProd=[20 30 25];
CapAlm=[25 35 15];
Demanda=[[[1 2 1][1 2 1]]
         [[1 2 3][1 1 2]]];
Sinicial=[3 5];
CosteProd=[[1 1 2]
           [1 1 1]];
CosteAlm=[[1 1 2]
           [2 4 1]];
CosteFijo=[[1 2 1]
           [1 8 0]];
Bonificacion=[200 250];
Penalizacion=[10 20];
LeadTime=[1 1];
R=[[0 0]
   [2 0]];
XLead=[[1]
       [3]];
```

Listado 9. Datos modelo 7.

De estos datos, es importante destacar la incorporación del parámetro "R" que representa el $r^{i,j}$ de la formulación del modelo y el "LeadTime" de cada producto φ^{i}, esta información viene dada por el BOM presentado en la . Además, el "TLead" es un dato de entrada que corresponde al valor más grande de entre todos los φ^{i} de todos los productos, dado que, en este ejemplo, ambos productos presentan un lead-time de 1 periodo, "TLead" será igual a 1.

Asimismo, se debe añadir "XLead" como un dato de entrada, que indica la cantidad de producto que ya había lanzada antes del periodo inicial, *t*=1. Recuerde la ecuación presentada en el apartado 4.7, la restricción de equilibrio (7.6), donde el término $X^{i}_{t-\gamma_i}$, para t=1 se corresponde con "XLead".

En este caso "XLead" se interpretaría de la manera siguiente, un periodo anterior a *t*=1, habían lanzadas 1 unidad de "producto 1" y 3 unidades de "producto 2", según Listado 9. Datos modelo 7.

$$S^{i}_{t-1} + X^{i}_{t-\gamma_i} = S^{i}_{t} + \sum_{p=1}^{l} \sum_{k=0}^{k=n-t} E^{p,i}_{t,t+k} + \sum_{j\,\in D(i)} r^{i,j} \cdot X^{j}_{t} \qquad (7.6)$$
$$i = 1, \dots, m, t = 1, \dots, n$$

La solución del problema se muestra a continuación:

```
// solution (optimal) with objective 440
YP = [1
            1];
E = [[[[1 0 0]
                     [0 2 0]
                     [0 0 1]]
                 [[1 0 0]
                     [0 2 0]
                     [0 0 1]]]
             [[[1 0 0]
                     [0 2 0]
                     [0 0 8]]
                 [[1 0 0]
                     [0 1 0]
                     [0 0 9]]]]];
X = [[3 3 0]
             [9 3 0]];
S = [[2 1 0]
             [1 0 0]];
Y = [[1 1 0]
             [1 1 0]];
YE = [[[1 1 1]
                 [1 1 1]]
             [[1 1 1]
                 [1 1 1]]];
```

Listado 10. Solución modelo 7.

Se ha obtenido un valor de la función objetivo de 440 €. En este caso, es capaz de satisfacer tanto la demanda independiente (entrega todos los pedidos) como la demanda dependiente (tiene materia prima “producto 2” suficiente para poder fabricar todas las demandas de producto final “producto 1”).

5.8 Modelo 8. Solución de CPLEX

Modelo con órdenes compra, priorización de un pedido, multinivel, con demanda desagregada a pedidos, multiproducto, con posibilidad de no satisfacer una demanda y de entregar una demanda en cualquier periodo anterior a su fecha de entrega

En este modelo se añadían las órdenes de compra y la posibilidad de dar prioridad a un pedido.

Se presentan los datos del problema:

```
T=4;
TLead=3;
I=4;
L=2;
CapProd=[5 20 2 16];
CapAlm=[50 10 40 20];
Demanda=[[[1 4 3 4][7 2 5 2][3 1 4 2][3 5 2 1]]
          [[2 0 5 1][1 3 1 5][1 1 2 1][2 0 1 1]]];
Sinicial=[1 5 3 4 ];
CosteProd=[[1 1 4 1]
            [1 1 1 1]
            [1 1 1 1]
            [1 1 1 3]];
CosteAlm=[[1 1 2 1]
           [2 1 4 2]
           [2 1 1 2]
           [2 4 1 2]];
CosteFijoProd=[[5 1 1 1]
               [1 4 2 3]
               [1 2 5 1]
               [1 1 1 3]];
CosteCompra=[[4 1 1 1]
              [1 5 2 1]
              [1 2 2 1]
              [1 1 5 2]];
CosteFijoCompra=[[4 1 5 1]
                  [1 2 1 1]
                   [1 2 1 1]
                   [1 4 5 1]];
Bonificacion=[1000 1250];
Priorizacion=[2 1];
Penalizacion=[1 2 3];
LeadTimeProd=[0 0 1 3];
LeadTimeCompra=[0 0 1 1];
R=[[0 0 0 0]
   [0 0 0 0]
   [1 0 0 0]
   [2 0 0 0]];
 XLead=[[1 0 1]
         [0 0 1]
         [0 0 0]
         [0 0 0]];
 XCLead=[[1 0 0]
          [0 0 1]
          [0 0 0]
          [0 1 0]];
```

Listado 11. Datos del modelo 8.

Dado que se han añadido las órdenes de compra, de manera análoga a lo explicado en el apartado 5.7 para las órdenes de fabricación, se incluye “XCLead” y “LeadTimeCompra”. Y como nuevos datos de entrada, se incluyen los costes fijos por la compra de producto en un periodo y los costes asociados a cada unidad de producto comprada.

Resolviendo el problema se obtiene la siguiente solución (en esta imagen se muestran solo las variables con valor distinto de cero).

```
// solution (optimal) with objective 1944
YP = [1
         0];
E = [[[[1 0 0 0]
                     [0 4 0 2]
                     [0 0 3 0]
                     [0 0 0 2]]
                 [[7 0 0 0]
                     [0 2 0 0]
                     [0 0 5 0]
                     [0 0 0 2]]
                 [[3 0 0 0]
                     [0 1 0 0]
                     [0 0 4 0]
                     [0 0 0 2]]
                 [[3 0 1 0]
                     [0 5 1 0]
                     [0 0 0 0]
                     [0 0 0 1]]]
            [...]
```

Listado 12. Solución modelo 8.

En el listado anterior, se ha omitido el resultado de la variable E_{t_1,t_2} del pedido 2, dado que no es posible entregar este pedido y por tanto todas las variables E_{t_1,t_2}asociadas a este toman valor 0.

En el Listado 12 se puede observar que únicamente se ha podido servir el pedido 1. Esto es debido a que se ha priorizado la entrega del pedido 1 frente al 2, poniendo la constante de priorización v^p del pedido 1 con valor dos, mientras que la del pedido 2 con valor uno.

Por último, en cuanto a las órdenes de compra, se realiza la compra de productos debido a que no existe capacidad suficiente en algunos periodos para fabricar todos productos demandados (tanto de demanda dependiente como independiente) en el pedido 1.

```
X = [[0 0 0 0]
             [0 5 2 0]
             [0 8 0 0]
             [5 0 0 0]];
S = [[0 3 0 0]
             [0 3 0 0]
             [0 0 4 0]
             [0 0 0 0]];
Y = [[0 0 0 0]
             [0 1 1 0]
             [0 1 0 0]
             [1 0 0 0]];
XC = [[0 9 0 2]
             [2 0 0 2]
             [10 0 0 0]
             [24 0 0 0]];
YC = [[0 1 0 1]
             [1 0 0 1]
             [1 0 0 0]
                [1 0 0 0]];
```

Listado 13. Continuación solución modelo 8.

5.9 Modelo 9. Solución de CPLEX

Modelo sin *splitting*, con órdenes compra, priorización de un pedido, multinivel, con demanda desagregada a pedidos, multiproducto, con posibilidad de no satisfacer una demanda y de entregar una demanda en cualquier periodo anterior a su fecha de entrega

Tras ir desarrollando pasa a paso cada uno de los modelos anteriores, se llega a este último modelo que aúna todas las especificaciones y funcionalidad del algoritmo MRP.

La diferencia de este modelo con respecto al anterior es únicamente la condición de que no se produzca *splitting*. Por lo tanto, ahora se va a resolver este problema con los mismos datos del apartado anterior, para ver el efecto que tiene el que haya o no *splitting*.

De manera intuitiva, el hecho de que no se permita realizar el fraccionamiento de entregas en las demandas, hace que el modelo sea menos flexible. Ya que, reduce el número de combinaciones posibles para realizar la entrega de una demanda.

Para el ejemplo que nos ocupa, la nueva solución obtenida es la siguiente:

```
   // solution (optimal) with objective 1943
YP = [1
         0];
E = [[[[1 0 0 0]
                     [0 4 0 0]
                     [0 0 3 0]
                     [0 0 0 4]]
                 [[7 0 0 0]
                     [0 2 0 0]
                     [0 0 5 0]
                     [0 0 0 2]]
                 [[3 0 0 0]
                     [0 1 0 0]
                     [0 0 4 0]
                     [0 0 0 2]]
                 [[3 0 0 0]
                     [0 5 0 0]
                     [0 0 2 0]
                    [0 0 0 1]]]
   [...]
```

Listado 14. Solución modelo 9.

Tal y como se había predicho, la F.O. ha empeorado y ahora se puede observar que todas las demandas se entregan en un solo periodo. Sin embargo, no se ha apreciado una disminución significativa de la F.O. (ha disminuido un 0,05%), dado que, en los resultados del apartado anterior, se observa que únicamente había una demanda que necesitaba servir en más de un periodo.

En resumen, el no permitir que se realice *splitting* de una demanda, por lo general reduce la eficiencia de la función objetivo (empeora el valor de la F.O.), puesto que añade más restricciones al problema. Aunque, según el problema que se resuelva, esta condición puede ser fundamental, puesto que garantiza la trazabilidad de fabricación de los productos.

Para ver el código de programación ir al anexo 2.2.

5.10 Comparativa del modelo matemático con el algoritmo MRP

Para finalizar este capítulo 5, se va a realizar una comparativa entre la solución que proporciona el modelo matemático final en CPLEX (modelo 9) y la que proporcionaría el algoritmo MRP de GeInfor.

Dado que GeInfor se encuentra actualmente desarrollando el algoritmo y por tanto, no se dispone del mismo para resolver el problema, se va a proponer un ejemplo muy simple que se pueda resolver de forma manual, siguiendo los pasos exactos del algoritmo descritos en el apartado 2.2 pero realizados a mano.

Los datos del problema son los siguientes:

```
T=3;
TLead=1;
I=2;
L=1;
CapProd=[20 20 20];
CapAlm=[15 15 15];
Demanda=[[[2 3 1][1 1 0]]];
Sinicial=[0 8];
CosteProd=[[1 1 1]
           [1 1 1]];
CosteAlm=[[1 1 1]
           [3 3 3]];
CosteFijoProd=[[4 4 4]
               [1 1 1]];
CosteCompra=[[7 7 7]
             [5 5 5]];
CosteFijoCompra=[[6 6 6]
                 [5 5 5]];
Bonificacion=[1500];
Priorizacion=[1];
Penalizacion=[10 10];
LeadTimeProd=[0 1];
LeadTimeCompra=[0 0];
R=[[0 0]
   [1 0]];
 XLead=[[0]
        [0]];
 XCLead=[[0]
         [0]];
```

Listado 15. Datos ejemplo comparativa entre modelo matemático y algoritmo MRPc.

Como se ha comentado anteriormente, se necesita un ejemplo pequeño para poder calcularlo a mano, pero que permita visualizar las diferencias existentes entre la solución del modelo y la del algoritmo “manual”. Se va a considerar que existen únicamente 1 pedido, 3 periodos de planificación y 2 productos distintos.

La solución que facilita el Solver de CPLEX se muestra a continuación en el Listado 16. Se ha obtenido un valor de la F.O. de 1490€, donde se ha podido entregar el único pedido que había, YP^p toma valor 1. En cuanto a las entregas, y según el valor de las variables $E^{p,i}_{t_1,t_2}$, se entrega exactamente la cantidad de producto demandada y se efectúa en la misma fecha de entrega. En este caso no se sirve ninguna demanda con adelanto.

```
// solution (optimal) with objective 1490
YP = [1];
E = [[[[2 0 0]
                     [0 3 0]
                     [0 0 1]]
                 [[1 0 0]
                     [0 1 0]
                     [0 0 0]]]];
X = [[6 0 0]
             [0 0 0]];
S = [[4 1 0]
             [1 0 0]];
Y = [[1 0 0]
             [0 0 0]];
XC = [[0 0 0]
             [0 0 0]];
YC = [[0 0 0]
             [0 0 0]];
W = [[[[1 0 0]
                     [0 1 0]
                     [0 0 1]]
                 [[1 0 0]
                     [0 1 0]
                        [0 0 0]]]];
```

Listado 16. Solución ejemplo comparativo CPLEX.

Se puede observar que la salida de CPLEX indica explícitamente que la solución obtenida es óptima. En este caso se recuerda que se buscaba minimizar los costes y al mismo tiempo trataba de entregar las demandas lo más cerca posible a su fecha de entrega. Gracias a la manera en la que se ha definido la F.O., CPLEX permite saber el valor que toma cada uno de los términos dentro de la F.O.

Expresiones de decisión (4)	
AdelantoEntregas	0
CostesTotales	18
Entregas	8
IngresoPedidos	1500

Listado 17. Salida CPLEX Ventana examinador de problemas.

CPLEX indica que no se ha entregado ninguna demanda con adelanto y que se han entregado en total 8 productos. Además, muestra que los costes totales asociados a las entregas (fijos y variables) son 18€.

A continuación, la solución que proporcionaría el algoritmo MRP de GeInfor. Las variables que no se muestras en las siguientes tablas es debido a que su valor es cero.

Solución		$t1$	$t2$	$t3$
$X_t^i(uds.)$	$i1$	2	3	1
	$i2$	0	0	0

Solución		$t1$	$t2$	$t3$
$S_t^i(uds.)$	$i1$	0	0	0
	$i2$	5	1	0

Solución		$t1$	$t2$	$t3$
$Y_t'^i(uds.)$	$i1$	1	1	1
	$i2$	0	0	0

Solución	
$Costes$ $totales$ (€)	36

Solución		$t1$	$t2$	$t3$
$E_{t_1,t_2}^{p,i}$ $(uds.)$	$t1$	2	0	0
$i1$	$t2$		3	0
	$t3$			1
$E_{t_1,t_2}^{p,i}$ $(uds.)$	$t1$	1	0	0
$i2$	$t2$		1	0
	$t3$			0

Solución		$t1$	$t2$	$t3$
$W_{t_1,t_2}^{p,i}$	$t1$	1	0	0
$i1$	$t2$		1	0
	$t3$			1
$W_{t_1,t_2}^{p,i}$	$t1$	1	0	0
$i2$	$t2$		1	0
	$t3$			0

Tabla 22. Solución algoritmo MRPc manual.

En primer lugar, en cuanto a similitudes, se puede observar que las variables $E_{t_1,t_2}^{p,i}$y $W_{t_1,t_2}^{p,i}$ toman exactamente los mismos valores, dado que hay capacidad suficiente de fabricación y almacenamiento, para poder hacer efectiva la entrega justo en la fecha de entrega.

Sin embargo, se puede observar que las variables de producción X_t^i y almacenamiento S_t^i no presentan el mismo resultado. Por un lado, la solución que ofrece Solver indica que fabrica únicamente en un periodo (en $t = 1$) todas las unidades necesarias (6 unidades de producto 1), y presenta 4 y 1 unidades de stock (de producto 1) en el periodo 1 y 2, respectivamente. Esto se debe a varias razones: la primera es que los precios de fabricación son iguales en todos los periodos, por lo que no hay ningún periodo en el que sea más económico producir; la segunda es que los costes fijos de fabricación son superiores a los costes variables, por lo que es preferible fabricar en el menor número de periodos posibles para reducir los costes fijos; y la tercera es que los costes asociados a almacenamiento no son superiores a los costes fijos, y en este caso dado que son pocas unidades las que se fabrican y es poco el tiempo que se tiene que retener un producto hasta entregarlo en su fecha de entrega, Solver decide producir muchas unidades en un solo periodo y almacenarlas hasta su fecha de entrega antes que fabricar en varios periodos. Recordemos que el Solver proporciona la solución óptima (mejor posible valor de la función objetivo, que es una función de costes totales).

Por otro lado, la solución que proporciona el algoritmo es producir 2,3 y 1 unidades de producto 1 en cada uno de los periodos, es decir que fabrica exactamente las unidades que se demandan en cada periodo. Al producir, directamente para el periodo que se requiere y no haber stock inicial, no habrá stock en ninguno de los periodos del producto 1, pero entonces se tendrá que almacenar la materia prima (producto 2) hasta que se consuma en el periodo en el que se fabrique el producto final (producto 1). Por lo que habrá menos gastos de almacenamiento del producto 1 pero más de almacenamiento de producto 2 y al producir en distintos periodos, los costes fijos serán superiores.

La diferencia fundamental reside en el hecho de que el algoritmo, calcula el primer periodo disponible (con capacidad suficiente) en el que es posible fabricar, sin tener en consideración los costes, ni los variables ni los fijos. Mientras que el modelo matemático, sí que busca el momento óptimo para lanzar las órdenes de fabricación y compra (en el caso que las hubiera) en función de los costes que lleve asociada la entrega. Por este motivo, el algoritmo calcula una solución aceptable con un coste total de 36€ mientras que el modelo calcula la solución óptima con un coste total menor, 21€.

Por tanto, se puede afirmar que el modelo matemático siempre va a proporcionar la solución óptima, a diferencia del algoritmo heurístico que ofrece una solución aceptable y que difícilmente será la óptima. En ejemplos de mayor tamaño, con estructuras de costes más complejas, múltiples productos y listas de materiales, es de esperar que la solución óptima se vaya alejando más de la solución heurística que irá empeorando, comparativamente hablando, conforme aumente la dificultad del problema.

Por todo lo anterior mencionado, queda claro que el modelo matemático propuesto (modelo 9) es preferible frente a un algoritmo heurístico (algoritmo MRPc de GeInfor) en cuanto al tipo de solución que ofrece. No obstante, las empresas generalmente trabajan con volúmenes de datos muy superiores a los presentados, y en estos casos resolver el problema de la planificación con un modelo, podría resultar costoso desde el punto de vista computacional, pero cabe plantearse la pregunta de hasta dónde es viable usar un Solver que proporciona soluciones óptimas, frente a una heurística que no lo hace.

Para finalizar este capítulo y ofrecer una perspectiva más realista de lo que supondría resolver un problema de mayor tamaño, se ha resuelto un problema que incluía 20 productos, 10 periodos y 10 pedidos. Al resolver el problema en CPLEX, la solución obtenida sigue siendo la óptima, puesto que se obtiene un % de GAP igual a 0. En particular, este problema presenta aproximadamente 75000 restricciones y 41000 variables de decisión y ha sido resuelto en apenas 41,69 segundos en CPLEX. El problema podría ser de dimensiones muy superiores, y cabría plantearse en estos casos si la solución del modelo matemático es mejor o peor y en qué medida, respecto de la que proporcionaría el algoritmo heurístico.

6 CONCLUSIONES

Hoy en día, el problema de la planificación de la producción resulta ser uno de los más frecuentes a los que se enfrentan las empresas del sector de la producción. Si añadimos aprovisionamientos a través de compras, esto es extensibles a empresas distribuidoras que no tienen actividad propiamente manufacturera. Estas empresas deben elaborar una planificación detallada de las cantidades de producto que deben fabricar y/o comprar, para alcanzar los objetivos económicos que se propongan. Sin embargo, elaborar dicha planificación para un horizonte temporal a corto-medio plazo, puede ser complejo.

Por este motivo se emplean técnicas como el MRP o los modelos matemáticos, que ofrecen una solución al problema de la planificación y proporcionan la información necesaria para tomar mejores decisiones en este aspecto.

En un primer instante se ha estudiado el MRP estándar, un método que permitía determinar el momento y la cantidad en los que se debían lanzar las órdenes de fabricación y compra. Pero, el inconveniente que este presentaba era, que no tenía en cuenta las restricciones de capacidad, cosa que resulta inaceptable pues en la actualidad la mayoría de los procesos sí que presentan capacidades limitadas, como pueden ser la capacidad de fabricación o la capacidad de almacenamiento.

Esta es la razón por la que se crean algoritmos heurísticos, como el MRPc que está desarrollando GeInfor, cuya mejora es la incorporación de dichas restricciones de capacidad. Sin embargo, la desventaja que presenta resolver un problema con un algoritmo heurístico es, que en muchos casos no proporciona la solución óptima.

Esto contrasta, a los modelos matemáticos, que sí son capaces de proporcionar una solución óptima. Por ello, en este TFG se ha diseñado un modelo matemático que replicase el comportamiento y la funcionalidad que presenta el MRPc. Para formular dicho modelo, se ha empleado como punto de partida el modelo estándar de la planificación de la producción. Sobre este mismo, se han ido realizando distintas modificaciones para ir incorporando las especificaciones del algoritmo, dando lugar a la formulación de nueve modelos distintos. Donde el paso clave, ha sido decidir qué variables debían crearse en cada caso, para poder reproducir fielmente las características del algoritmo mediante las ecuaciones matemáticas. Otro aspecto, que ha tenido un papel fundamental a la hora de formular los modelos, ha sido el uso de las variables binarias, que permitían que los parámetros tuviesen un grado de libertad y de este modo, dejasen de tomar un valor fijo en las restricciones.

El punto crítico para llevar a cabo la formulación del modelo ha sido identificar de qué manera, podía cuantificarse el tiempo de adelanto con el que se servía una demanda respecto a su fecha de entrega y desagregar la demanda a pedidos. Además de, diseñar una función objetivo que satisficiese dos objetivos, que eran: maximizar el número de pedidos entregados y minimizar la diferencia entre el periodo en el que se hace efectiva una entrega y el periodo de fecha de entrega. Teniendo en cuenta, que era preferible entregar un pedido con adelanto, que no entregarlo por completo. Y en el caso de que no fuese posible satisfacer todas las demandas de un pedido, no se entregase ningún producto de este.

Posteriormente, para verificar que los modelos estaban correctamente formulados, ha sido necesario resolver cada uno de ellos con distintos escenarios, para comprobar que las soluciones obtenidas evolucionaban de manera coherente con las características del algoritmo que se iban incluyendo. Para ello, se han empleado dos herramientas Excel y CPLEX, según la dimensión que presentase el modelo. Los modelos simples (modelos 1-4) se han resuelto con Excel, mientras que los de complejidad creciente (modelos 5-9) se han resuelto con CPLEX. Por una parte, Excel es una herramienta que, para problemas de menor tamaño, sí presenta la facilidad de modelizar un problema matemático en una hoja de cálculo. Sin embargo, a medida que el modelo aumentaba de tamaño, hacía cada vez más compleja su maniobrabilidad. Por este motivo, a partir del modelo 5, se emplea la herramienta CPLEX, que presenta un lenguaje de modelización orientado a resolver problemas matemáticos. Por tanto, facilita la adición de nuevas variables y restricciones al problema. Además, este software proporciona información del proceso de búsqueda de solución, y en el caso de no encontrar una solución óptima, proporciona una solución relajada junto con el GAP que presenta. Desgraciadamente, el uso de una herramienta profesional como CPLEX ha requerido el aprendizaje de un lenguaje de programación específico en este ámbito, como es el OPL, además de un considerable tiempo dedicado al desarrollo, depuración y prueba de los modelos desarrollados.

Finalmente se ha mostrado un ejemplo en el que se comparaba la solución que proporciona CPLEX con la que proporcionaría el algoritmo de GeInfor. A partir del mismo, se ha podido verificar que un modelo matemático ofrece una solución óptima mientras que el algoritmo proporciona una solución aceptable. Cabe preguntarse hasta dónde pueden llegar, computacionalmente hablando, los modelos matemáticos y los solvers profesionales actuales. ¿Son las máximas dimensiones admisibles computacionalmente compatibles con los problemas reales?

En conclusión, en este TFG se ha observado que es posible formular modelos matemáticos complejos, empleando únicamente técnicas de programación lineal entera mixta, dónde es fundamental, realizar un estudio previo del problema que se pretende modelizar, identificar correctamente las variables de decisión que deben intervenir y verificar con distintos escenarios la formulación del modelo. Con el fin de intentar obtener soluciones óptimas al problema de la planificación de la producción.

7 LÍNEAS DE FUTURO

Ningún trabajo termina nunca, y menos un problema real como es el caso que nos ocupa. Se plantean unas líneas posibles de trabajo que podrían continuarse en un Trabajo de Fin de Máster (TFM).

- Se podría incrementar la dimensión del modelo, incluyendo un mayor número de productos y pedidos, ampliando el BOM y el número de periodos en el horizonte de planificación.
- También se podría incluir que existiese un stock de seguridad mínimo.
- Incluso, se podría formular distintos modelos empleando otra técnica de lotificación como, por ejemplo, la de capacidad constante o la de periodo constante.
- En el modelo formulado, únicamente se permite entregar con adelanto. Cabría estudiar un modelo con la posibilidad de entregar los productos con retraso, *backlogging*, es decir en periodos posteriores a la fecha de entrega de la demanda.
- Por último, se podría estudiar si es posible la implementación del modelo matemático en una empresa para resolver problemas de tamaño real. Y en qué medida difiere la solución del modelo matemático de modelo MRP de GeInfor. Esto último no ha sido posible en el marco del presente TFG dado que GeInfor aún se encuentra desarrollando su algoritmo MRPc.

8 BIBLIOGRAFÍA

Andrés Romano, C. (2020). Asignatura de Producción. *Introducción a la Planificación de Requerimiento de Materiales*. Universidad de Valencia.

Corporation, International Business Machines. (2020). *User's Manual for CPLEX: IBM*. <https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-users-manual > [Consulta: 4 de junio de 2021]

Gass, S. I., & Assad, A. A. (2006). *An Annotated Timeline of Operations Research. An Informal History*. Boston: Springer.

Gestión Empresarial Infromatizada. *Geinfor, ERP Industria 4.0*. <https://geinfor.com> [Consulta: 8 de marzo de 2021]

Gultkein, Hakan, "*CPLEX OPL Tutorial 01*". *YouTube* < https://youtu.be/70HH-GNR9uM> [Consulta: 3 de junio de 2021]

Hillier, F. S., & Lieberman, G. J. (2015). *Introduction to operations research*. New York: McGraw-Hill.

International Business Machines Corporation. *CPLEX Optimizer*. <https://www.ibm.com/es-es/analytics/cplex-optimizer > [Consulta: 18 de junio de 2021]

Land, A. H., & Doig, A. G. (1960). "An Automatic Method of Solving Discrete Programming Problems" en *Econometrica*, vol. 28, No. 3, p. 497-520.

Lova Ruiz, A., & Tormos Juan, P. (2003). *Investigación operativa para ingenieros*. Valencia: Universidad Politécnica de Valencia.

Maroto Álvarez, C., Alcaraz Soria, J., & Ruiz García, R. (2003). *Operations Research. Modeling and Optimization Techniques*. Valencia: Universidad Politécnica de Valencia.

Nash, S. G. (1990). *A History of Scientific Computing*. New York: Association of Computing Machinery, ACM.

Pochet, Y., & Wolsey, L. A. (2006). *Production Planning by Mixed Integer Programming*. New York: Springer.

Ptak, C., & Smith, C. (2011). *Orlicky's Material Requirements Planning*. New York: McGraw-Hill Education.

Taha, H. A. (2017). *Investigación de Operaciones*. México: Pearson

ANEXOS

1 HOJAS DE CÁLCULO

1.1 Modelo 2. Hoja de cálculo

Modelo 2. Posibilidad de no satisfacer una demanda

Demanda (dt)	T1	T2	T3	T4	T5	T6
Producto	15	10	20	15	10	8

F.O	
Z	1

Stock iniciak (S0)	
Producto	1

Demandas satisfechas	
	1

Fabricacion (Xt)	T1	T2	T3	T4	T5	T6
Producto	14	0	0	0	0	0

Almacenamiento (St)	T1	T2	T3	T4	T5	T6
Producto	0	0	0	0	0	0

Binaria entrega (Yt)	T1	T2	T3	T4	T5	T6
Producto	1	0	0	0	0	0

Cap. Fabricacion (ct)	T1	T2	T3	T4	T5	T6
Disponible	14	0	0	0	0	0
	<=	<=	<=	<=	<=	<=
Máxima	15	5	5	10	5	5

Cap. Almacenamiento (at)	T1	T2	T3	T4	T5	T6
Disponible	0	0	0	0	0	0
	<=	<=	<=	<=	<=	<=
Máxima	5	5	5	5	5	5

Restricción equilibrio			
T1	15	=	15
T2	0	=	0
T3	0	=	0
T4	0	=	0
T5	0	=	0
T6	0	=	0

Uso binaria YEt			
T1	14	<=	15
T2	0	<=	0
T3	0	<=	0
T4	0	<=	0
T5	0	<=	0
T6	0	<=	0

Figura 21. Hoja de cálculo modelo 2. Fuente: Elaboración propia.

1.2 Modelo 3. Hoja de cálculo

Modelo 3.Posibilidad a no satisfacer una demanda y adelantar un periodo la entrega

Demanda (dt)	T1	T2	T3	T4	T5	T6
Producto	15	10	20	15	10	8

F.O	
Z	6

Stock iniciak (S0)	
Producto	0

Demandas satisfechas	
	3

Fabricación (Xt)	T1	T2	T3	T4	T5	T6
Producto	5	5	5	10	5	3

Almacenamiento (St)	T1	T2	T3	T4	T5	T6
Producto	5	0	5	0	5	0

Binaria entrega (YEt)	T1	T2	T3	T4	T5	T6
Producto	0	1	0	1	0	1

Producto entregado (Et1,t2)	T1	T2	T3	T4	T5	T6
T1	0	0				
T2		10	0			
T3			0	0		
T4				15	0	
T5					0	0
T6						8

Binaria entrega de producto (Wt1,t2)	T1	T2	T3	T4	T5	T6
T1	0	0				
T2		1	0			
T3			0	0		
T4				1	0	
T5					0	0
T6						1

Cap. Fabricacion (ct)	T1	T2	T3	T4	T5	T6
Disponible	5	5	5	10	5	3
	<=	<=	<=	<=	<=	<=
Máxima	15	5	5	10	5	5
			3			

Cap. Almacenamiento (at)	T1	T2	T3	T4	T5	T6
Disponible	5	0	5	0	5	0
	<=	<=	<=	<=	<=	<=
Máxima	5	5	5	5	5	5

Restricción equilibrio			
T1	5	=	5
T2	10	=	10
T3	5	=	5
T4	15	=	15
T5	5	=	5
T6	8	=	8

Uso binaria YEt			
T1	0	=	0
T2	10	=	10
T3	0	=	0
T4	15	=	15
T5	0	=	0
T6	8	=	8

Uso binaria Wt,t					
T1	0	<=	0	<=	0
T2	1	<=	10	<=	10
T3	0	<=	0	<=	0
T4	1	<=	15	<=	15
T5	0	<=	0	<=	0
T6	1	<=	8	<=	8
Uso binaria Wt-1,t					
T1					
T2	0	<=	0	<=	0
T3	0	<=	0	<=	0
T4	0	<=	0	<=	0
T5	0	<=	0	<=	0
T6	0	<=	0	<=	0

Figura 22. Hoja de cálculo modelo 3. Fuente: Elaboración propia.

1.3 Modelo 4. Hoja de cálculo

Modelo 4.Posibilidad a no satisfacer una demanda y adelantar numero indefinido de periodo la entrega

Demanda (dt)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Producto	15	10	20	15	10	8

F.O	
Z	39

Stock iniciak (S0)	
Producto	1

Entregas completas	
	4

Fabricacion (Xt)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Producto	13	5	5	9	5	5

Coeficientes FO	
Entrega	10
En t-1	1
En t-2	3
En t-3	5
En t-4	7
En t-5	9

Almacenamiento (St1)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Producto	5	0	5	5	3	0

Binaria entrega (Yt2)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Producto	0	1	0	1	1	1

Producto entregado (Et1,t2)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
T1	0	9	0	0	0	0
T2		1	0	9	0	0
T3			0	0	0	0
T4				6	3	0
T5					7	0
T6						8

Binaria entrega producto(Wt1,t2)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
T1	0	1	0	0	0	0
T2		1	0	1	0	0
T3			0	0	0	0
T4				1	1	0
T5					1	0
T6						1

Cap. Fabricación (ct)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Disponible	13	5	5	9	5	5
	<=	<=	<=	<=	<=	<=
Máxima	15	5	5	10	5	5

Cap. Almacenamiento (at)	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Disponible	5	0	5	5	3	0
	<=	<=	<=	<=	<=	<=
Máxima	5	5	5	5	5	5

Restricción equilibrio			
T1	14	=	14
T2	10	=	10
T3	5	=	5
T4	14	=	14
T5	10	=	10
T6	8	=	8

Uso binaria YEt			
T1	0	=	0
T2	10	=	10
T3	0	=	0
T4	15	=	15
T5	10	=	10
T6	8	=	8

Figura 23. Hoja de cálculo modelo 4. Fuente: Elaboración propia.

Uso binaria Wt1,t2 (Wt,t)					
T1	0	<=	0	<=	0
T2	1	<=	1	<=	10
T3	0	<=	0	<=	0
T4	1	<=	6	<=	15
T5	1	<=	7	<=	10
T6	1	<=	8	<=	8
Uso binaria Wt1,t2 (Wt-1,t)					
T1					
T2	1	<=	9	<=	10
T3	0	<=	0	<=	0
T4	0	<=	0	<=	0
T5	1	<=	3	<=	10
T6	0	<=	0	<=	0
Uso binaria Wt1,t2 (Wt-2,t)					
T1					
T2					
T3	0	<=	0	<=	0
T4	1	<=	9	<=	15
T5	0	<=	0	<=	0
T6	0	<=	0	<=	0
Uso binaria Wt1,t2 (Wt-3,t)					
T1					
T2					
T3					
T4	0	<=	0	<=	0
T5	0	<=	0	<=	0
T6	0	<=	0	<=	0
Uso binaria Wt1,t2 (Wt-4,t)					
T1					
T2					
T3					
T4					
T5	0	<=	0	<=	0
T6	0	<=	0	<=	0
Uso binaria Wt1,t2 (Wt-5,t)					
T1					
T2					
T3					
T4					
T5					
T6	0	<=	0	<=	0

Figura 24. Continuación hoja de cálculo modelo 4. Fuente: Elaboración propia.

2 CÓDIGOS DE PROGRAMACIÓN

2.1 Modelo 6. Código CPLEX

```
/*INDICES*/
int T=...; // número de periodos
int I=...; //tipos de productos
int L=...; // número de pedidos
range NbT =1..T;
range NbF=1..T-1; //rango adelanto
range NbI=1..I; //rango productos
range NbP=1..L; //rango pedidos

/*PARAMETROS*/
int CapProd[NbT]=...;
int CapAlm[NbT]=...;
int Demanda[NbP][NbI][NbT]=...;
int Sinicial[NbI]=...;
float CosteProd[NbI][NbT]=...;
float CosteAlm[NbI][NbT]=...;
float CosteFijo[NbI][NbT]=...;
float Bonificacion[NbP]=...;
float Penalizacion[NbF]=...;
int DenP[NbP];
int epsilon=1;

  execute {
  for (var p in NbP){
      for (var i in NbI){
        for (var t in NbT) {
            if (Demanda[p][i][t]!=0)
                  DenP[p]++;} } }
  }

/*VARIABLES*/
 dvar int+ X[NbI][NbT];
 dvar int+ S[NbI][NbT];
 dvar int+ E[NbP][NbI][NbT][NbT];
 dvar boolean YE[NbP][NbI][NbT];
 dvar boolean YP[NbP];
 dvar boolean Y[NbI][NbT];

dexpr float IngresoPedidos=
      sum(p in NbP)Bonificacion[p]*YP[p];
dexpr int Entregas=
      sum(p in NbP, i in NbI,t in NbT)E[p][i][t][t];
dexpr float AdelantoEntrega =
      sum(p in NbP, i in NbI, t in 2..T,k in 1..t-1)(Penalizacion[t-
k]*E[p][i][k][t]);
```

```
dexpr float CostesTotales=
      sum(i in NbI,t in NbT)(CosteProd[i][t]*X[i][t]+CosteAlm[i][t]*S[i][t]
+CosteFijo[i][t]*Y[i][t]);

 /*FUNCION OBJETIVO*/
 maximize
      IngresoPedidos+ Entregas-AdelantoEntrega-CostesTotales;

 /*RESTRICCIONES*/
 subject to {

//Capacidad de produccion
 forall (t in NbT)
      Produccion: sum (i in NbI) X[i][t]<=CapProd[t];

// Capacidad de almacenamiento
 forall (t in NbT)
      Almacenamiento:sum (i in NbI)S[i][t]<=CapAlm[t];
//Continuidad par t =1
forall (i in NbI)
 Continuidad_0: Sinicial[i]+X[i][1]==S[i][1]+sum(p in NbP,k in 0..T-1)
E[p][i][1][1+k];

//Continuidad para el resto de t menos t=T
 forall (i in NbI, t in NbT: 1<t<T)
   Continuidad: S[i][t-1]+X[i][t]==S[i][t]+sum(p in NbP,k in 0..T-t)
E[p][i][t][t+k];

//Continuidad para t=T
forall (i in NbI)
Continuidad_T: S[i][T-1]+X[i][T]==S[i][T]+sum(p in NbP)E[p][i][T][T];

//Uso binaria YE para t=1
forall (p in NbP, i in NbI)
Binaria_YE1: E[p][i][1][1]==Demanda[p][i][1]*YE[p][i][1];

//Uso binaria YE para el resto de t
 forall (p in NbP, i in NbI,t in NbT: t>1)
 Binaria_YEt: sum(k in 1..t)E[p][i][k][t]==Demanda[p][i][t]*YE[p][i][t];

//Uso de la binaria Y
 forall(i in NbI, t in NbT)
   { epsilon*Y[i][t]<=X[i][t];
     X[i][t]<= CapProd[t]*Y[i][t];}

//Corner cases
 forall (p in NbP, i in NbI,t in NbT)
   YE[p][i][t]<=Demanda[p][i][t];

//Entrega completa de un pedido
 forall (p in NbP)
   DenP[p]*YP[p]==sum(i in NbI, t in NbT)YE[p][i][t]; }
```

Listado 18. Código de programación modelo 6.

2.2 Modelo 9. Código CPLEX

```
/*INDICES*/
int T=...;
int TLead=...;
int I=...;
int L=...;

range NbT =1..T;
range NbTLead=1..TLead;
range NbF=1..T-1;
range NbI=1..I;
range NbP=1..L;

/*PARAMETROS*/
int CapProd[NbT]=...;
int CapAlm[NbT]=...;
int Demanda[NbP][NbI][NbT]=...;
int Sinicial[NbI]=...;
float CosteProd[NbI][NbT]=...;
float CosteAlm[NbI][NbT]=...;
float CosteFijoProd[NbI][NbT]=...;
float CosteCompra[NbI][NbT]=...;
float CosteFijoCompra[NbI][NbT]=...;
float Bonificacion[NbP]=...;
float Penalizacion[NbF]=...;
float Priorizacion[NbP]=...;
int DenP[NbP];
int LeadTimeProd[NbI]=...;
int LeadTimeCompra[NbI]=...;
int R[NbI][NbI]=...;
int XLead[NbI][NbTLead]=...;
int XCLead[NbI][NbTLead]=...;
int epsilon=1;
int alfa=100000;

/*VARIABLES*/
 dvar int+ X[NbI][NbT];
 dvar int+ S[NbI][NbT];
 dvar int+ E[NbP][NbI][NbT][NbT];
 dvar int+ XC[NbI][NbT];
 dvar boolean W[NbP][NbI][NbT][NbT];
 dvar boolean YP[NbP];
 dvar boolean Y[NbI][NbT];
 dvar boolean YC[NbI][NbT];

execute {
  for (var p in NbP){
      for (var i in NbI){
        for (var t in NbT) {
            if (Demanda[p][i][t]!=0)
```

```
                  DenP[p]++;}}}
                  }

 dexpr float IngresoPedidos=
      sum(p in NbP)Priorizacion[p]*Bonificacion[p]*YP[p];
 dexpr int Entregas=
      sum(p in NbP, i in NbI,t in NbT)E[p][i][t][t];
 dexpr float AdelantoEntregas=
      sum(p in NbP, i in NbI, t in 2..T,k in 1..t-1)       (Penalizacion[t-
k]*E[p][i][k][t]);
 dexpr float CostesTotales=
      sum(i in NbI,t in NbT)
(CosteProd[i][t]*X[i][t]+CosteAlm[i][t]*S[i][t]+CosteFijoProd[i]
[t]*Y[i][t]+CosteCompra[i][t]*XC[i][t]+CosteFijoCompra[i][t]*YC[i][t]);

/*FUNCION OBJETIVO*/
 maximize
      IngresoPedidos+Entregas-AdelantoEntregas-CostesTotales;

 /*RESTRICCIONES*/
 subject to {

//Capacidad de produccion
 forall (t in NbT)
      Produccion: sum (i in NbI) X[i][t]<=CapProd[t];

//Capacidad de almacenamiento
 forall (t in NbT)
      Almacenamiento:sum (i in NbI)S[i][t]<=CapAlm[t];

//Continuidad para t =1
forall (i in NbI){
  if (LeadTimeProd[i]==0 && LeadTimeCompra[i]==0)
   Continuidad_0_1: Sinicial[i]+X[i][1]+XC[i][1]==S[i][1]+sum(p in NbP,k
in 0..T-1)E[p][i][1][1+k]+sum(j in NbI: j!=i) R[i][j]*(X[j][1]+XC[j][1]);
      else {if (LeadTimeProd[i]==0 && LeadTimeCompra[i]!=0)
                  Continudad_0_2: Sinicial[i]+X[i][1]+XCLead[i][1+TLead-
LeadTimeCompra[i]]==S[i][1]+sum(p in NbP,k in 0..T-1)E[p][i][1][1+k]+sum(j
in NbI: j!=i) R[i][j]*(X[j][1]+XC[j][1]);

else{
  if (LeadTimeProd[i]!=0 && LeadTimeCompra[i]!=0)
        Continuidad_0_3: Sinicial[i]+XLead[i][1+TLead-
LeadTimeProd[i]]+XCLead[i][1+TLead-LeadTimeCompra[i]]==
S[i][1]+sum(p in NbP,k in 0..T-1)E[p][i][1][1+k]+sum(j in NbI: j!=i)
R[i][j]*(X[j][1]+XC[j][1]);
      else
      Continuidad_0_4: Sinicial[i]+XLead[i][1+TLead-
LeadTimeProd[i]]+XC[i][1]==S[i][1]+sum(p in NbP,k in 0..T-
1)E[p][i][1][1+k]+sum(j in NbI: j!=i) R[i][j]*(X[j][1]+XC[j][1]);}
}}
```

```
//Continuidad para el resto de t menos t=T
 forall (i in NbI, t in NbT: 1<t<T){
   if (LeadTimeProd[i]<t && LeadTimeCompra[i]<t)
                  Continuidad_t_1: S[i][t-1]+X[i][t-
LeadTimeProd[i]]+XC[i][t-LeadTimeCompra[i]]==S[i][t]+sum(p in NbP,k in
0..T-t)E[p][i][t][t+k]+sum(j in NbI: j!=i) R[i][j]*(X[j][t]+XC[j][t]);
      else {
             if (LeadTimeProd[i]<t && LeadTimeCompra[i]>=t)
             Continuidad_t_2: S[i][t-1]+X[i][t-LeadTimeProd[i]]+
XCLead[i][t+TLead-LeadTimeCompra[i]]==S[i][t]+sum(p in NbP,k in 0..T-
t)E[p][i][t][t+k]+sum(j in NbI: j!=i) R[i][j]*(X[j][t]+XC[j][t]);
              else{
                  if (LeadTimeProd[i]>=t && LeadTimeCompra[i]>=t)
                  Continuidad_t_3: S[i][t-1]+XLead[i][t+TLead-
LeadTimeProd[i]]+XCLead[i][t+TLead-LeadTimeCompra[i]]==S[i][t]+sum(p in
NbP,k in 0..T-t)E[p][i][t][t+k]+sum(j in NbI: j!=i)
R[i][j]*(X[j][t]+XC[j][t]);
                   else
                  Continuidad_t_4: S[i][t-1]+XLead[i][t+TLead-
LeadTimeProd[i]]+XC[i][t-LeadTimeCompra[i]]==S[i][t]+sum(p in NbP,k in
0..T-t)E[p][i][t][t+k]+sum(j in NbI: j!=i) R[i][j]*(X[j][t]+XC[j][t]);}
}}
//Continuidad para t=T
forall (i in NbI){
  if (LeadTimeProd[i]<T && LeadTimeCompra[i]<T)
            Continuidad_T_1_a: S[i][T-1]+X[i][T-LeadTimeProd[i]]+XC[i][T-
LeadTimeCompra[i]]==S[i][T]+sum(p in NbP)E[p][i][T][T]+sum(j in NbI: j!=i)
R[i][j]*(X[j][T]+XC[j][T]);
 else {
      if (LeadTimeProd[i]<T && LeadTimeCompra [i]==T)
      Continuidad_T_2: S[i][T-1]+X[i][T-LeadTimeProd[i]]+
XCLead[i][T+TLead-LeadTimeCompra[i]]==S[i][T]+sum(p in NbP)
E[p][i][T][T]+sum(j in NbI: j!=i) R[i][j]*(X[j][T]+XC[j][T]);
            else{
            if (LeadTimeProd[i]==T && LeadTimeCompra[i]==T)
      Continuidad_T_3: S[i][T-1]+XLead[i][T+TLead-
LeadTimeProd[i]]+XCLead[i][T+TLead-LeadTimeCompra[i]]==S[i][T]+sum(p in
NbP)E[p][i][T][T]+sum(j in NbI: j!=i) R[i][j]*(X[j][T]+XC[j][T]);
else
Continuidad_T_4: S[i][T-1]+XLead[i][T+TLead-LeadTimeProd[i]]+XC[i][T-
LeadTimeCompra[i]]==S[i][T]+sum(p in NbP)E[p][i][T][T]+sum(j in NbI: j!=i)
R[i][j]*(X[j][T]+XC[j][T]);}}
}

//Uso de la binaria W
 forall (p in NbP,i in NbI, t1 in NbT, t2 in NbT: t1<=t2)
   {  epsilon*W[p][i][t1][t2]<=E[p][i][t1][t2];
   E[p][i][t1][t2]<=Demanda[p][i][t2]*W[p][i][t1][t2];}

//Corner cases (limitar las demandas que son 0)
 forall (p in NbP, i in NbI,t1 in NbT, t2 in NbT)
```

```
      Corner cases: sum (t1 in 1..t2 )W[p][i][t1][t2]<=Demanda[p][i][t2];

//Uso binaria E para t2=1
forall (p in NbP, i in NbI)
      Binaria_E1: E[p][i][1][1]==Demanda[p][i][1]*W[p][i][1][1];

//Uso binaria E para el resto de t
 forall (p in NbP, i in NbI,t1 in NbT,t2 in NbT: t2>1 && t1<=t2)
      Binaria_Et: sum(k in 1..t2)E[p][i][k][t2]==Demanda[p][i][t2]*sum (t1
in 1..t2 )W[p][i][t1][t2];

//Para que no se produzca splitting
forall (p in NbP, i in NbI, t2 in NbT)
  { sum (t1 in NbT: t1<=t2) W[p][i][t1][t2]<=1;}

//Uso de la binaria Y
 forall(i in NbI, t in NbT) {
     epsilon*Y[i][t]<=X[i][t];
     X[i][t]<= CapProd[t]*Y[i][t];}

//Uso de la binaria YC
 forall(i in NbI, t in NbT){
     epsilon*YC[i][t]<=XC[i][t];
     XC[i][t]<= alfa*YC[i][t]; }

//Entrega completa de un pedido
 forall (p in NbP)
   Entrega_pedido: DenP[p]*YP[p]==sum(i in NbI,t1 in NbT, t2 in NbT:
t1<=t2)W[p][i][t1][t2];

//Para que no me salgan los warnings
 forall (p in NbP, i in NbI,t1 in NbT, t2 in NbT:t1>t2){
   E[p][i][t1][t2]==0;
   W[p][i][t1][t2]==0;}
 }
```

Listado 19. Código de programación modelo 9.

PRESUPUESTO

1 PRESUPUESTO

En este documento se recoge el presupuesto necesario para el desarrollo de este proyecto. Para la realización del presupuesto total, se realiza un presupuesto parcial donde se tienen en cuenta los costes debidos al uso de recursos materiales y los costes de mano de obra.

2 PRESUPUESTO PARCIAL

2.1 Presupuesto parcial de recursos materiales

Dado que el proyecto presenta un carácter investigativo, los recursos materiales incluidos son únicamente los dispositivos electrónicos y las licencias de softwares empleados en el trabajo.

Se ha considerado un 25% de costes indirectos, según indica "el artículo 9 del Reglamento regulador de la gestión de actividades de investigación, desarrollo, transferencia de tecnología y formación no reglada en la UPV" (Guía Presupuesto UPV, 2018).

Concepto	Unidad	Número	Cantidad (meses)	Precio (€/mes)	Importe (€)
Microsoft 365	Licencia mensual	3	6	5,53	99,54
IBM ILOG CPLEX	Licencia mensual	1	1	151,68	151,68
			Coste Directo		251,22
		25%	Costes Indirectos		62,81
			Coste Total		**314,03**

Tabla 23. Presupuesto parcial recursos materiales (1).

Dado que el ordenador, es un bien material amortizable, se considera que el precio unitario del mismo es 800€ (con el 21% de IVA incluido), lo que supone 632€ como precio unitario (sin IVA). Se considera que un ordenador presenta una vida útil de 6 años. Por lo tanto, se calcula el coste que conllevaría emplearlo en un mes: 8,78 €/mes. Teniendo en cuento que se ha dispuesto del ordenador durante 7 meses, implica que el uso de un ordenador conlleve 61,46€.

$$^{632}/_{6\cdot 12} = 8{,}78€/mes$$

$$8{,}78 \cdot 7 = 61{,}46€$$

Concepto	Unidad	Cantidad (Uds.)	Precio unitario	Importe (€)
Ordenador	ud	3	61,46	184,38
		Coste Directo		184,38
	25%	Costes Indirectos		46,10
		Coste Total		**230,48**

Tabla 24. Presupuesto parcial recursos materiales (2).

2.2 Presupuesto parcial de mano de obra

En cuanto a los costes de mano de obra asociados al proyecto, se tiene en consideración las horas empleadas por la alumna para el desarrollo del TFG y las horas dedicadas a reuniones de tutorización y supervisión del trabajo. Los precios asignados a cada uno se han extraído de la Guía de presupuesto 2018 (Servicio de Gestión de la I+D+i).

Concepto	Unidad	Cantidad (horas)	Precio (€/hora)	Importe (€)
Estudiante Ingeniería Industrial	h	300	15,26	4578,00
Catedrático de Universidad	h	30	51,40	1542,00
Profesor Titular de Universidad	h	2	39,00	78,00
		Coste Directo		6198,00
	25%	Costes Indirectos		1549,50
		Coste Total		**7747,50**

Tabla 25. Presupuesto mano de obra.

3 PRESUPUESTO TOTAL

A partir de los presupuestos parciales de recursos materiales y mano de obra, se calcula el presupuesto ejecución de material. Sobre este se han aplicado un 13% de gastos generales y un 6% asociado a beneficio industrial. Finalmente, se calcula el presupuesto base licitación, con un 21% de IVA.

Concepto	Cantidad	Importe (€)
Presupuesto parcial de recursos materiales		544,50
Presupuesto parcial de mano de obra		7747,50
Presupuesto ejecución de material		8292,00
Gastos generales	13%	1077,96
Beneficio industrial	6%	497,52
Presupuesto ejecución de contrata		9867,48
IVA	21%	2072,17
Presupuesto base licitación		11939,65

Tabla 26. Presupuesto final.

Finalmente, el presupuesto total asciende a ONCE MIL NOVECIENTOS TREINTA Y NUEVE euros con SESENTA Y CINCO céntimos. En este presupuesto se tiene en consideración únicamente el desarrollo de este TFG.

Para poder implementar este proyecto en una compañía, se debería realizar un estudio específico de la empresa en la que se quisiese implementar, ya los precios y cantidades de los conceptos variarían según el tamaño de la misma.