



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

Sistema para la Reprogramación Dinámica de Producción
bajo eventos inesperados.

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Organización y Logística

AUTOR/A: Figueroa Avila, Andy Jose

Tutor/a: Poler Escoto, Raúl

Cotutor/a: Andrés Navarro, Beatriz

CURSO ACADÉMICO: 2023/2024

Resumen

Este Trabajo Fin de Máster propone un enfoque basado en un Modelo de Programación Lineal Mixta (MILP) para abordar el problema de programación de producción en talleres de trabajo sujetos a la aparición de eventos inesperados, con el objetivo de minimizar los costos totales asociados a la producción y a los cambios de programación. En una primera etapa se calcula un plan de producción inicial con la asignación óptima de lotes de producción a máquinas, considerando las operaciones requeridas. En sucesivas etapas se ejecuta la reprogramación dinámica del plan de producción inicial en respuesta a eventos inesperados, como averías de maquinaria o escasez de materias primas. La reprogramación se adapta al estado actual del taller de producción y se ejecuta con el objetivo de encontrar una nueva solución óptima o casi óptima que pueda adaptarse rápidamente a cambios en las condiciones del entorno de producción, con el fin de minimizar cambios significativos respecto al plan de producción inicial.

Palabras clave:

Programación de Producción; MILP; optimización, eventos inesperados; reprogramación

Abstract

This Master Thesis proposes an approach based on a Mixed Linear Programming Model (MILP) to address the production scheduling problem in job shops subject to the occurrence of unexpected events, with the objective of minimizing the total costs associated with production and scheduling changes. In a first stage, an initial production plan is calculated with the optimal allocation of production batches to machines, considering the required operations. In successive stages, dynamic rescheduling of the initial production plan is executed in response to unexpected events, such as machine breakdowns or raw material shortages. The rescheduling is adapted to the current state of the production shop floor and is executed with the objective of finding a new optimal or near-optimal solution that can quickly adapt to changes in the conditions of the production environment, in order to minimize significant changes with respect to the initial production plan.

Keywords:

Production Scheduling; MILP; optimization; unexpected events; rescheduling.

Resum

Aquest Treball Fi de Màster proposa un enfocament basat en un Model de Programació Lineal Mixta (MILP) per a abordar el problema de programació de producció en tallers de treball subjectes a l'aparició d'esdeveniments inesperats, amb l'objectiu de minimitzar els costos totals associats a la producció i als canvis de programació. En una primera etapa es calcula un pla de producció inicial amb l'assignació òptima de lots de producció a màquines, considerant les operacions requerides. En successives etapes s'executa la reprogramació dinàmica del pla de producció inicial en resposta a esdeveniments inesperats, com a avaries de maquinària o escassetat de matèries primeres. La reprogramació s'adapta a l'estat actual del taller de producció i s'executa amb l'objectiu de trobar una nova solució òptima o quasi òptima que puga adaptar-se ràpidament a canvis en les condicions de l'entorn de producció, amb la finalitat de minimitzar canvis significatius respecte al pla de producció inicial.

Paraules clau:

Programació de Producció; MILP; optimització, esdeveniments inesperats; reprogramació

Agradecimientos

Antes de comenzar con este trabajo, quisiera agradecer a Dios por darme la oportunidad de cursar mis estudios de máster con éxito. También deseo expresar mi gratitud a mi familia, en especial a mi madre, que ha sido mi mayor apoyo en todo momento, alentándome a seguir adelante y cumplir con mis objetivos.

Quiero agradecer especialmente a mis tutores, Raúl Poler y Beatriz Navarro, por su constante apoyo y por tomarse el tiempo para ayudarme a comprender aspectos y conceptos necesarios para desarrollar este trabajo.

Finalmente, agradezco a mi gran amigo Emanuel, quien en muchas ocasiones me ayudo a superar momentos de estrés y estancamiento.

Índice General

Capítulo 1. Introducción	1
1.1 Introducción.....	1
1.2 Objetivo General.....	2
1.3 Objetivos Específicos	2
1.4 Estructura del Trabajo.....	3
Capítulo 2. Descripción del Problema.....	5
2.1 Descripción del problema	5
Capítulo 3. Estado del Arte	10
3.1 Gestión de Planificación y Programación de la Producción	10
3.2 Eventos Inesperados en la Programación de la Producción	14
3.3 Reprogramación de la producción.....	16
3.4 Soluciones Propuestas en la Literatura.....	20
Capítulo 4. Descripción de la Solución	25
4.1 Modelo de Programación Lineal Entera Mixta (MILP).....	25
4.2 Estrategia de Reprogramación.....	32
Capítulo 5. Experimentos.....	37
5.1 Generación de Datos	38
5.2 Experimentos De Programación Inicial.....	40
5.3 Experimentos de Reprogramación Bajo Eventos Inesperados	47
5.3.1 Reprogramación en Respuesta a Averías de Maquinaria	47
5.3.2 Reprogramación en Respuesta a Escasez de Materias Primas	52

Capítulo 6. Conclusiones.....	57
Bibliografía.....	59
Anexos	64
Anexo 1. Código de iterativo de reprogramación.....	64
Anexo 2. Código de cálculo de escasez de materia prima.....	71

Índice de Figuras

Figura 1. Representación de Asignaciones de Producción a Máquinas	6
Figura 2. Costos de Reprogramación	19
Figura 3. Estrategia iterativa-reactiva de Reprogramación.....	34
Figura 4. Generación de Datos	39

Índice de Tablas

Tabla 1. Revisión Bibliográfica de Métodos de Solución	23
Tabla 2. Nomenclatura de MILP.....	26
Tabla 3. Experimentos De Programación Inicial: Tamaño de instancia	40
Tabla 4. Experimentos De Programación Inicial: Inventario inicial, coste de mantenimiento, coste de diferir	40
Tabla 5. Experimentos De Programación Inicial: Demanda.....	41
Tabla 6. Experimentos De Programación Inicial: Datos de máquinas	41
Tabla 7. Experimentos De Programación Inicial: Datos de producción	41
Tabla 8. Experimentos De Programación Inicial: Inventario inicial de materia prima....	42
Tabla 9. Experimentos De Programación Inicial: Lista de materiales y tasa de consumo	42
Tabla 10. Resultados De Programación Inicial: Plan inicial de producción.....	43
Tabla 11. Resultados De Programación Inicial: Configuración de máquinas.....	43
Tabla 12. Resultados De Programación Inicial: Inventarios de producto terminado, pedidos pendientes	44
Tabla 13. Resultados De Programación Inicial: Inventarios de materia prima	44
Tabla 14. Resultados experimentales de programación inicial	45
Tabla 15. Experimento de reprogramación (Avería de máquina): Plan de Producción .	48
Tabla 16. Experimento de reprogramación (Avería de máquina): Estado de las máquinas	48
Tabla 17. Experimento de reprogramación (Avería de máquina): Productos Inamovibles	49
Tabla 18. Experimento de reprogramación (Avería de máquina): Cantidad de Producto Producida en Máquina Averíada	49
Tabla 19. Experimento de reprogramación (Avería de máquina): Resultado comparativo de plan de producción	50
Tabla 20. Experimento de reprogramación (Avería de máquina): Resultados comparativos de pedidos pendientes por entregar	51

Tabla 21. Experimento de reprogramación (Escasez de materia prima): Plan de producción inicial.....	52
Tabla 22. Experimento de reprogramación (Escasez de materia prima): Lista de materiales	53
Tabla 23. Experimento de reprogramación (Escasez de materia prima): Inventario inicial de materia prima	53
Tabla 24. Experimento de reprogramación (Escasez de materia prima): Materia prima faltante.....	54
Tabla 25. Experimento de reprogramación (Escasez de materia prima): Plan de producción.....	54
Tabla 26. Experimento de reprogramación (Escasez de materia prima): Pedidos pendientes a entregar	55

Capítulo 1. Introducción

1.1 Introducción

En el entorno competitivo y dinámico de la manufactura moderna, la eficiencia en la programación de producción es crucial para mantener la competitividad y minimizar los costos operativos. Los talleres de trabajo, que suelen estar equipados con múltiples máquinas y equipos, enfrentan constantemente desafíos asociados con eventos inesperados que pueden perturbar el flujo de producción, como averías de maquinaria y escasez de materias primas. Estos eventos inesperados pueden causar interrupciones significativas y aumentar los costos asociados a la producción y a los ajustes en los planes de producción.

Este Trabajo Fin de Máster aborda el Problema de Programación de Producción en Talleres Sujetos a Eventos Inesperados mediante la aplicación de un Modelo de Programación Lineal Mixta (MILP). El MILP es una herramienta valiosa que permite la formulación y resolución de problemas de optimización que involucran variables continuas y discretas, lo que lo convierte en un enfoque adecuado para el complejo entorno de producción descrito.

La investigación se centra en desarrollar un enfoque práctico para la programación de producción que no solo optimiza el plan inicial, sino que también incluye un mecanismo de reprogramación dinámica. El objetivo es minimizar los costos totales asociados a la producción y cambios en el plan de producción inicial, garantizando al mismo tiempo que el sistema sea capaz de adaptarse rápidamente a los cambios en las condiciones del entorno de producción.

Para la implementación del modelo y la realización de experimentos, se utiliza el lenguaje de programación Python. Para validar el modelo, se realizan experimentos diseñados para demostrar la efectividad del enfoque propuesto y su capacidad para manejar diversas situaciones inesperadas en el taller.

1.2 Objetivo General

El objetivo general de este TFM es desarrollar un MILP para la resolución del problema de El Problema de Programación de Producción sujeto a Eventos Inesperados sujeto a eventos inesperados, con el objetivo de minimizar los costos totales asociados a la producción y a los cambios en el plan de producción. El trabajo propuesto se centra en la formulación de un modelo inicial para la asignación óptima de lotes de producción a máquinas y una estrategia de reprogramación dinámica que permita ajustar el plan ante eventos inesperados, como averías de maquinaria y escasez de materias primas

1.3 Objetivos Específicos

I. Estado del Arte:

- Analizar las soluciones existentes y los enfoques actuales en la literatura sobre la programación de producción y la reprogramación dinámica en presencia de eventos inesperados.
- Identificar las limitaciones y brechas en la investigación actual para adoptar características y aspectos fundamentales que puedan aportar para el desarrollo del modelo propuesto.

II. Diseño del Sistema de Reprogramación:

- Desarrollar un sistema de reprogramación que incluya la formulación de un MILP para la asignación óptima de lotes de producción a máquinas en un taller.
- Diseñar un mecanismo de reprogramación dinámica que permita ajustar el plan de producción inicial en respuesta a eventos inesperados.

III. Validación del Sistema:

- Implementar el MILP y la estrategia de reprogramación en un entorno de resolución utilizando el lenguaje de programación Python.
- Validar el sistema mediante la realización de experimentos que demuestren la efectividad del modelo y la estrategia de reprogramación en diferentes escenarios.

1.4 Estructura del Trabajo

El contenido de este trabajo está estructurado de la siguiente manera:

Capítulo 2. Descripción del Problema:

Este capítulo proporciona una visión general del problema de programación de producción en talleres de trabajo sujeto a eventos inesperados, así como también los supuestos considerados en el problema.

Capítulo 3. Estado del Arte:

Se realiza una revisión de la literatura sobre planificación y programación de la producción, con un enfoque en el impacto de los eventos inesperados en estos procesos. En primer lugar, se contextualiza la planificación y programación de la producción en un entorno general. Posteriormente, se analiza como las averías de maquinaria y escasez de materias primas afectan la planificación de producción. Finalmente, se revisan las soluciones propuestas en la literatura para la reprogramación frente a estos eventos inesperados, y se resumen las principales estrategias y enfoques encontrados.

Capítulo 4. Descripción de la Solución Propuesta:

En este capítulo se detalla el MILP desarrollado para abordar el problema de programación de producción. Se describe la formulación matemática del modelo encargado de optimizar la asignación de recursos y gestionar eventos inesperados. Además, se presenta la estrategia de reprogramación dinámica implementada para ajustar el plan de producción en función de estos eventos.

Capítulo 5. Experimentos:

La sección se centra en la aplicación práctica del modelo propuesto a través de la realización de experimentos. En primer lugar, se describe el proceso de generación de datos necesarios para evaluar el modelo, detallando la metodología empleada para crear instancias representativas de diferentes tamaños.

Posteriormente, se presentan y analizan los resultados de los experimentos realizados con instancias pequeñas y medianas, enfocadas al cálculo de la programación inicial.

Finalmente, se llevan a cabo experimentos adicionales para evaluar la capacidad del sistema de reprogramación para adaptarse a eventos inesperados.

Capítulo 6. Conclusiones

Resume los resultados obtenidos, evalúa la efectividad del modelo y la estrategia de reprogramación, y propone recomendaciones para futuras líneas de investigación y aplicaciones.

Capítulo 2. Descripción del Problema

2.1 Descripción del problema

El Problema de Programación de Producción sujeto a Eventos Inesperados consiste en determinar el plan o programa de producción inicial con la asignación óptima de lotes de producción de N productos a M máquinas paralelas no idénticas en un taller de trabajo (ver Figura 1, pág. 6).

Cada producto i ($i = 1, 2, \dots, N$) consume una cantidad determinada de materia prima r ($r = 1, 2, \dots, R$) durante su proceso de producción, y existe un coste asociado a su producción y configuración, dependiente de la máquina j ($j = 1, 2, \dots, N$) a la que se le asigne. Asimismo, cada máquina está sujeta a posibles averías que pueden interrumpir el proceso de producción, y existe la posibilidad de escasez de materia prima por retraso en la cadena de suministro.

Los productos deben cumplir con su fecha de entrega dentro de un horizonte temporal discretizado en períodos de tiempo t ($t = 1, 2, \dots, T$). La finalización anticipada de la producción de un producto antes de su fecha de vencimiento incurre en costes de inventario. Las máquinas pueden procesar como máximo un producto a la vez, aunque pueden tener asignado más de un producto i esperando a ser procesado. Cada máquina j está limitada por su capacidad de producción(tiempo), la cual varía en función del producto asignado.

El objetivo es encontrar una asignación inicial óptima o casi óptima de lotes de producción que minimice los costos totales de producción. Esto incluye adaptar rápidamente el plan de producción al estado actual del taller en caso de alteración de las condiciones iniciales por la aparición de eventos inesperados, como averías de maquinaria o escasez de materias primas, tratando minimizar cambios significativos respecto al plan de producción inicial durante la reprogramación.

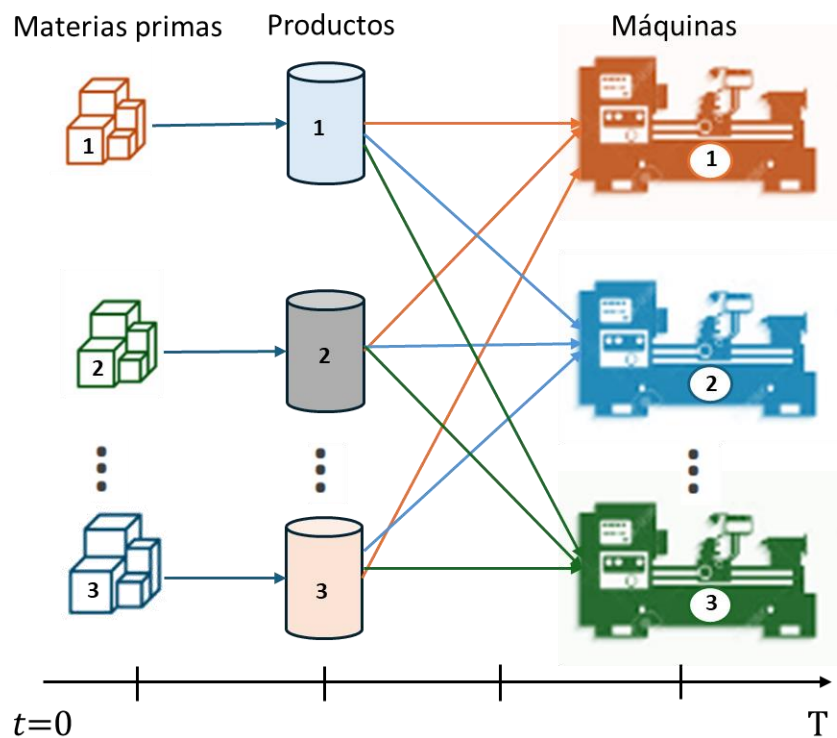


Figura 1. Representación de Asignaciones de Producción a Máquinas

La figura muestra la asignación de productos a máquinas en el taller de trabajo, presentando las opciones disponibles para asignarlos a diferentes máquinas en un horizonte temporal T , discretizado en períodos de tiempo t . Esta asignación inicial es crucial para determinar el plan de producción óptimo, que busca minimizar los costos totales y cumplir con los plazos de entrega.

2.1.1 Suposiciones complementarias

Se considera los siguientes supuestos complementarios en el problema:

I. Configuración:

- Costos de configuración dependientes de la máquina.
- Cada producto requiere una configuración específica en la máquina asignada, lo que conlleva un tiempo asociado.
- El tiempo de configuración se suma al tiempo de procesamiento y se considera como parte del tiempo total de producción en cada período de tiempo t .

II. **Materia prima:**

- El procesamiento de productos conlleva el consumo de materia prima, establecida según la lista de materiales (BOM). La lista de materiales detalla las cantidades requeridas de materia prima r para la producción de cada producto i .
- Se supone que el inventario inicial de materia prima utilizado en la planificación de la producción refleja la cantidad total esperada, incluyendo las unidades físicamente presentes y aquellas que se esperan recibir del proveedor dentro del plazo planificado.
- No se considera el costo de almacenamiento de materia prima en el problema. Se enfoca únicamente en las limitaciones relacionadas con la disponibilidad de materia prima en inventario para su consumo.

III. **Eventos Inesperados:** Se consideran dos tipos principales de eventos capaces de causar interrupciones en el taller de trabajo: la escasez de materia prima y las averías de máquinas. Lo supuestos vinculados a dichos eventos son:

- Las averías de maquinaria dejan a una máquina fuera de servicio durante uno o más períodos.
- En caso de interrupción en el sistema de producción, los productos no afectados continuarán con su proceso normal.
- Todas las máquinas se consideran continuamente disponibles hasta que se presente una avería.
- La escasez de materia prima resulta de retrasos en los períodos de entrega preestablecidos por el proveedor.
- Los retrasos en la entrega de materia prima están limitados a un solo período.
- Para abordar el evento de escasez de materia prima, se supone que cada tipo de materia prima está asignado exclusivamente a un solo producto, de modo que ningún tipo de materia prima se comparte entre diferentes productos. En caso de escasez, esta afecta únicamente al

producto específico asociado con la materia prima en cuestión, y no a otros productos.

IV. **Reprogramación:** Se consideran los siguientes aspectos relacionados con la reprogramación:

- La reprogramación se lleva a cabo cada vez que el sistema de producción se interrumpe debido a eventos inesperados.
- La reprogramación por avería de máquina únicamente se considera cuando una máquina estará fuera de servicio durante un período completo debido a una avería prolongada.
- Los costes asociados con la reprogramación, como los de instalación, transporte, etc. (ver Figura 5, pág.19), no se incluyen explícitamente en el problema. Sin embargo, el impacto de una reprogramación se refleja en los costos operativos totales, dependiendo de cómo se actualice el plan de producción con respecto al inicial.

Capítulo 3. Estado del Arte

3.1 Gestión de Planificación y Programación de la Producción

Planificación de la Producción

En la actualidad, la gestión eficiente de la producción en entornos industriales constituye un elemento crucial para el éxito y la competitividad de las empresas. Este éxito depende en gran medida de dos fases fundamentales: una sólida planificación y una eficaz programación. Estas etapas no solo establecen las bases estratégicas, sino que también optimizan la ejecución de los procesos productivos, permitiendo a las empresas alcanzar sus objetivos de manera eficiente y mantenerse competitivas en el mercado.

Según (García, 2007) la planificación es un proceso por el cual cada departamento de una empresa organiza sus recursos en el tiempo con el objetivo de optimizar su uso y obtener el mayor beneficio posible.

Por lo tanto, el objetivo principal del proceso de planificación de la producción es satisfacer las demandas del cliente en cuanto a cantidad, coste y servicio, minimizando los costes para la empresa y optimizando la utilización de los recursos disponibles.

La planificación de la producción implica la evaluación de la demanda esperada, la determinación de las capacidades de producción, la programación de la utilización de recursos y la coordinación de actividades para lograr los resultados deseados. En esencia, la planificación de la producción sirve como el marco estratégico que guía todas las acciones relacionadas con la fabricación de productos o servicios.

En la gestión de la producción, es esencial comprender las fases implicadas en la planificación de la producción, fases que de acuerdo con (Domínguez et al., 1995) se deberán ejecutar en cualquier industria.

La literatura describe una jerarquía de las fases a seguir en la planificación, organizadas según su horizonte temporal en un orden descendente que refleja el impacto de las decisiones, logrando una integración vertical entre los objetivos estratégicos, tácticos y operativos. Diversos autores establecen en términos generales que el modelo jerárquico comúnmente se divide en tres horizontes temporales, aunque el tiempo asignado a cada uno puede variar según el sector o el autor que lo describa (Companys et al., 1993; Chase et al., 1994; Everett et al., 1991). Según el enfoque de horizontes temporales propuesto por Chase et al. (1994), se describe de la siguiente manera:

I. **Planificación Estratégica (Largo Plazo).**

Nivel: Estratégico

Aborda un horizonte temporal de entre tres y cinco años y se enfoca en establecer los objetivos a largo plazo de la organización, así como en diseñar las estrategias para alcanzarlos.

II. **Planificación Agregada y Planificación Maestra de Producción (Mediano Plazo)**

Nivel: Táctico

Cubre un periodo de entre 6 y 18 meses y se centra en la planificación táctica de la producción. Incluye la Planificación agregada, que se enfoca en la determinación de los niveles de producción a nivel global y en la asignación de recursos, y la Planificación maestra de producción (MPS), que se encarga de generar cantidades y fechas para la fabricación de productos específicos.

III. **Planificación de Necesidades y Planificación Detallada (Corto Plazo).**

Nivel: Operativo

Se extiende desde un día hasta seis meses y se concentra en la ejecución de las operaciones diarias de producción. La Planificación de necesidades (MPR) determina los requisitos de materiales y recursos para cumplir con el plan maestro de producción, mientras que la Planificación detallada se encarga de programar las actividades de producción a corto plazo, como la asignación de tareas a los trabajadores y la programación de la maquinaria.

Programación de la Producción

La programación de la producción se define como el proceso de asignar recursos generalmente limitados y programar las actividades necesarias para cumplir con los objetivos a corto plazo de producción de manera eficiente. Esto implica el determinar cuándo procesar un trabajo, con qué máquina hacerlo y qué prioridad asignar a cada tarea (Valledor et al., 2018). Las decisiones tomadas en este contexto son cruciales para maximizar la eficiencia operativa y minimizar los tiempos de inactividad y los costos de producción. Esta etapa de planificación se enfoca en la ejecución de las operaciones diarias de producción, asegurando que los recursos estén disponibles y que las tareas se realicen en el momento adecuado. A diferencia de la planificación estratégica y táctica, que abordan horizontes temporales más largos y decisiones de alto nivel, la programación de la producción se centra en el día a día en los talleres de producción.

Al ser un proceso que puede estar expuesto a un entorno dinámico, la programación de producción debe adaptarse rápidamente a los cambios en la demanda, la disponibilidad de recursos y las condiciones del mercado. Por lo tanto, es fundamental que este proceso esté en constante interacción con las demás etapas que componen la planificación de producción. Esto asegura que la producción se alinee con las metas estratégicas y tácticas de la organización, manteniendo la flexibilidad necesaria para responder a las variaciones del entorno productivo.

Entre los aspectos fundamentales para lograr una programación de la producción eficaz se encuentran los siguientes.

- **Demanda del mercado:** Retroalimentación sobre las fluctuaciones en la demanda diaria o semanal, lo cual ayuda a ajustar rápidamente los planes en respuesta a cambios en la demanda.
- **Gestión de Inventarios:** El control de stock ayuda a mantener un seguimiento preciso de los niveles de inventario para asegurar que los materiales necesarios estén disponibles cuando se necesiten, evitando sobre stock o desabastecimiento.

- **Gestión de Proveedores:** La evaluación de Proveedores y monitorización del rendimiento de los proveedores es esencial para asegurar la calidad, puntualidad de las entregas y mejorar la sincronización de la cadena de suministro.
- **Estado del Taller de Trabajo:** Mantener información actualizada sobre el estado de las máquinas y los trabajos en curso es esencial para tomar decisiones informadas. Esto incluye el seguimiento de la disponibilidad de máquinas, el progreso de las órdenes de trabajo y cualquier problema que pueda surgir en el taller.

Según (Pinedo, 2012) los objetivos de la programación de producción también pueden adoptar otras formas diferentes que puede comprenderse desde la minimización del tiempo de finalización de la última tarea, hasta la minimización del número de tareas completadas.

Las empresas deben adherirse a los plazos de producción y las fechas de entrega acordadas con los clientes, ya que el no cumplir puede derivarse en pérdidas económicas considerables y en la disminución de la confianza del cliente (Fanjul, 2010). Un retraso en la entrega conlleva una penalización que puede resultar en una pérdida de la reputación comercial. La gravedad de esta penalización varía según la importancia del pedido o del cliente y la duración del retraso, por lo que programar la producción de manera efectiva ayuda a reducir estas consecuencias.

Actualmente, con la introducción de la Industria 4.0 en el sector productivo, numerosas instalaciones industriales están adoptando la tecnología digital para mejorar, automatizar y modernizar todas las operaciones (Mordor Intelligence, 2024). Esto está alterando la forma en que las empresas manejan los sistemas, siendo cada vez más sofisticados debido al crecimiento tecnológico, estos sistemas facilitan en gran medida la interacción de información con las funciones de programación de la producción.

3.2 Eventos Inesperados en la Programación de la Producción

La programación de la producción se enfrenta a desafíos significativos debido a la compleja coordinación de múltiples operaciones y recursos en entornos industriales. Estos desafíos son aún más pronunciados en los talleres de trabajo, donde la eficiencia de la producción depende de la sincronización precisa de maquinaria, mano de obra y materiales. En este contexto, uno de los factores críticos que merece atención especial son los eventos inesperados, los cuales pueden tener un impacto significativo en la ejecución y planificación de la producción.

En este sentido, el resto de la sección se centra en explorar cómo los eventos inesperados afectan a los talleres de trabajo, la importancia de la reprogramación y las soluciones descritas en la literatura para gestionar estos desafíos de manera efectiva.

Impactos en la Producción por Eventos Inesperados

La mayoría de los sistemas de fabricación se enfrentan a entornos dinámicos en los que los eventos inesperados pueden provocar un cambio en los planes programados (Takeda-Berger et al., 2022). Durante la ejecución de los trabajos, pueden aparecer muchos eventos inesperados que alteran el plan de producción ya establecido. (Dutta, 1990; Dhingra et al., 1992) Denominan a estos eventos inesperados como factores de reprogramación, debido a que la presencia de uno de ellos desencadenará una necesidad por ajustar el programa de producción establecido inicialmente. Entre ellos los más comunes son:

- Avería de maquinaria
- Retraso en la llegada o escasez de materiales
- Cambio de fecha de vencimiento del pedido (retraso o adelanto)
- Cambio de prioridad laboral

La mayoría de los sistemas de fabricación se enfrentan a entornos dinámicos en los que los eventos inesperados pueden provocar un cambio en los planes programados (Takeda-Berger et al.,2022).

El impacto que puede generar la avería de maquinaria y la escasez de materia prima puede ser significativo en la programación de la producción. Según publicaciones de portales como (TheCircularCampus, 2023; Deskera Academy, 2023), estos eventos pueden influir en gran medida de la siguiente manera:

- **Interrupciones en la Producción:** La maquinaria averiada puede detener la producción de un producto por completo hasta que se realicen las reparaciones necesarias, provocando tiempos muertos y en ciertas circunstancias puede paralizar la línea de producción entera. De igual manera, la falta de materiales esenciales impide la continuidad del proceso de fabricación, deteniendo las operaciones hasta que se reabastezca el inventario.
- **Retrasos en la Entrega:** Tanto las averías de máquina como la escasez de materiales esenciales obligan a reprogramar las tareas y ajustar los tiempos de producción, provocando en muchos casos el diferimiento de pedidos a producir y consecuentemente retrasando la entrega a los clientes, lo cual afecta la capacidad de la empresa para cumplir con los plazos acordados y puede producir insatisfacción en el cliente.
- **Aumento de Costos Operativos:** Las reparaciones de emergencia y el reemplazo de componentes rotos suelen ser costosos. Además, costos asociados a reconfiguraciones y el tiempo de inactividad de las máquinas pueden incurrir en gastos adicionales por la necesidad de acelerar la producción posteriormente mediante horas extras. También, la urgencia por obtener materiales puede incrementar los costos debido a los precios más altos y gastos de envío, obligando a la empresa a adquirir insumos a un precio elevado.

- **Reducción en la Productividad:** Los tiempos de inactividad y la necesidad de reprogramar tareas disminuyen la eficiencia del taller cuando no se cuenta con capacidad disponible suficiente, reduciendo la cantidad de productos fabricados en un período determinado.
- **Pérdida de Ingresos:** Las interrupciones prolongadas reducen la capacidad de producción, resultando en menos productos disponibles para la venta y, por tanto, en una disminución de los ingresos. Además, el tiempo perdido durante las reparaciones no se puede recuperar fácilmente. La falta de materiales también puede resultar en una menor producción y menos productos disponibles para la venta, disminuyendo los ingresos y en muchos casos provocando la cancelación de contratos y pérdida de ventas.

Los fabricantes necesitan nuevas soluciones para actualizar su plan de producción cuando se produce un cambio en el sistema de producción o en su entorno (Tliba et al., 2023). Por lo que la gestión efectiva requiere la capacidad de reaccionar rápidamente ante ellos, y mitigar sus efectos en la medida de lo posible. Una estrategia ampliamente discutida y considerada como una de las mejores opciones para abordar estos desafíos es la reprogramación.

3.3 Reprogramación de la producción

La reprogramación se define como el proceso de actualización de la programación de la producción existente en respuesta a las interrupciones que puedan ocurrir (Jiménez et al., 2018). Su objetivo principal es proporcionar una respuesta inmediata a eventos inesperados, con la finalidad de mantener la eficiencia operativa, logrando reducir tanto los tiempos de inactividad como los costes operativos que conlleva sus consecuencias.

(Janak et al., 2006; Rahman et al., 2018; Vieira et al., 2003; Shafaei & Brunn, 1999) Detallan que una reprogramación adecuada y dinámica en respuesta a disrupciones inesperadas en el taller de producción puede minimizar los impactos en el plan de producción, concluyendo que, a medida que aumenta la incertidumbre en los

talleres, la reprogramación frecuente se convierte en una estrategia más eficaz para lograr mayor solidez en el plan de producción.

Para una reprogramación efectiva, (Vieira et al., 2003) identifica los aspectos esenciales a considerar. A continuación, se presentan los aspectos clave según el estudio:

- **Entorno de Reprogramación:** Define el contexto y las condiciones específicas en las que se lleva a cabo la reprogramación. Esto incluye identificar el conjunto de trabajos y tareas que deben ser considerados para la programación, así como las restricciones operativas y la disponibilidad de recursos.
- **Estrategia de Reprogramación:** Especifica la metodología para actualizar el programa de producción existente, especificando que eventos inesperados o cambios en el entorno de producción son los que desencadenan la reprogramación. Además, determina si es necesario generar un nuevo programa de producción completo o solo realizar ajustes al plan de producción actual durante el proceso de reprogramación.
- **Método de reprogramación:** Método de Reprogramación: Establece el tipo de método que se encargará de modificar el plan de producción inicialmente establecido. Este método puede incluir la generación de horarios, que se dividen en horarios nominales, basados en condiciones ideales sin interrupciones, y horarios robustos, diseñados para resistir variaciones y interrupciones. También puede incluir la reparación de horarios, que abarca la reprogramación parcial, ajustando solo una parte del plan; la regeneración completa, creando un nuevo plan desde cero; y la reprogramación por desplazamiento a la derecha, que ajusta las tareas hacia adelante en el tiempo sin modificar su orden.
- **Política de Reprogramación:** Determina cuándo y cómo se realiza la reprogramación, asimismo estableciendo los objetivos como la estabilidad de la programación y/o la minimización de costes.

Costos de reprogramación

Las industrias buscan minimizar múltiples objetivos mediante la reprogramación de la producción. Entre estos objetivos se incluyen la minimización del coste total, la reducción del inventario en proceso (WIP) y los costes asociados con las fechas de entrega incumplidas. En la actualidad, estos objetivos son de mayor prioridad para los encargados de la planificación de la producción, ya que mejoran el desempeño económico frente a interrupciones, en comparación con objetivos basados únicamente en medidas de desempeño temporal para alcanzar eficiencia en el cronograma. (Olumolade, 1996; Gopalakrishnan et al., 1997) indican que los costos de reprogramación se dividen en las siguientes categorías:

- **Costos Computacionales:** Incluyen la carga necesaria para ejecutar sistemas de programación, junto con costos no recurrentes de inversiones en sistemas de información (sensores, redes, hardware y software) y costos recurrentes de administración y mantenimiento. Estos costos pueden ser significativos.
- **Costos de Instalación:** Ocurren al asignar herramientas y accesorios según el programa de producción, generando costos adicionales al modificar el plan inicial. Este proceso puede implicar la reconfiguración de equipos, ajustes en el flujo del proceso, lo que puede añadir complejidad a la reprogramación.
- **Costos de Transporte y Manejo de Materiales:** Relacionados con la entrega anticipada de materiales y la manipulación adicional de trabajos dentro del taller, afectando la eficiencia y generando costos adicionales. Estos costos también incluyen el tiempo y esfuerzo necesarios para mover materiales y productos a través del taller, lo que puede crear cuellos de botella y reducir la productividad.

Estos costes pueden variar dependiendo de la magnitud de los cambios y la frecuencia de reprogramaciones que se realizaran a lo largo del horizonte de tiempo.

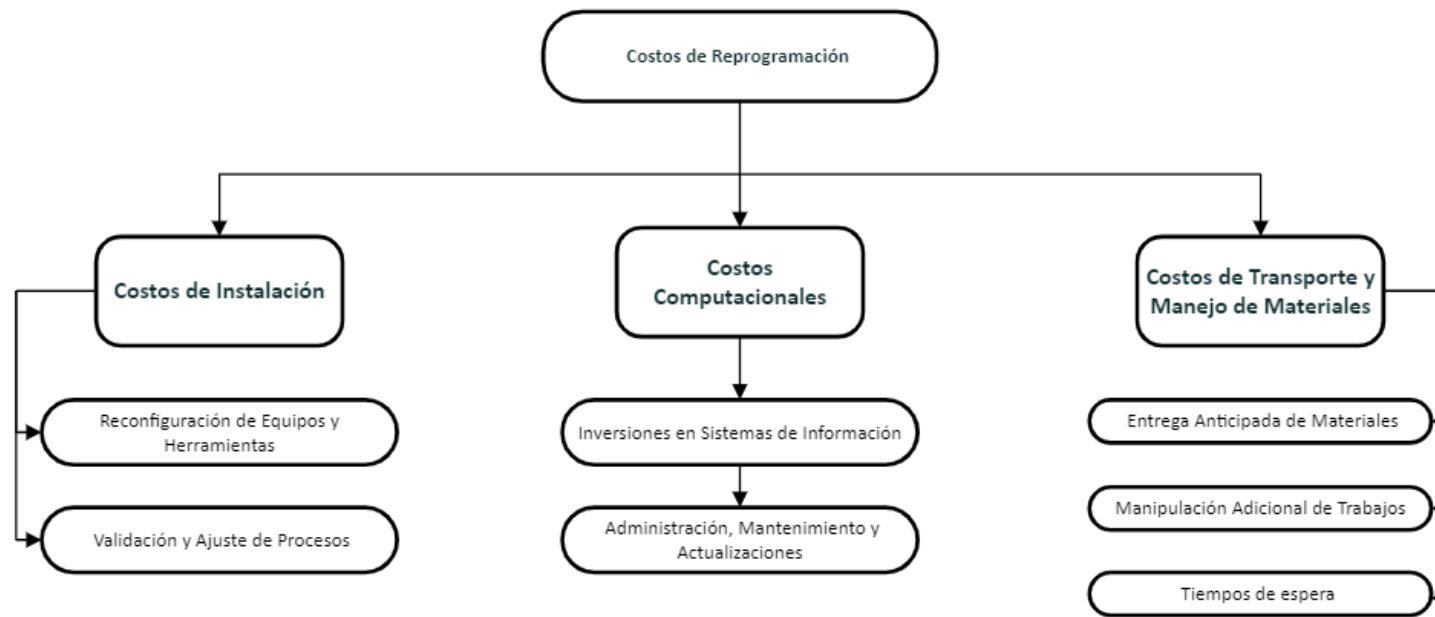


Figura 2. Costos de Reprogramación

Los costos computacionales incluyen tanto costos recurrentes como no recurrentes. Los costos no recurrentes hacen referencia a las inversiones iniciales en hardware y software necesarios para ejecutar sistemas de programación, como sensores, redes de comunicación y software especializado. Por otro lado, los costos recurrentes comprenden los gastos continuos de administración, mantenimiento y actualizaciones de estos sistemas, asegurando su funcionamiento óptimo y adaptabilidad a lo largo del tiempo.

3.4 Soluciones Propuestas en la Literatura

A continuación, se presenta una breve descripción de diversas soluciones propuestas en la literatura (ver Tabla 1, pág. 23) para abordar los problemas asociados con la aparición de eventos inesperados que requieren una reprogramación dinámica en entornos de producción. Cada estudio examina diferentes aspectos de estos problemas, utilizando enfoques y modelos variados para mejorar la eficiencia y la estabilidad de los programas de producción obtenidos. Se detallan las soluciones encontradas, destacando los objetivos y métodos de resolución utilizados.

(Tighazoui & Sauvey, 2021) proponen un modelo Programación Lineal Entera Mixta (MILP) para la reprogramación de una sola máquina en presencia de trabajos que llegan como eventos inesperados. Implementan una estrategia predictiva-reactiva iterativa para abordar la reprogramación en tiempo real. Sus objetivos incluyen minimizar el total de tiempos de espera ponderados (TWWT) y un nuevo criterio de estabilidad, medido por la desviación total ponderada del tiempo de finalización (TWCTD).

(Liu et al., 2018) abordan el problema de programación de taller de flujo de permutación (PFSP) con la llegada de nuevos pedidos, utilizando un algoritmo metaheurístico llamado codicioso iterado. Este método mejorado de destrucción y construcción, junto con un nuevo método de reparación (elG_Rep), permite una reprogramación dinámica eficiente para enfrentar los cambios generados en las condiciones iniciales.

(Janak et al., 2006) presentan un modelo MILP para la programación reactiva, con el objetivo de minimizar la variación del tiempo de finalización total. Consideran cierres de unidades de procesamiento y modificaciones de pedidos como eventos inesperados, proponiendo una estrategia de programación reactiva con fijación de variables no afectadas por estos eventos, logrando cronogramas más estables en las reprogramaciones posteriores a la presencia de interrupciones causadas por dichos eventos.

(Fang & Xi, 1997) estudian el problema de programación de taller de trabajo (JSP) en un entorno dinámico, considerando la llegada de nuevos trabajos, averías de máquinas y fechas de vencimiento. Proponen una estrategia de programación de horizonte móvil para minimizar la duración total del tiempo de finalización y penalizar los trabajos tardíos, considerando también tiempos de configuración dependientes de la secuencia.

(Valledor et al., 2020) desarrollan una formulación MILP para la reprogramación en entornos de taller de flujo de permutación dinámico (PFSP) con un enfoque multiobjetivo: minimizar la duración total del tiempo de finalización (Makespan), tardanza ponderada total (TWT), y estabilidad del cronograma. Abordan averías de máquinas, cambios en tiempos operativos y la aparición de nuevos trabajos, utilizando un enfoque predictivo-reactivo y un algoritmo metaheurístico de Pareto codicioso iterada reiniciada, logrando resolver eficazmente la reprogramación en un entorno multiobjetivo.

(Takeda-Berger et al., 2022) proponen un enfoque de programación de producción reactiva combinando un modelo de simulación basado en optimización y un algoritmo genético (AG). Consideran la disponibilidad de inventario como evento inesperado. La combinación del modelo de simulación y el AG permite que la función objetivo de la heurística de optimización dependa de los resultados de la simulación, como el número de trabajos tardíos y el tiempo medio de producción. Evalúan su enfoque en una empresa de fabricación de componentes automotrices, mostrando eficacia ante eventos estocásticos e interrupciones.

(Karas & Ozcelik, 2021) investigan el problema de asignación y reequilibrio de trabajadores en líneas de montaje (ALWARBP) interrumpidas por averías o paradas de estaciones de trabajo. Proponen un modelo MILP con una función multiobjetivo que incluye el cambio de tiempo de ciclo y el número de tareas trasladadas después del reequilibrio. También desarrollan un algoritmo de colonia artificial de abejas (ABC) que muestra un mayor rendimiento en instancias de gran tamaño en comparación con el MILP.

(Tighazoui et al., 2021) desarrollan una estrategia predictiva-reactiva para la reprogramación de máquinas paralelas idénticas, utilizan un modelo MILP y una metodología iterativa basada en la lógica de horizonte rodante. Consideran la llegada continua de trabajos y establecen una función multiobjetivo para evaluar la eficiencia del cronograma (TWWT) y su estabilidad (TWCTD).

(Hamzadayi & Yildiz, 2016) proponen un modelo de simulación y una estrategia de reprogramación, utilizan recocido simulado y reglas de despacho para resolver el problema de programación de máquinas paralelas dinámicas idénticas. Consideran la llegada de nuevos trabajos y la ejecución de operaciones de configuración dependientes de la secuencia por un solo servidor.

(Reddy et al., 2018) desarrollan un modelo de programación entera mixta no lineal (MINLP) para el problema de programación de taller de trabajo flexible (FJSP) con el objetivo de reducir la variación de la carga total de la máquina y el intervalo de reparación. Analizan el problema considerando las averías de las máquinas como eventos reales y proponen un algoritmo de optimización evolutiva basado en el aprendizaje con múltiples objetivos. Utilizan el tiempo medio entre fallos (MTBF) y el tiempo medio de reparación (MTTR) para evaluar el sistema durante una avería.

(Shen & Yao, 2015) Formulan un modelo matemático y un algoritmo evolutivo multiobjetivo para abordar el problema dinámico FJSP. Consideran eventos inesperados como la llegada de nuevos trabajos, averías y reparaciones de máquinas. Evalúan la capacidad de captura, tardanza y carga máxima de trabajo. Compararan su algoritmo con otros tres, demostrando un mejor desempeño.

(Wang et al., 2021) proponen una programación dinámica utilizando un método de aprendizaje por refuerzo profundo (DRL) para resolver el JSP, consideran averías de máquina y la llegada de nuevos trabajos como eventos responsables de desencadenar una reprogramación.

Tabla 1. Revisión Bibliográfica de Métodos de Solución

Autores	Objetivos	Eventos Inesperados	Características	Método de resolución
(Tighazoui & Sauvey, 2021)	Minimización del Total de Tiempos de Espera Ponderados (TWWT) y estabilidad del cronograma mediante la desviación total ponderada del tiempo de finalización (TWCTD).	Llegada de trabajos	Demanda variable, máquina única	Algoritmo metaheurístico codicioso iterado (IG)
(Liu et al., 2018)	Minimización del tiempo total más largo	Llegada de nuevos pedidos	Demanda variable, tiempo de configuración dependiente de la secuencia	Algoritmo metaheurístico codicioso iterado y método de reparación (elG_Rep)
(Janak et al., 2006)	Minimización de la variación del tiempo de finalización total	Cierre de unidades de procesamiento y modificaciones de pedidos	Máquinas paralelas idénticas, tiempo de configuración dependiente de la secuencia, lista de materiales, almacenamiento intermedio limitado	
(Fang & Xi, 1997)	Minimización de la duración del tiempo de finalización total y penalización por trabajos tardíos	Llegada de nuevos trabajos, averías de máquinas y fechas de vencimiento de trabajos	Demanda variable, Máquinas paralelas no idénticas, tiempo de configuración dependiente de la secuencia, penalización por trabajos tardíos	Estrategia de programación de horizonte móvil (RHS)
(Valledor et al., 2020)	Minimización del tiempo total más largo, tardanza ponderada total (TWT) y estabilidad del cronograma (TWCTD)	Averías de máquinas, cambios en los tiempos operativos y llegada de nuevos trabajos	Sin priorización de trabajos, sin superposición de operaciones consecutivas, almacenamiento intermedio ilimitado	Algoritmo metaheurístico de Pareto codicioso iterado (IPG)
(Takeda-Berger et al., 2022)	Minimización de la duración del tiempo de finalización total, trabajos tardíos	Disponibilidad de inventario	Priorización de trabajos, priorización de asignación de materiales, bloqueo de trabajos sin materia prima	Algoritmo genético (AG)
(Karas & Ozcelik, 2021)	Minimización del cambio de tiempo de ciclo y el número de tareas trasladadas entre estaciones tras el reequilibrio	Averías o paradas de estaciones de trabajo en líneas de montaje	Múltiples estaciones de trabajo, variación en tiempos de tareas	Algoritmo de colonia artificial de abejas (ABC)
(Tighazoui et al., 2021)	Eficiencia del cronograma (TWWT) y estabilidad del cronograma (TWCTD)	Llegada de nuevos trabajos	Demanda variable, máquinas paralelas idénticas, modificación de pedidos	Estrategia predictiva-reactiva iterativa basada en programación matemática
(Hamzadayi & Yildiz, 2016)	Minimización del tiempo total más largo	Llegada de nuevos trabajos	Demanda variable máquinas paralelas idénticas, servidor único para configuración, priorización de trabajos	Reprogramación con recocido simulado (SA) y reglas de despacho
(Reddy et al., 2018)	Minimización de la variación de la carga total de la máquina e intervalo de reparación	Averías de máquinas	Máquinas paralelas, almacenamiento intermedio ilimitado, tiempo de transporte entre máquinas excluido	Algoritmo evolutivo (EA)
(Shen & Yao, 2015)	Minimización del tiempo total más largo, medida de	Llegada de nuevos trabajos, averías y	Demanda variable, Máquinas Paralelas, almacenamiento intermedio ilimitado	Algoritmo evolutivo (EA)

	tardanza, carga de trabajo máxima de la máquina, estabilidad del cronograma	reparaciones de máquinas		
(Wang et al., 2021)	Minimización del tiempo total más largo	Averías de máquinas y llegada de nuevos trabajos	Máquinas específicas a cada trabajo	Aprendizaje por refuerzo profundo (DRL)

En la revisión de la literatura realizada sobre soluciones para la programación de producción y manejo de eventos inesperados, se han identificado diversos enfoques y métodos que abordan el problema desde distintas perspectivas. Entre las estrategias más destacadas se encuentran el uso de programación matemática para la optimización en entornos determinísticos y la aplicación de técnicas heurísticas para situaciones más dinámicas. Los modelos MILP y algoritmos de optimización estocástica ofrecen robustas soluciones para la programación inicial, mientras que los enfoques adaptativos y reactivos se centran en la flexibilidad ante eventos inesperados.

A través de la presente revisión, se ha observado una tendencia hacia el uso de enfoques híbridos que combinan técnicas de optimización con metodologías de ajuste dinámico. Sin embargo, persisten vacíos en la literatura en cuanto a la integración de estas técnicas con modelos de simulación que pueden ofrecer una visión más detallada del impacto de eventos en tiempo real.

Capítulo 4. Descripción de la Solución

El presente capítulo se centra en la formulación y metodología de solución para el problema de Programación de Producción sujeta a Eventos Inesperados. Este enfoque abarca desde la formulación matemática del modelo MILP, hasta la estrategia dinámica de reprogramación diseñada para adaptar el plan de producción en respuesta a condiciones variables del entorno. Inicialmente, se detalla el Modelo MILP, encargado de optimizar la asignación de recursos y definir los lotes de producción bajo condiciones normales y eventos inesperados como averías de máquinas y escasez de materias primas. Posteriormente, se determina la metodología de reprogramación, que utiliza un enfoque iterativo y reactivo ejecutando el MILP, con el objetivo de ajustar el plan de producción sin necesidad de rehacerlo por completo, asegurando así la eficiencia y adaptabilidad del sistema ante dichos eventos inesperados.

4.1 Modelo de Programación Lineal Entera Mixta (MILP)

A continuación, La tabla 2 presenta la nomenclatura utilizada para abordar el problema de Programación de Producción sujeta a Eventos Inesperados. Esta tabla incluye los índices, parámetros y variables de decisión clave relacionados con máquinas, materia prima y periodos de tiempo.

Tabla 2. Nomenclatura de MILP

Índices	
i	Índice de productos $i \in \{1, \dots, I\}$
j	Índice de máquinas $j \in \{1, \dots, J\}$
r	Índice de materia prima $r \in \{1, \dots, R\}$
t	Índice de periodos $t \in \{1, \dots, T\}$
Parámetros del modelo	
d_{it}	Demanda del producto i durante el periodo de tiempo t
ca_j	Capacidad de máquina j (unidades de tiempo)
req_{ij}	Tiempo requerido para producir una unidad de producto i en máquina j
cp_{ij}	Costo unitario de producción de producto i en máquina j
cc_j	Costo de configuración de la máquina j
st_{ij}	Tiempo de configuración del producto i en la máquina j
ur_{ri}	Tasa de materia prima r requerida para producir el producto i
$InvIp_i$	Inventario inicial del producto i
$InvIr_r$	Inventario inicial de la materia prima r
chp_i	Costo de mantenimiento del producto i
cd_i	Costo de diferimiento para el producto i
sm_{jt}	Estado de la máquina j en el período t (1 si está averiada o en tiempo de inactividad programada, 0 en caso contrario)
pb_{ijt}	Cantidad de producto i producida en la máquina j hasta el momento justo de la avería durante el período de tiempo t
p_{ijt}	1 si el producto i ha comenzado su producción en una máquina j diferente a la averiada en el período de tiempo t , 0 en caso contrario
Variables de decisión	
X_{ijt}	1 si el producto i se asigna a la máquina j en el período t , 0 en caso contrario
QP_{ijt}	Cantidad de producto i a producir en la máquina j en el período de tiempo t
BO_{it}	Cantidad diferida del producto i en el período de tiempo t
$InvFP_{it}$	Inventario final del producto i en el período de tiempo t
$InvFR_{rt}$	Inventario final de la materia prima r en el período de tiempo t
XR_{ijt}	1 si el producto i se asigna a la máquina j en el período de tiempo t y periodo de $t-1$, 0 en caso contrario
Z_{jt}	Variable auxiliar dependiente de XR_{ijt}

STT_{jt}	Tiempo total de configuración de la máquina j en período de tiempo t
$STMAX_{jt}$	Tiempo de configuración tomado del producto i con mayor tiempo entre los asignados en máquina j en periodos de tiempo $t-1$ y t
CFT_{jt}	Número total de configuraciones en la máquina j en el período de tiempo t

A continuación, se describe la formulación del modelo MILP para la programación de producción en talleres de trabajo.

La función objetivo minimiza los costos totales, incluyendo los costos de producción, costos de configuración, inventario de productos terminados y costos de pedidos pendientes.

Función objetivo

$$\sum_{i \in I} \sum_{j \in J} \sum_{t \in T} cp_{ij} \cdot QP_{ijt} + \sum_{i \in I} \sum_{t \in T} cd_i \cdot BO_{it} + \sum_{j \in J} \sum_{t \in T} cc_j \cdot CFT_{jt} + \sum_{i \in I} \sum_{t \in T} chp_i \cdot InvFP_{it} \quad (1)$$

Restricciones de inventario

Producto terminado:

$$InvFP_{it} = InvIp_i + \sum_{j \in J} QP_{ijt} - d_{it} + \sum_{j \in J} pb_{ijt} + BO_{it}, \quad \forall i, t = 1, \quad (2a)$$

$$InvFP_{it} = InvFP_{i,t-1} + \sum_{j \in J} QP_{ijt} - d_{it} + \sum_{j \in J} pb_{ijt} + BO_{it} - BO_{i,t-1}, \quad \forall i, t \geq 1. \quad (2b)$$

Las ecuaciones de balance de inventario (2a) y (2b) garantizan los valores apropiados para las existencias, las cantidades a producir, y los pedidos pendientes de entregar (demanda diferida) para cada período de tiempo $t = 1$ y $t \geq 1$. Ambas restricciones tienen en cuenta las cantidades producidas hasta el momento justo en el que se genera una avería en máquina j , en caso de que se suceda.

Materia prima:

$$InvFR_{rt} = InvIr_r - \sum_{i \in I} \sum_{j \in J} ur_{ri} \cdot QP_{ijt}, \quad \forall r, \forall t = 1, \quad (3a)$$

$$InvFR_{rt} = InvFR_{r,t-1} - \sum_{i \in I} \sum_{j \in J} ur_{ri} \cdot QP_{ijt}, \quad \forall r, \forall t \geq 1. \quad (3b)$$

$$\sum_{i \in I} \sum_{j \in J} ur_{ri} \cdot QP_{ijt} \leq InvIr_r, \quad \forall r, \forall t. \quad (4)$$

Las restricciones (3a) y (3b) aseguran que los niveles de inventario de materia prima sean precisos para cada período de tiempo $t = 1$ y $t \geq 1$, teniendo en cuenta el consumo planificado de materia prima. La restricción (4) garantiza que el consumo de materia prima r se mantenga dentro de los límites establecidos por el inventario inicial disponible.

Restricción de cumplimiento global de la demanda

$$\sum_{j \in J} \sum_{t \in T} QP_{ijt} + InvIp_i = \sum_{t \in T} d_{it} - \sum_{j \in J} \sum_{t \in T} pb_{ijt}, \quad \forall i. \quad (5)$$

La restricción (5) garantiza el cumplimiento de la demanda global, asegura que la cantidad producida de producto i durante los períodos de tiempo t sea suficiente. También considera el ajuste de la producción según la cantidad de producto i ya producida hasta el momento en que ocurre una avería en máquina j (pb_{ijt}).

Restricción de capacidad y avería de maquinaria

$$\sum_{i \in I} X_{ijt} = 0, \quad \forall j, \forall t, \quad sm_{jt} = 1, \quad (6a)$$

$$\sum_{i \in I} req_{ij} \cdot QP_{ijt} + STT_{jt} \leq ca_j, \quad \forall j, \forall t, \quad sm_{jt} = 0. \quad (6b)$$

La restricción (6a) determina que el producto i no sea asignado a la máquina j en el período de tiempo t en caso de avería ($sm_{jt}=1$) cuando se necesite hacer una reprogramación. En caso contrario, cuando la máquina j está operativa ($sm_{jt}=0$), la

restricción (6b) garantiza que el tiempo total requerido para la producción y configuración de cada producto i no exceda la capacidad de tiempo disponible de la máquina j en el período de tiempo t .

Restricciones de cálculo de productos que repiten su producción

$$XR_{ijt} \leq X_{ijt}, \quad \forall i, \forall j, \forall t \geq 1, \quad (7)$$

$$XR_{ijt} \leq X_{ij,t-1}, \quad \forall i, \forall j, \forall t \geq 1, \quad (8)$$

$$XR_{ijt} \geq X_{ijt} + X_{ij,t-1} - 1, \quad \forall i, \forall j, \forall t \geq 1. \quad (9)$$

Las restricciones (7), (8) y (9) indican si un producto i es asignado o producido en la máquina j en el período de tiempo $t-1$ y t , registrando una producción en periodos consecutivos.

Restricciones de configuración

$$STT_{jt} = \sum_{i \in I} st_{ij} \cdot X_{ijt}, \quad \forall j, \forall t = 1, \quad (10)$$

$$STT_{jt} \geq \sum_{i \in I} st_{ij} \cdot X_{ijt} - STMAX_{jt}, \quad \forall j, \forall t \geq 1, \quad (11)$$

$$STT_{jt} \leq \sum_{i \in I} st_{ij} \cdot X_{ijt} - STMAX_{jt}, \quad \forall j, \forall t \geq 1. \quad (12)$$

La restricción (10) determina el tiempo total de configuración para la máquina j durante el período de tiempo $t = 1$. Las restricciones (11) y (12) establecen límites para el tiempo de configuración en períodos de tiempo $t \geq 1$, descontando el valor máximo de tiempo de configuración de uno de los productos i que repiten su asignación en la máquina j en el periodo de tiempo t ($XR_{ijt} = 1$).

$$STMAX_{jt} \geq st_{ij} \cdot XR_{ijt}, \quad \forall i, \forall j, \forall t \geq 1, \quad (13)$$

$$STMAX_{jt} \leq \sum_{i \in I} st_{ij} \cdot XR_{ijt}, \quad \forall j, \forall t \geq 1. \quad (14)$$

La restricción (13) asegura que $STMAX_{jt}$ capture el mayor tiempo de configuración entre los productos que se producen en la máquina j durante períodos de tiempo $t - 1$ y t

($XR_{ijt} = 1$). La restricción (14) garantiza que $STMAX_{jt}$ no supere el mayor tiempo de configuración obtenido entre el producto i o los productos i asignados.

$$CFT_{jt} = \sum_{i \in I} X_{ijt} \quad \forall j, \forall t = 1 \quad (15)$$

$$CFT_{jt} = \sum_{i \in I} X_{ijt} - Z_{jt} \quad \forall j, \forall t \geq 1 \quad (16)$$

La restricción (15) determina el número de configuraciones totales en la máquina j en periodo de tiempo $t = 1$.

La restricción (16) calcula las configuraciones totales en periodos $t \geq 1$, descontando una configuración para reflejar el ahorro cuando no se requiere un cambio adicional efectuado por un producto asignado en periodo de tiempo $t-1$ y t .

Limites (Bounds)

$$QA_{ijt} \leq M \cdot X_{ijt}, \quad \forall i, \forall j, \forall t \quad (17)$$

$$X_{ijt} \leq M \cdot QA_{ijt}, \quad \forall i, \forall j, \forall t \quad (18)$$

$$Z_{jt} \leq M \cdot \sum_{i \in I} XR_{ijt}, \quad \forall j, \forall t \quad (19)$$

La restricción (17) garantiza que un producto i sea producido en una máquina j durante el período de tiempo t si está asignado a esa máquina. La restricción (18) establece que el producto i no puede ser asignado a una máquina j durante un período de tiempo t si no se determina ningún tamaño de lote de producción. La restricción (19) determina el valor de la variable auxiliar, activándose únicamente cuando al menos un producto i se asigna a una máquina j en periodos de tiempo $t-1$ y t .

Naturaleza de variables binarias:

$$X_{ijt}, XR_{ijt} \in \{0,1\}, \quad \forall i, \forall j, \forall t, \quad (20)$$

$$Z_{jt} \in \{0,1\}, \quad \forall j, \forall t. \quad (21)$$

La restricción (20) y (21) indican la naturaleza binaria de las variables X_{ijt} que representan la asignación de productos; XR_{ijt} producto asignado en periodos consecutivos y la variable auxiliar Z_{jt} .

Naturaleza de variables Tipo entero (N):

$$QP_{ijt} \in \mathbb{N}, \quad \forall i, \forall j, \forall t, \quad (22)$$

$$BO_{it} \in \mathbb{N}, \quad \forall i, \forall t, \quad (23)$$

$$STT_{jt}, \\ STMAX_{jt}, CFT_{jt} \in \mathbb{N}, \quad \forall j, \forall t, \quad (24)$$

$$InvFP_{it} \in \mathbb{N}, \quad \forall i, \forall t, \quad (25)$$

$$InvFR_{rt} \in \mathbb{N}, \quad \forall r, \forall t. \quad (26)$$

Las restricciones (22), (23), (24), (25) y (26) indican la naturaleza continua de las variables representadas.

4.2 Estrategia de Reprogramación

La reprogramación se encarga de actualizar el plan de producción inicial en respuesta a interrupciones consideradas en el estudio del problema de Programación de Producción sujeto a Eventos Inesperados.

La estrategia de reprogramación abordada en el presente caso de estudio utiliza el enfoque basado en los aspectos clave para la reprogramación identificados por (Vieira et al., 2003):

- Entorno de Reprogramación: Se enfoca en la programación inicial y reprogramación para entornos de talleres de trabajo con máquinas paralelas no idénticas, sujetos a eventos inesperados como la escasez de materias primas o averías de maquinaria.
- Estrategia de Reprogramación: Se adopta una estrategia reactiva, actualizando el programa de producción en respuesta a eventos inesperados que afectan el entorno de producción.
- Método de Reprogramación: Se implementa el enfoque de reprogramación parcial, que se modifica solo una parte del plan de producción inicial mediante la fijación de variables de decisión en el MILP. Este enfoque mantiene inalteradas las partes del programa que no pueden ser cambiadas, logrando así ajustes menores sin rehacer el plan por completo.

Para implementación del enfoque de reprogramación parcial, se requiere un método de ejecución que permita ajustar el plan de en respuesta a eventos inesperados. Este método debe ser capaz de resolver el MILP y aplicar las modificaciones necesarias sin rehacer el plan completo. Con este objetivo, se propone un código iterativo encargado de resolver el MILP, el cual sigue una secuencia determinada por varias fases, cada una encargada de asegurar la adaptabilidad y eficiencia del programa de producción frente a eventos inesperados. A continuación, se describen brevemente las fases:

I. Fase de Cálculo del Plan Inicial:

En la presente fase, se calcula un plan de producción inicial utilizando el MILP. Este plan incluye la asignación óptima de lotes de producción a las máquinas, considerando los recursos disponibles en el taller de trabajo.

II. Fase de Monitoreo del Entorno de Producción:

Tras el cálculo del plan de producción inicial, se lleva a cabo un monitoreo continuo del entorno de producción para detectar cualquier cambio asociado a los eventos inesperados, como averías de maquinaria o escasez de materias primas.

III. Fase de Reprogramación Dinámica:

Se identifican las partes del programa de producción inicial que no pueden ser modificadas. Las variables de decisión correspondientes a estas partes del programa son fijadas para que permanezcan inalteradas en el proceso de reprogramación.

Durante la fase de reprogramación dinámica, se identifican las partes del plan de producción inicial que deben permanecer inalteradas, como la producción de productos que ya han comenzado en caso de averías, o diferimientos (pedidos pendientes a entregar) de producción en caso de escasez de materia prima.

Las variables de decisión asociadas a estas partes (QP_{ijt} representa la cantidad de producción y BO_{it} , que indica la cantidad de pedido pendiente) se fijan para garantizar que su asignación y cantidad permanezcan constantes. El modelo se recalcula para ajustar el resto del plan según las nuevas condiciones del taller.

Esta estrategia de fijar variables puede incluir, por ejemplo, la producción de ciertos productos que ya han comenzado y no pueden interrumpirse. Finalmente, se recalcula el programa de producción para tener en cuenta los cambios detectados, mientras se mantiene inalterado lo que no puede ser modificado. Este proceso de reprogramación dinámica utiliza nuevamente el MILP para encontrar una nueva solución óptima o casi óptima que se adapte a

las nuevas condiciones del entorno de producción, minimizando los cambios significativos respecto al plan de producción inicial.

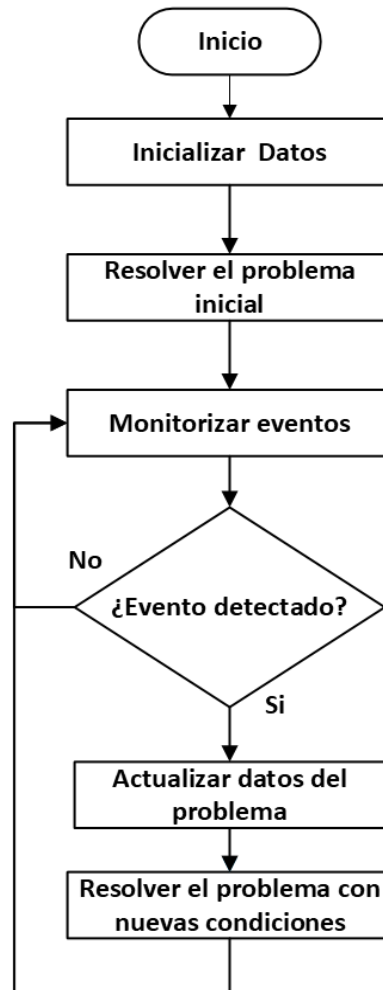


Figura 3. Estrategia iterativa-reactiva de Reprogramación

La Figura 6 ilustra el proceso operativo de la programación inicial y reprogramación, estructurado en ciclos iterativos. Este enfoque implica la resolución de la programación inicial, monitorización continua del entorno de producción, la detección de eventos y la reprogramación en respuesta a cambios. Esta naturaleza iterativa y reactiva es fundamental para adaptar el plan de producción según las condiciones variables del entorno.

El sistema está diseñado para ser adaptable y receptivo mediante el uso de funciones denominadas hilos (threading). Un hilo se encarga de la detección continua de eventos en segundo plano, permitiendo que el programa principal permanezca activo y receptivo

a cambios sin interrupciones mientras espera la detección de condiciones del entorno (ver Anexo 1, pág. 64).

Estos eventos inesperados están representados por los parámetros sm_{jt} (avería de máquina) y mpf (escasez de materia prima), aseguran que el sistema pueda detectar rápidamente dichos eventos y realizar nuevamente el cálculo de reprogramación. En el código iterativo, cuando se genera un evento asociado a avería de máquina, sm_{jt} toma valor 1, indicando la señal asociada a una avería de máquina y permitiendo su detección actualizando el cambio de estado de la máquina, mientras que mpf se detecta cuando su valor es mayor a 1, indicando la escasez y el período de tiempo en el que se contara nuevamente con la materia prima.

Políticas de Reprogramación

La funcionalidad de este sistema iterativo-reactivo se ve complementada por las políticas de reprogramación, que guían cómo y cuándo se realizan los ajustes necesarios. Estas políticas establecen los criterios y procedimientos para implementar cambios en el plan de producción, asegurando que las decisiones tomadas sean coherentes y eficientes. A continuación, se describen las políticas de reprogramación establecidas:

Máquina Averiada:

- La reprogramación se ejecuta únicamente cuando una máquina está fuera de servicio por un período prolongado.
- Los productos asignados a máquinas operativas que ya han comenzado su producción en el momento de la avería en otra máquina ($p_{ijt} = 1$) se establecen como inamovibles. Esto significa que su asignación y cantidad a producir, calculadas en el plan de producción inicial (QP_{ijt}), se mantienen fijas durante el cálculo de reprogramación.

Escasez de Materia Prima:

- Se realiza la reprogramación cuando ocurre un faltante de materia prima causado por un retraso en el envío por parte del proveedor.

- Se determina la cantidad proporcional de producto terminado que se puede obtener a partir de la cantidad de materia prima faltante, fijando esta cantidad como pedido pendiente (BO_{it}) y posponiendo su producción al período siguiente, cuando se espera recibir la materia prima faltante.

Capítulo 5. Experimentos

En este capítulo, se presentan los experimentos realizados para evaluar el desempeño del enfoque propuesto basado en un MILP para la programación de producción en talleres sujetos a eventos inesperados. El objetivo de estos experimentos es medir la eficiencia del modelo en términos de tiempo de resolución y adaptación a cambios en el entorno de producción.

El enfoque consiste en calcular un plan de producción inicial y, posteriormente, ejecutar una reprogramación dinámica en respuesta a eventos inesperados, como averías de maquinaria o escasez de materias primas. Los experimentos para evaluar la programación de producción inicial se llevaron a cabo utilizando instancias pequeñas y medianas para analizar la eficiencia temporal del modelo. Posteriormente, se realizaron experimentos adicionales para evaluar la capacidad del sistema para adaptarse dinámicamente a estos eventos inesperados.

El modelo MILP y el código asociado a la estrategia iterativa de reprogramación se han desarrollado en Python, utilizando el lenguaje de modelado Pyomo. Para la resolución del modelo, se empleó el solver Gurobi. Las pruebas se realizaron en un ordenador con 8 GB de RAM, ejecutando el sistema operativo Windows 10 Enterprise de 64 bits.

5.1 Generación de Datos

Para llevar a cabo los experimentos del presente trabajo, se ha desarrollado un generador de datos aleatorios programado en Python utilizando la biblioteca NumPy. Este generador permite simular diferentes tamaños de instancias para el problema abordado, siguiendo un conjunto de límites y parámetros predefinidos que varían según las pruebas y el tamaño de la instancia.

El proceso de generación de datos se describe a continuación:

I. Parámetros Definidos:

Se definen los límites de valores que pueden tomar los parámetros. Estos parámetros son:

- Productos: Número mínimo y máximo de productos.
- Máquinas y Períodos: Número de máquinas y períodos considerados en la planificación.
- Tiempos de Procesamiento y Configuración: Límites para los tiempos de procesamiento y configuración de las máquinas.
- Costos: Límites para los costos de configuración, producción y pedidos pendientes (diferimiento de demanda), así como los costos de inventario.
- Inventario Inicial: Límites para el inventario inicial de productos y materias primas.
- Demanda: Límites para la demanda de productos en cada período.

II. Generación de Datos Aleatorios:

Se generan datos aleatorios dentro de los rangos definidos, asegurando una representación adecuada de distintas situaciones y escenarios. Estos datos son utilizados para crear instancias representativas del problema, permitiendo la evaluación del modelo bajo diversas condiciones.

III. Almacenamiento de Datos:

Los datos generados se almacenan en archivos de texto separados para cada conjunto de parámetros, siguiendo un formato tabular.

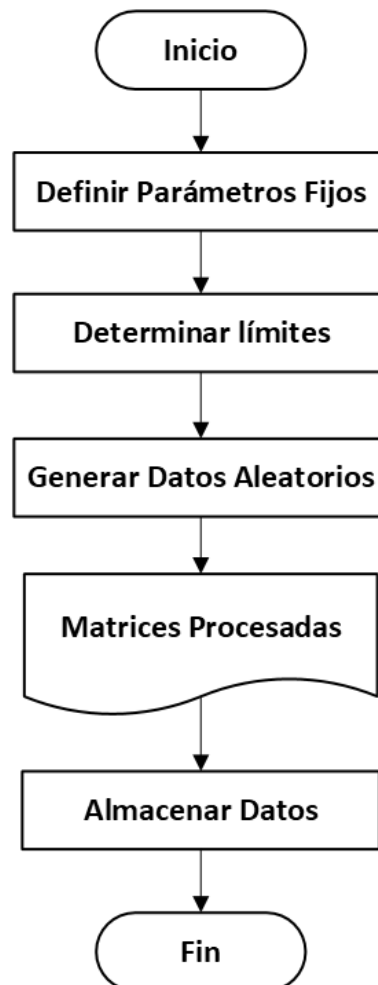


Figura 4. Generación de Datos

5.2 Experimentos De Programación Inicial

En esta sección, se presentan los resultados de los experimentos realizados con instancias pequeñas y medianas, diseñadas para evaluar el rendimiento del modelo en el cálculo de la programación inicial. En primer lugar, se proporciona unas tablas ejemplares que muestran los parámetros específicos a utilizar para la resolución de una instancia pequeña. Posteriormente, se ofrece un resumen general de los resultados experimentales obtenidos para las diferentes instancias utilizadas en los experimentos, incluyendo los tiempos de resolución (ver Tabla 14, pág. 45).

A continuación, se muestran los datos empleados para la ejecución de la primera instancia:

Tabla 3. Experimentos De Programación Inicial: Tamaño de instancia

i	j	r	t
1	1	1	1
2	2	2	2
3	-	3	3

Tabla 4. Experimentos De Programación Inicial: Inventario inicial, coste de mantenimiento, coste de diferir

i	$InvIp_i$	chp_i	cd_i
1	0	9	40
2	0	9	32
3	0	8	37

Tabla 5. Experimentos De Programación Inicial: Demanda

i	t	d_{it}
1	1	20
1	2	10
1	3	13
2	1	15
2	2	10
2	3	20
3	1	16
3	2	13
3	3	0

Tabla 6. Experimentos De Programación Inicial: Datos de máquinas

j	ca_j	cc_j
1	100	32
2	100	43

Tabla 7. Experimentos De Programación Inicial: Datos de producción

i	j	cp_{ij}	req_{ij}	st_{ij}
1	1	2	2	8
1	2	3	3	10
2	1	2	2	15
2	2	1	2	8
3	1	2	2	5
3	2	2	3	10

Tabla 8. Experimentos De Programación Inicial: Inventario inicial de materia prima

r	$InvIr_r$
1	500
2	500
3	500

Tabla 9. Experimentos De Programación Inicial: Lista de materiales y tasa de consumo

r	i	ur_{ri}
1	2	2
2	1	1
2	2	2
3	2	2
3	3	2

Resultados de programación inicial

A continuación, se presentan los resultados obtenidos del primer cálculo de programación inicial. La Tabla 10 muestra la asignación de productos a máquinas y los tamaños de lotes de producción por período de tiempo (Plan de producción). La Tabla 11 detalla los costos y tiempos de configuración de las máquinas. La Tabla 12 proporciona información sobre los inventarios de productos terminados y los pedidos pendientes a entregar. La Tabla 13 ilustra los inventarios de materia prima disponibles en cada período de tiempo. Estas tablas ofrecen una visión general de la planificación de producción obtenida mediante el cálculo de programación utilizando el modelo MILP.

Tabla 10. Resultados De Programación Inicial: Plan inicial de producción

i	j	t	X_{ijt}	XR_{ijt}	QP_{ijt}
1	1	1	1	0	20
1	1	2	1	1	10
1	1	3	1	1	13
2	2	1	1	0	15
2	2	2	1	1	10
2	2	3	1	1	20
3	1	1	1	0	16
3	1	2	1	1	13
3	1	3	0	0	0

Tabla 11. Resultados De Programación Inicial: Configuración de máquinas

j	t	CFT_{jt}	$STMAX_{jt}$	STT_{jt}
1	1	2	0	17
1	2	1	9	8
1	3	1	9	8
2	1	1	0	8
2	2	0	8	0
2	3	0	8	0

Tabla 12. *Resultados De Programación Inicial: Inventarios de producto terminado, pedidos pendientes*

i	t	$InvFP_{it}$	BO_{it}
1	1	0	0
1	2	0	0
1	3	0	0
2	1	0	0
2	2	0	0
2	3	0	0
3	1	0	0
3	2	0	0
3	3	0	0

Tabla 13. *Resultados De Programación Inicial: Inventarios de materia prima*

r	t	$InvFR_{rt}$
1	1	460
2	2	440
3	3	414
1	1	450
2	2	420
3	3	367
1	1	438
2	2	392
3	3	352

A continuación, se presenta una tabla que resume los resultados obtenidos para las demás instancias, incluyendo información sobre la cantidad de productos, máquinas, períodos, y los tiempos de resolución. Las instancias pequeñas están representadas por P (P1, P2, P3, ...), mientras que las instancias medianas se indican con M (M1, M2, M3, ...). Cabe destacar que estos cálculos corresponden únicamente a la programación inicial.

Tabla 14. Resultados experimentales de programación inicial

Instancias	i	j	r	t	Número de restricciones	Número de variables binarias	Número de variables enteras	Número de nonzeros	Tiempo(seg.)
P1	3	2	3	3	146	42	103	433	0.0120
P2	8	5	8	3	745	255	487	2785	0.0599
P3	10	5	10	7	2335	735	1395	8615	0.2500
M1	13	10	13	14	11769	3780	6556	44783	11.2390
M2	24	14	24	14	29396	9604	15890	117486	73.8960
M3	32	16	32	16	50688	16640	27120	210512	342.5230

El análisis de los resultados experimentales revela una tendencia clara en el comportamiento del modelo con respecto al tamaño de las instancias. A través de la Tabla 14 Se observa que el tiempo de resolución aumenta de manera significativa a medida que se incrementa el tamaño del problema, desde 0.012 segundos para la instancia más pequeña (P1) hasta 342.523 segundos para la instancia más grande (M3). Este crecimiento en el tiempo de procesamiento refleja la creciente complejidad del modelo con el aumento en el número de productos, máquinas, materia prima y períodos. Además, el número de variables enteras y binarias también aumenta considerablemente con el tamaño del problema, lo cual sugiere una mayor dificultad de encontrar soluciones óptimas a medida que el modelo se vuelve más complejo.

El tiempo de resolución para la instancia M3 puede parecer elevado, pero se considera razonable dado el tamaño de la instancia, que incluye 32 productos, 16 máquinas y 16 períodos de tiempo. Este tiempo de resolución es aceptable para problemas de gran escala como M3, donde la complejidad del modelo, el número de variables y restricciones juegan un papel importante en el tiempo total requerido. Sin embargo, para instancias aún mayores, el tiempo de resolución podría incrementarse considerablemente, lo que podría demandar mayores recursos computacionales y optimizaciones adicionales en el modelo. En tales casos, sería recomendable explorar estrategias de mejora, como el ajuste de parámetros, la simplificación de la formulación del problema, o el uso de hardware más potente, para asegurar una resolución eficiente y efectiva.

El modelo demuestra un rendimiento satisfactorio en las instancias pequeñas y medianas. Los tiempos de resolución para estos casos son razonables y adecuados considerando la escala y la complejidad del problema. Para estas instancias, el modelo maneja eficazmente la resolución, ofreciendo tiempos de procesamiento aceptables en función del tamaño y la complejidad de los problemas.

5.3 Experimentos de Reprogramación Bajo Eventos Inesperados

En esta sección se llevan a cabo los experimentos relacionados con la presencia de eventos inesperados, como averías de maquinaria y escasez de materias primas. Se aborda la reprogramación para ajustar el plan de producción a las condiciones actuales del taller.

Los experimentos se dividen en dos escenarios: En primer lugar, se realiza el cálculo de reprogramación en respuesta a avería de máquina. Posteriormente, se analiza la reprogramación debido a la escasez de materia prima.

Para facilitar la comprensión y el análisis de los resultados, se utilizan instancias pequeñas para el cálculo de reprogramación, lo que permite una mejor visualización y manejo de los datos.

A continuación, se presentan los experimentos realizados, comenzando con la reprogramación en respuesta a averías de maquinaria, seguida de la reprogramación debida a la escasez de materia prima.

5.3.1 Reprogramación en Respuesta a Averías de Maquinaria

Para abordar el planteamiento de averías de máquinas, se considera un escenario posterior a la programación inicial, en el que ya se ha obtenido un plan de producción previo.

A continuación, se presenta el plan de producción inicial, el cual actuara como base realizar la reprogramación de la producción:

Tabla 15. Experimento de reprogramación (Avería de máquina): Plan de Producción

i	j	t	X_{ijt}	QP_{ijt}
1	1	1	1	40
1	1	2	1	30
1	1	3	1	15
2	2	1	1	32
2	2	2	1	45
2	2	3	1	30
3	3	1	1	31
3	3	2	1	25
3	3	3	1	25
4	3	1	1	25
4	3	2	1	19
4	3	3	0	0

Para simular la avería de una máquina, se define la siguiente tabla:

Tabla 16. Experimento de reprogramación (Avería de máquina): Estado de las máquinas

j	t	sm_{ij}
1	1	0
1	2	0
1	3	0
2	1	1
2	2	0
2	3	0
3	1	0
3	2	0
3	3	0

Esta Tabla muestra que en el período de tiempo $t=1$, se produce una avería en la máquina $j=2$, la cual permanecerá inhabilitada durante el resto del período.

A continuación, se presentan los productos cuya producción ha comenzado en otras máquinas en el momento en que se generó la avería, por lo que se consideran inamovibles (ver Tabla 17). Esta condición es crucial para la reprogramación, ya que permite reducir los cambios respecto al plan de producción inicial, evitando así costos adicionales y perturbaciones en el flujo de trabajo.

Tabla 17. Experimento de reprogramación (Avería de máquina): Productos Inamovibles

i	j	t	p_{ijt}
1	1	1	1
2	2	1	0
3	3	1	0
4	3	1	1

Finalmente, se determina la cantidad producida del producto $i=2$, justo en el momento de la avería en la máquina $j=2$, en el periodo de tiempo $t=1$

Tabla 18. Experimento de reprogramación (Avería de máquina): Cantidad de Producto Producida en Máquina Averiadada

i	j	t	pb_{ijt}
1	1	1	0
2	2	1	25
3	3	1	0
4	3	1	0

Resultados Obtenidos:

Tras definir los parámetros de entrada relacionados con la avería de la máquina, se plantean escenarios para ejecutar el cálculo de reprogramación. En primer lugar, se generó un plan de producción sin considerar los parámetros pb_{ijt} y p_{ijt} , limitándose a conocer únicamente el estado de la máquina (sm_{ij}). Posteriormente, se añadieron dichos parámetros para evaluar su impacto en el plan de producción. A continuación, se presenta la comparativa que refleja los resultados de ambos enfoques:

Tabla 19. Experimento de reprogramación (Avería de máquina): Resultado comparativo de plan de producción

<i>i</i>	<i>j</i>	<i>t</i>	Plan de Producción	Plan de Producción
			Sin Parámetros	Con Parámetros
			pb_{ijt} y p_{ijt}	pb_{ijt} y p_{ijt}
1	1	1	54	55
1	1	2	31	30
1	1	3	15	15
2	1	1	10	7
2	2	1	-	-
2	3	1	16	-
2	2	2	51	45
2	2	3	30	30
3	3	1	53	55
3	3	2	37	35
3	3	3	25	25
4	3	1	38	48
4	3	2	29	19

Tabla 20. Experimento de reprogramación (Avería de máquina): Resultados comparativos de pedidos pendientes por entregar

i	t	Pedidos pendientes a entregar (BO_{it})	Pedidos pendientes a entregar (BO_{it})
		Sin Parámetros	con Parámetros
		pb_{ijt} y p_{ijt}	pb_{ijt} y p_{ijt}
1	1	1	0
1	2	0	0
1	3	0	0
2	1	10	0
2	1	6	0
2	1	0	0
2	2	0	0
2	3	2	0
3	1	0	0
3	2	0	0
3	3	10	0
4	1	0	0
4	2	0	0

Ambas tablas mostradas con anterioridad muestran las diferencias que se obtienen en la reprogramación cuando no se tienen en cuenta ciertos parámetros dinámicos. La tabla 16 ilustra los planes de producción obtenidos en ambos casos y destaca las diferencias en los lotes de productos a producir y las respectivas asignaciones a máquinas, evitando que se asignen productos a la máquina averiada. La tabla 20 proporciona una comparativa de los productos determinados por el cálculo de reprogramación como pedidos pendientes a entregar, donde se puede observar que en el primer escenario se difieren ciertos productos para ser producidos en el próximo periodo, evidenciando cómo los parámetros actualizados contribuyen a evitar cambios significativos en el plan de producción que pueden causar costos de configuración de máquinas adicionales y costos por diferimientos de demanda.

Contar con datos de entrada actualizados sobre el estado del taller es crucial para que las reprogramaciones sean realistas y efectivas, como se observa en el segundo escenario en ambas tablas.

5.3.2 Reprogramación en Respuesta a Escasez de Materias Primas

En la presente sección, se aborda el cálculo de reprogramación sujeto a escasez de materia prima, mostrando una serie de resultados obtenidos tras dicho cálculo. Cabe destacar que, para abordar el evento, se deben considerar los siguientes supuestos:

- La escasez de materia prima resulta de retrasos en los períodos de entrega preestablecidos por el proveedor.
- Los retrasos en la entrega de materia prima están limitados a un solo período.
- Se supone que cada tipo de materia prima está asignado exclusivamente a un solo producto, de modo que ningún tipo de materia prima se comparte entre diferentes productos. En caso de escasez, esta afecta únicamente al producto específico asociado con la materia prima en cuestión, y no a otros productos.

Para la ejecución del cálculo de reprogramación sujeto a escasez de materia prima, se cuentan con los siguientes datos:

Tabla 21. Experimento de reprogramación (Escasez de materia prima): Plan de producción inicial

i	j	t	QP_{ijt}
1	1	1	55
1	1	2	30
1	1	3	0
2	2	1	32
2	2	2	45
2	2	3	30
3	3	1	55
3	3	2	36
3	3	3	25
4	3	1	48
4	3	2	20
4	3	3	0

Tabla 22. Experimento de reprogramación (Escasez de materia prima): Lista de materiales

r	i	ur_{ri}
1	1	2
2	2	2
3	3	2
4	4	2

Tabla 23. Experimento de reprogramación (Escasez de materia prima): Inventario inicial de materia prima

r	$InvIr_r$
1	170
2	214
3	232
4	136

La Tabla 22 muestra la lista de materiales asociada a cada producto, y la Tabla 23 ilustra el inventario inicial de la materia prima con el que se cuenta para cumplir con el plan de producción. Cabe destacar que, para el planteamiento del problema, la programación inicial (ver Tabla 21) se realiza suponiendo que se cuenta con toda la materia prima en el inventario inicial, aunque en realidad aún haya materia prima por llegar. Este enfoque permite establecer un escenario donde haya faltantes de materia prima retrasada para el modelado del evento de escasez de materias primas. En este caso, se consideran faltantes de materia prima $r = 1,4$, asociadas a la producción de los productos $i = 1,4$.

La siguiente tabla muestra los datos sobre el nuevo período de tiempo de llegada de la materia prima, y su respectiva cantidad.

Tabla 24. Experimento de reprogramación (Escasez de materia prima): Materia prima faltante

r	t	<i>Cantidad de Materia Prima</i>
1	-	60
2	-	-
3	3	-
4	3	40

Resultados Obtenidos:

En las Tablas 25 y 26 se puede observar cómo se modifica el plan de producción cuando se presenta la escasez de materia prima, consecuencia de retrasos causados por el proveedor. Al no contar con la materia prima suficiente de un producto, rápidamente sus valores se fijan como pedidos pendientes a entregar, para ser producidos en el período de tiempo siguiente, cuando se espera que llegue la materia prima necesaria. Esta situación requiere que el modelo MILP y al sistema de reprogramación a reajusten las asignaciones de producción, convirtiéndolo en un sistema adaptable y dinámico.

Tabla 25. Experimento de reprogramación (Escasez de materia prima): Plan de producción

i	j	t	QP_{ijt}	QP_{ijt}
1	1	1	55	40
1	1	2	30	0
1	1	3	0	30
2	2	1	32	32
2	2	2	45	45
2	2	3	30	30
3	3	1	55	55
3	3	2	36	36
3	3	3	25	25
4	3	1	48	48
4	3	2	20	0
4	3	3	0	20

Tabla 26. Experimento de reprogramación (Escasez de materia prima): Pedidos pendientes a entregar

i	t	Antes del evento inesperado BO_{it}	Después del evento inesperado BO_{it}
3	1	0	40
3	2	0	0
3	3	0	0
4	1	0	0
4	2	0	0
4	3	0	20

El análisis de los experimentos realizados demuestra la efectividad y adaptabilidad del modelo MILP y la estrategia iterativa de reprogramación para el Problema de Programación de Producción en Talleres Sujetos a Eventos Inesperados. Inicialmente, se observa que el modelo maneja satisfactoriamente las instancias pequeñas y medianas, con tiempos de resolución razonables. Sin embargo, en instancias más grandes, el tiempo de resolución aumenta considerablemente, lo que sugiere la necesidad de optimizaciones adicionales. Mediante la estrategia de reprogramación, se demostró que, en respuesta a averías de maquinaria, el sistema de reprogramación ajusta satisfactoriamente el plan de producción. Además, incluir parámetros actualizados al momento de reprogramar evita asignaciones erróneas y minimiza cambios significativos en el plan de producción, reduciendo costos adicionales que pueden generarse en consecuencia. Asimismo, se demostró que el sistema de reprogramación ajusta satisfactoriamente el plan de producción frente a la escasez de materia prima, determinando pedidos pendientes a entregar que requieren ajustar las asignaciones de producción.

En conjunto, estos resultados destacan la importancia de contar con sistemas de reprogramación adaptables y basados en datos actualizados. La capacidad de estos sistemas para ajustarse a imprevistos y cambios en las condiciones operativas es esencial

para mantener la continuidad en los planes de producción. Un sistema de reprogramación eficaz no solo permite gestionar mejor los eventos inesperados, como averías de maquinaria y escasez de materia prima, sino que también ayuda a minimizar los costos adicionales derivados de interrupciones y ajustes no planificados, garantizando así una operación más fluida y económica del taller.

Capítulo 6. Conclusiones

El presente trabajo ha demostrado la efectividad del modelo MILP y la estrategia iterativa de reprogramación para resolver el Problema de la Programación de Producción en Talleres Sujetos a Eventos Inesperados, logrando tiempos de resolución razonables y minimizando los costos de producción. Los resultados indican que el modelo funciona satisfactoriamente en instancias pequeñas y medianas, aunque se requieren optimizaciones adicionales para manejar instancias más grandes de manera eficiente. La reprogramación ha mostrado ser dinámica frente a eventos inesperados, evitando grandes cambios en el plan de producción inicial y manteniendo la eficiencia operativa.

Sin embargo, este estudio no está exento de limitaciones, se identifican áreas que requieren mayor atención y desarrollo para mejorar el desempeño y la aplicabilidad del modelo en escenarios más complejos. En particular, la eficiencia computacional en instancias mayores es una limitación que podría abordarse mediante la aplicación de técnicas metaheurísticas como algoritmos genéticos, búsqueda tabú y recocido simulado. Estas técnicas pueden proporcionar soluciones iniciales que optimicen el proceso de resolución del MILP.

Otro aspecto importante que abre brecha de investigación es el análisis del comportamiento de la variable de decisión $STMAX_{jt}$ en escenarios donde la capacidad de las máquinas se aproxima a su límite. Esta variable en ocasiones no refleja con precisión el tiempo de configuración, lo que puede llevar a estimaciones imprecisas del tiempo total de configuración de una máquina. Mejorar esta precisión es esencial para brindar estimaciones más fiables y mejorar el modelo en su totalidad. De igual manera, otra área de estudio es permitir que el sistema maneje múltiples tipos de materia prima en la lista de materiales de un producto, para abordar el evento de escasez de materias primas. Aunque esto aumentaría la complejidad del modelo, haría que el sistema sea más adaptable y realista, reflejando mejor las condiciones dinámicas de los entornos de producción.

Además, la integración del sistema de reprogramación con bases de datos vinculadas a sistemas de control en tiempo real es una integración prometedora. Este enfoque permitiría desarrollar un sistema inteligente, capaz de realizar reprogramaciones inmediatas en tiempo real.

Estas líneas de investigación futuras no solo mejorarán la precisión y eficiencia del modelo MILP y el sistema de reprogramación, sino que también contribuirán al desarrollo de sistemas de producción más resilientes y adaptativos, capaces de manejar la complejidad y dinamismo en talleres de trabajo.

Bibliografía

- Bermudez, J. (2022). *tracelink*. Obtenido de <https://www.tracelink.com/resources/resource-center/supply-chain-planning-strategies-reduce-material-shortages>
- Chase, R. B., & Aquilano, N. J. (1994). *Dirección y administración de la producción y las operaciones*. Wilmington, Delaware, Estados Unidos: Adison-Wesley Iberoamericana.
- Company, R., & Corominas, A. (1993). *Organización de la producción I*. Barcelona: Edicions UPC.
- Deskera Academy. (2023). *Deskera*. Obtenido de <https://www.deskera.com/blog/ai-revolution-production-scheduling-manufacturing/>
- Dhingra, J. S., Musser, K. L., & Blankenship, G. L. (1992). Realtime operations scheduling for flexible manufacturing systems. *In Proceedings of the 24th Conference on Winter Simulation*, (págs. 849-855). doi:10.1145/167293.16774
- Dominguez, J., Garcia, S., Ruiz, A., & Álvarez, J. (1995). *Dirección de operaciones. Aspectos tácticos y operativos en la producción y los servicios*. Madrid: McGraw Hill.
- Dutta, A. (1990). Reacting to scheduling exceptions in FMS environments. *IIE transactions*, 22, 300-314. doi:10.1080/07408179008964185
- Ecoembes TheCircularCampus. (2023). *thecircularcampus*. Obtenido de <https://www.ecoembesthecircularcampus.com/escasez-de-materias-primas-causas-consecuencias-y-soluciones/>
- Everett, A., & Ebert, R. (1991). *Administración de la producción y de las operaciones*. México D.F: Prentice Hall.

- Fang, J., & Xi, Y. (1997). A rolling horizon job shop rescheduling strategy in the dynamic environment. *The International Journal of Advanced Manufacturing Technology*, 13, 227–232. doi:10.1007/bf01305874
- Fanjul, L. (2010). Riunet: Repositorio Institucional de la Universitat Politècnica de València. *Nuevos algoritmos para el problema de secuenciación en máquinas paralelas no relacionadas y generalizaciones*. Obtenido de <https://riunet.upv.es/bitstream/handle/10251/9312/tesisUPV3454.pdf>
- García, J. S. (2007). *Organización de la Producción II. Planificación de procesos productivos*. Gipuzkoa-España: Unicopia, C.B. (Impresor).
- Gopalakrishnan, M., Ahire, S. L., & Miller, D. M. (1997). Maximizing the effectiveness of a preventive maintenance system: an adaptive modeling approach. *Management Science*, 827-840.
- Hamzadayi, A., & Yildiz, G. (2016). Event driven strategy based complete rescheduling approaches for dynamic m identical parallel machines scheduling problem with a common server. *Computers & Industrial Engineering*, 91, 66-84. doi:10.1016/j.cie.2015.11.005
- Janak, S. L., Floudas, C. A., Kallrath, J., & Vormbrock, N. (28 de May de 2006). Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling. *Industrial & engineering chemistry research*, 45, 8253-8269. doi:doi:10.1021/ie0600590
- Janak, S., Floudas, C., Kallrath, J., & Vormbrock, N. (2006). Production scheduling of a large-scale industrial batch plant. II. Reactive scheduling. *Industrial & Engineering Chemistry Research*, 8253-8269.
- Jiménez, J., González-Neira, E., & Zambrano-Rey. (2018). An adaptive genetic algorithm for a dynamic . *Management Science Letters*, 1117-1132.
- Karas, A., & Ozcelik, F. (2021). Assembly line worker assignment and rebalancing problem: A mathematical model and an artificial bee colony algorithm. *Computers & Industrial Engineering*, 156. doi:10.1016/j.cie.2021.107195

- Li, Y., Carabelli, S., Fadda, E., Manerba, D., Tadei, R., & Terzo, O. (October de 2019). Integration of machine learning and optimization techniques for flexible job-shop rescheduling in Industry 4.0. *DAUIN-Politecnico di Torino Internal Report*.
- Liu, W., Jin, Y., & Price, M. (2018). New meta-heuristic for dynamic scheduling in permutation flowshop with new order arrival. *The International Journal of Advanced Manufacturing Technology*, 98, 1817-1830. doi:10.1007/s00170-018-2171-y
- Mordor Intelligence. (2024). *Mercado Industria 4.0 - Tamaño, estadísticas y tendencias*. Obtenido de <https://www.mordorintelligence.com/es/industry-reports/industry-4-0-market>
- Olumolade, M. O. (1996). Reactive scheduling system for cellular manufacturing with failure-prone machines. *International Journal of Computer Integrated Manufacturing*, 131-144. doi:10.1080/002075499189682
- Ovacik, I. M., & Uzsoy, R. (1995). Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. *International journal of production research*, 33, 3173-3192. doi:10.1080/00207549508904867
- Pinedo, M. (2016). Design and Implementation of Scheduling Systems. *Springer*, 485-508.
- Pinedo, M. L. (2012). *Scheduling. Theory, Algorithms, and Systems*. New York: Springer.
- Rahman, H., Sarker, R., & Essan, D. (2018). Multiple-order permutation flow shop scheduling under process interruptions. *The International Journal of Advanced Manufacturing Technology*, 2781-2808.
- Reddy, M. S., Ratnam, C., Rajyalakshmi, G., & Manupati, V. K. (2018). An effective hybrid multi objective evolutionary algorithm for solving real time event in flexible job shop scheduling problem. *Measurement*, 78-90. doi:10.1016/j.measurement.2017.09.022
- Shafaei, R., & Brunn, P. (1999). Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and

- stochastic environment. *International Journal of Production Research*, 4105-4117. doi:10.1080/002075499189682
- Shen, X. N., & Yao, X. (2015). Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. *Information Sciences*, 198-224. doi:10.1016/j.ins.2014.11.036
- Silva, N. C., Scarpin, C. T., Ruiz, A., & Pécora, J. E. (2019). Rescheduling production on identical parallel machines upon new jobs arrivals. *IFAC-PapersOnLine*, 52, 2525-2530. doi:10.1016/j.ifacol.2019.11.586
- Stevenson, W. J. (2018). *Operations Management*. Nueva York: McGraw-Hill Education.
- Takeda-Berger, S. L., Agostino, Í. R., da Silva, M. R., & Frazzon, E. M. (2022). Reactive production scheduling approach based on inventory availability. *IFAC-PapersOnLine*, 55, 418-423. doi:doi.org/10.1016/j.ifacol.2022.09.429
- Takeda-Berger, S. L., Agostino, Í. R., Silva, M. R., & Frazzon, E. M. (2022). Reactive production scheduling approach based on inventory availability[Enfoque reactivo de programación de producción basado en la disponibilidad de inventario]. *IFAC-PapersOnLine*, 418-423.
- Tighazoui, A., Sauvey, C., & Sauer, N. (29 de May de 2021). Predictive-reactive strategy for identical parallel machine rescheduling. *Computers & Operations Research*, 134. doi:10.1016/j.cor.2021.105372
- Tighazoui, y., & Sauvey, C. (2021). Minimizing the Total Weighted Waiting Times and Instability. (P. Renna, Ed.) *Applied Sciences*.
- Tliba, K., Diallo, T. M., Peñas, O., Ben Khalifa, R., Ben Yahia, N., & Choley, J.-Y. (2023). Digital twin-driven dynamic scheduling of a hybrid flow shop. *Journal of Intelligent Manufacturing*, 2281-2306.
- Valledor, P., Gómez, A., Priore, P., & Puente, J. (2018). Solving multi-objective rescheduling problems in . *International Journal of Production*, 6363-6377.

- Valledor, P., Gomez, A., Priore, P., & Puente, J. (2020). Modelling and Solving Rescheduling Problems in Dynamic Permutation Flow Shop Environments. *Complexity*, 1-17. doi:10.1155/2020/2862186/bf01305874
- Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 39-62.
- Wang, L., Hu, X., Wang, Y., Xu, S., Ma, S., Yang, K., & Wang, W. (2021). Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. *Computer Networks*, 190. doi:10.1016/j.comnet.2021.107969

Anexos

Anexo 1. Código de iterativo de reprogramación

```
import threading
import time
import os
import pandas as pd
from pyomo.environ import *
import A_S_G_R as gt
from calc_rms import cargar_datos, calcular_pedido_pendiente

code = '3_2_1'
folder = 'data/'

def list_to_file(l, filename):
    with open(filename, 'w') as f:
        for s in l:
            f.write(s + '\n')
        f.close()

def extract_Var(var):
    vname = str(var).split('[')[0]
    vvalue = str(var).split('[')[1].split(']')[0].replace(',', '\t') + '\t' + str(var.value)
    return vname, vvalue

# Función para resolver modelo MILP
def resolver_modelo(inst):
    opt = SolverFactory('gurobi')
    resultado = opt.solve(inst)
    inst.solutions.store_to(resultado)
```

```
# Imprimir resultados

if resultado.solver.status == SolverStatus.ok and
resultado.solver.termination_condition == TerminationCondition.optimal:
    resultado.write(filename=os.path.join(folder, code, 'output', 'Results.json'),
format='json')
    vnameant = ""
    for v in inst.component_data_objects(Var):
        vname, vvalue = extract_Var(v)
        if vname != vnameant:
            if vnameant != "":
                list_to_file(table, os.path.join(folder, code, 'output', vnameant + '.txt'))
            table = []
            vnameant = vname
            table.append(vvalue)
        if len(table) > 0:
            list_to_file(table, os.path.join(folder, code, 'output', vnameant + '.txt'))
    print("Resultados:")
    print(resultado)
    print('Total cost: ', inst.obj.expr())
else:
    print('Solver status is not OK or termination condition is not OPTIMAL')

# Variable global para controlar el evento de reprogramación
evento_reprogramacion = threading.Event()

# Función para cargar el estado de las máquinas
def cargar_valor_de_estado(folder, code):
    ruta_Dpm = os.path.join(folder, code, 'input', 'Status_Machine.tab')
    dm = pd.read_csv(ruta_Dpm, sep='\t')
    return dm
```

Función para cargar el estado de las materias primas

```
def cargar_estado_materia_prima(folder, code):  
    ruta_mpf = os.path.join(folder, code, 'input', 'MPF.tab')  
    df = pd.read_csv(ruta_mpf, sep='\t')  
    return df
```

Función para detectar eventos (por ejemplo, averías de máquinas o Escasez de materia prima)

```
def detectar_eventos(folder, code):  
    estado_anterior_dpm = cargar_valor_de_estado(folder, code)  
    estado_anterior_materia_prima = cargar_estado_materia_prima(folder, code)  
    while True:  
        estado_actual_dpm = cargar_valor_de_estado(folder, code)  
        estado_actual_materia_prima = cargar_estado_materia_prima(folder, code)  
  
        if (estado_actual_dpm != estado_anterior_dpm).any().any():  
            if (estado_actual_dpm['Sm'] != 0).any():  
                print("Avería detectada, reprogramando...")  
                evento_reprogramacion.set()  
            else:  
                print("Máquina en funcionamiento normal.")  
            # Actualizar el estado anterior  
            estado_anterior_dpm = estado_actual_dpm.copy()  
  
        if (estado_actual_materia_prima != estado_anterior_materia_prima).any().any():  
            if (estado_actual_materia_prima['IdPeriod'] != 0).any():  
                print("Se detectó un cambio en el estado de materia prima.  
Reprogramando...")  
                evento_reprogramacion.set()
```

```
# Actualizar el estado anterior
estado_anterior_materia_prima = estado_actual_materia_prima.copy()

# Tiempo de espera antes de la próxima detección
time.sleep(1)

# Función para crear una instancia del modelo y cargar de parámetros de entrada
def crear_instancia():
    dp = DataPortal()
    dp.load(filename=os.path.join(folder, code, 'input', 'Products.tab'), set=gt.m.I)
    dp.load(filename=os.path.join(folder, code, 'input', 'Machines.tab'), set=gt.m.J)
    dp.load(filename=os.path.join(folder, code, 'input', 'Periods.tab'), set=gt.m.T)
    dp.load(filename=os.path.join(folder, code, 'input', 'Machine_Data.tab'),
    param=(gt.m.ca, gt.m.cc), index=gt.m.J)
    dp.load(filename=os.path.join(folder, code, 'input',
    'Initial_Inventory_RawMaterial.tab'), param=gt.m.InvI, index=gt.m.R)
    dp.load(filename=os.path.join(folder, code, 'input', 'Initial_Inventory_Products.tab'),
    param=gt.m.InvIp, index=gt.m.I)
    dp.load(filename=os.path.join(folder, code, 'input', 'HoldingCost_Product.tab'),
    param=gt.m.chp, index=gt.m.I)
    dp.load(filename=os.path.join(folder, code, 'input', 'deferring_Cost.tab'),
    param=gt.m.cd, index=gt.m.I)
    dp.load(filename=os.path.join(folder, code, 'input', 'Demand.tab'), param=gt.m.d,
    index=(gt.m.I, gt.m.T))
    dp.load(filename=os.path.join(folder, code, 'input', 'Production_Requirements.tab'),
    param=(gt.m.req, gt.m.cp, gt.m.st), index=(gt.m.I, gt.m.J))
    dp.load(filename=os.path.join(folder, code, 'input', 'BOM.tab'), param=gt.m.ur,
    index=(gt.m.R, gt.m.I))
    dp.load(filename=os.path.join(folder, code, 'input', 'Status_Machine.tab'),
    param=gt.m.sm, index=(gt.m.J, gt.m.T))
```

```
dp.load(filename=os.path.join(folder, code, 'input', 'InProcess_Products.tab'),
param=gt.m.P, index=(gt.m.I, gt.m.J, gt.m.T))
dp.load(filename=os.path.join(folder, code, 'input', 'Produced.tab'), param=gt.m.pb,
index=(gt.m.I, gt.m.J, gt.m.T))
inst = gt.m.create_instance(dp, name='A_S_G_R')
return inst
```

Función para cargar el plan de producción inicial desde un archivo

```
def cargar_valores_QP(folder, code):
    valores_QP = {}
    filename = os.path.join(folder, code, 'output', 'QP.txt')
    with open(filename, 'r') as f:
        for line in f:
            i, j, t, valor = line.strip().split('\t')
            valores_QP[(int(i), int(j), int(t))] = float(valor)
    return valores_QP
```

Función principal para la lógica de programación y reprogramación

```
def ejecutar_programa():
    # Ruta de archivo del estado de planificación
    status_file_path = os.path.join(folder, code, 'output', 'status.tab')
    # Verificar si el archivo status.tab existe
    if not os.path.exists(status_file_path):
        # Si no existe, crearlo con el valor 0 (no programado)
        with open(status_file_path, 'w') as f:
            f.write('0')
    # Leer el valor del archivo status.tab
    with open(status_file_path, 'r') as f:
        status = f.read().strip()
```

```
if status == '0': # Verificar la existencia de una programación inicial
    # Crear instancia inicial y realizar la programación inicial
    inst = crear_instancia()
    print("Realizando la programación inicial...")
    resolver_modelo(inst)

    # Actualizar el archivo status.tab a 1 (programación inicial existente)
    with open(status_file_path, 'w') as f:
        f.write('1')
else:
    print("Inspeccionando estado del taller...")

# Iniciar el hilo para la detección de eventos
hilo_eventos = threading.Thread(target=detectar_eventos, args=(folder, code))
hilo_eventos.daemon = True # Permite que el hilo se cierre al terminar el programa
principal
hilo_eventos.start()

while True: #Ciclo de reprogramación
    evento_reprogramacion.wait() # Espera a que se detecte un evento
    evento_reprogramacion.clear() # Resetea el evento para la próxima detección
    inst = crear_instancia()

    # Cargar los últimos valores calculados del plan de producción
    valores_QP_actual = cargar_valores_QP(folder, code)

    # Lógica para evento materia prima faltante
    materia_prima_faltante, bom = cargar_datos()
    pedido_pendiente = calcular_pedido_pendiente(materia_prima_faltante, bom)
    # Fijar pedidos pendientes a entregar (Demanda diferida)
    for (i, t), valor in pedido_pendiente.items():
```

```
inst.BO[i, t].fix(valor)

# Lógica ante evento máquina averiada
# Fijar plan de producción de un producto
for (i, j, t), valor in valores_QP_actual.items():
    if inst.sm[j, t] == 0: # Máquina no averiada
        if inst.P[i, j, t] != 0: # Producción de un producto ha comenzado
            inst.QP[i, j, t].fix(valor)
            print(f"Variable QP[{i}, {j}, {t}] fijada en {valor}")

# Realizar la reprogramación (resolver el modelo nuevamente)
resolver_modelo(inst)
print("Inspeccionando estado del taller...")

# Ejecutar el programa
if __name__ == "__main__":
    ejecutar_programa()
```


Anexo 2. Código de cálculo de escasez de materias primas

```
import os
import pandas as pd

code = '3_2_1'
folder = 'data/'
def cargar_datos():
    ruta_materia_prima = os.path.join(folder, code, 'input', 'MPF.tab')
    ruta_BOM = os.path.join(folder, code, 'input', 'BOM.tab')

    # Cargar datos de materia prima faltante
    df_materia_prima_faltante = pd.read_csv(ruta_materia_prima, sep='\t')
    # Cargar datos de lista de materia prima desde TAB
    df_BOM = pd.read_csv(ruta_BOM, sep='\t')

    return df_materia_prima_faltante, df_BOM

# Función para calcular la cantidad de pedido pendiente
def calcular_pedido_pendiente(df_materia_prima_faltante, df_BOM):
    pedido_pendiente = {}

    for _, row in df_materia_prima_faltante.iterrows():
        if row['IdPeriod'] != 0:
            mp_faltante = row['IdRawMaterial']
            cantidad_faltante = row['Quantity']

            # Encontrar producto relacionado en el archivo BOM
            BOM_relacion = df_BOM[df_BOM['IdRawMaterial'] == mp_faltante]

            for _, bom_row in BOM_relacion.iterrows():
                i = bom_row['IdProduct']
                cantidad_necesaria = bom_row['UtilRate']

            # Evitar la división por cero
            if cantidad_necesaria != 0:
                unidades_no_fabricadas = cantidad_faltante / cantidad_necesaria

            # Determinar el periodo de pedido pendiente(demanda diferida)
            periodo_llegada = row['IdPeriod']
            t = periodo_llegada - 1

            # Almacenar el resultado en el diccionario
```

```
        if (i, t) in pedido_pendiente:
            pedido_pendiente[(i, t)] += unidades_no_fabricadas
        else:
            pedido_pendiente[(i, t)] = unidades_no_fabricadas

    return pedido_pendiente

# Ejecutar el programa
if __name__ == "__main__":

    MPF, BOM = cargar_datos()
    # Calcular la pedidos pendientes a entregar(demanda diferida)
    pedido_pendiente = calcular_pedido_pendiente(MPF, BOM)

    # Mostrar resultados
    print("Variables de Pedido pendiente:")
    for (i, t), valor in pedido_pendiente.items():
        print(f"BO[{i}, {t}] = {valor:.1f} unidades")
```

