



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL



Planificación diaria para la asignación de cargas de trabajo en plantas de madera en CMPC Maderas spa.

Por: Sofía Paulina Rodríguez Retamales

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al título profesional de Ingeniera Civil Industrial

Profesor Guía

Carlos Emilio Contreras Bolton

Enero 2023

Concepción (Chile)

© 2023 Sofía Paulina Rodríguez Retamales

© 2023 Sofía Paulina Rodríguez Retamales

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Agradecimientos

En primer lugar, quiero agradecer a mis padres por haberme dado la oportunidad de llevar a cabo una carrera universitaria, además de su apoyo en los momentos difíciles y su compañía en los momentos de felicidad.

Agradecer también a mi hermano Pablo que es mi mayor compañía, especialmente en esta etapa, la persona que siempre está para mí en todo momento y un pilar fundamental en mi vida.

Por otro lado, a mis amigos y amigas, que sin duda han sido un apoyo durante estos largos años, con quienes comparto penas y alegrías.

Finalmente, agradecer tanto a los profesores como a los ayudantes que demuestran interés en nuestro aprendizaje, también al profesor Carlos y a Sergio Cuevas de CMPC que me apoyaron durante esta gran etapa.

Resumen

La presente memoria de título tiene como objetivo abordar el problema de planificación diaria de cargas de trabajo en la planta de remanufactura Los Ángeles de CMPC Maderas. El problema es modelado como un problema de programación de taller de trabajo flexible (Flexible Job Shop Scheduling Problem, FJSSP), que consiste en asignar y secuenciar un mínimo dado de trabajos en diferentes centros de procesamiento, y cada centro de procesamiento contiene diferentes máquinas o recursos. Las restricciones adicionales del FJSSP consisten en distinguir los diferentes subconjuntos de máquinas para las respectivas operaciones de cada pedido y calcular el tiempo de procesamiento de cada operación dependiendo de la o las máquinas que le sean asignadas. Esta dificultad se debe a que las operaciones tienen subconjuntos de máquinas que pueden trabajar en paralelo o no. El problema modelado es implementado mediante programación lineal entera mixta y programación por restricciones. El modelo de programación lineal entera mixta es resuelto mediante el solucionador IBM ILOG CPLEX Optimization Studio. Mientras el modelo de programación por restricciones es resuelto usando el solucionador CP-SAT OR-tools de Google. Se presentan cinco instancias para el análisis, abarcando pedidos con muy baja demanda hasta pedidos con sobrecarga. Los resultados presentan soluciones óptimas para las instancias con muy baja demanda y soluciones factibles para las instancias con mayor cantidad de pedidos. Los modelos difieren en cuanto a tiempos de ejecución, valor objetivo encontrado y error relativo, siendo el modelo por restricciones el que entrega mejores resultados.

Abstract

This thesis aims to address the daily workload scheduling problem at the Los Angeles remanufacturing plant of CMPC Maderas, which is approached as a Flexible Job Shop Scheduling Problem (FJSSP) which consists in assigning and sequencing a given minimum number of jobs in different processing centres, and each processing centre contains different machines or resources. The additional constraints of the FJSSP are to distinguish the different subsets of machines for the respective operations of each order and to calculate the processing time of each operation depending on the machine(s) assigned to it. This difficulty is due to the fact that the operations have subsets of machines that may or may not work in parallel. The modelled problem is implemented using mixed integer linear programming and constraint programming. The mixed integer linear programming model is solved using the IBM ILOG CPLEX Optimization Studio solver. While the constraint programming model is solved using Google's CP-SAT OR-tools solver. Five instances are presented for analysis, covering orders with very low demand up to orders with overload. The results present optimal solutions for the instances with very low demand and feasible solutions for the instances with more orders. The models differ in terms of execution times, objective value found and relative error, with the constraint model delivering the best results.

Índice de Contenidos

Agradecimientos	3
Resumen	4
Abstract	5
Índice de Contenidos	6
Índice de Tablas	8
Índice de Figuras	9
Capítulo 1. Introducción	10
1.1 Antecedentes generales	10
1.2 Objetivo general	12
1.3 Objetivos específicos	12
1.4 Estructura del informe	12
Capítulo 2. Problema de asignación de tareas	14
2.1 Planificación diaria	14
2.2 Programación de taller de trabajo flexible	15
2.3 Revisión de la literatura	19
Capítulo 3. Modelos propuestos	23
3.1 Introducción	23
3.2 Modelo de programación lineal entera mixta	24
3.3 Modelo de programación por restricciones	26
Capítulo 4. Implementación y resultados	30
4.1 Detalles computacionales	30
4.2 Instancias de prueba	30
4.3 Resultados computacionales	31
Capítulo 5. Conclusiones	35
Referencias	37
	6

Índice de Tablas

Tabla 1: Etapas de modelo Capacity.	11
Tabla 2: Instancia de ejemplo.	16
Tabla 3: Máquinas con sus respectivas velocidades.	17
Tabla 4: Producción requerida en cada operación.	18
Tabla 5: Tabla de resultados con los cinco escenarios.	32

Índice de Figuras

Figura 1: Carta Gantt solución.	19
Figura 2: Resultado Carta Gantt modelo CP escenario 1.	34
Figura 3: Resultado Carta Gantt modelo MILP escenario 1.	34
Figura A.4: Resultado Carta Gantt modelo CP escenario 2.	40
Figura A.5: Resultado Carta Gantt modelo MILP escenario 2.	40
Figura A.6: Resultado Carta Gantt modelo CP escenario 3.	40
Figura A.7: Resultado Carta Gantt modelo MILP escenario 3.	41
Figura A.8: Resultado Carta Gantt modelo CP escenario 4.	41
Figura A.9: Resultado Carta Gantt modelo MILP escenario 4.	41
Figura A.10: Resultado Carta Gantt modelo CP escenario 5.	42
Figura A.11: Resultado Carta Gantt modelo MILP escenario 5.	42

Capítulo 1. Introducción

En este capítulo se presentan los antecedentes generales de este trabajo, sus objetivos y, por último, como se organiza la memoria de título.

1.1 Antecedentes generales

El éxito de un sistema depende de una distribución efectiva de las tareas entre los usuarios y el mismo sistema. De modo que una planificación operativa establece la programación detallada de las actividades que se ejecutan en el corto plazo, con el fin de cumplir con los requerimientos de los clientes y técnicas de la manera más competente posible. Esto permite que se utilicen los activos de manera eficiente y se genere una mayor producción en tiempos acotados, entregando un mejor servicio al cliente y a su vez marcando una ventaja competitiva dentro del mercado.

No obstante, y para cualquier nivel de decisión, el uso de modelos de planificación ha facilitado los procesos de toma de decisiones, mejorando la gestión tanto productiva como económica de las empresas forestales. En este sentido, la técnica más usada como herramienta para la planificación forestal ha sido la programación matemática (Prieto & Díaz-Balteiro, 1999). Esta técnica ha dado buenos resultados en la mayoría de las empresas forestales chilenas, generando principalmente una disminución de los costos involucrados en las operaciones forestales. Su éxito se debe a la adaptabilidad de los modelos de optimización a un amplio rango de problemas, que varían con el tipo de empresa (estatal o privada), con el entorno ecológico (plantaciones o bosque nativo) y con el enfoque de interés (estratégico, táctico u operativo) (Anderson et al., 2003).

Una de las empresas forestal más grande de Chile, es CMPC, que produce y comercializa productos forestales, celulosa, papeles y productos tissue, provenientes de plantaciones certificadas con manejo forestal responsable. A su vez, CMPC Maderas es una unidad de negocios de la empresa que tiene como misión agregar valor al patrimonio forestal, a través de la producción y comercialización de productos de madera (CMPC, 2022).

En la actualidad, CMPC Maderas está trabajando en un modelo de simulación y optimización de procesos llamado Capacity, que busca generar la planificación de todas las plantas de aserradero y remanufactura de CMPC. Capacity tiene como objetivo principal maximizar los

resultados del negocio integrado. El alcance del proyecto considera cuatro plantas de aserraderos: Mulchén, Nacimiento, Bucalemu y Loncoche, y dos plantas de remanufactura: Los Ángeles y Coronel. Además, este proyecto contempla tres segmentos para la planificación: estratégica (anual), táctica (mensual-semanal) y operativa (hora-hora) y a su vez se puede subdividir en cuatro etapas que se reflejan en la **Tabla 1**.

Tabla 1: Etapas de modelo Capacity.

#	Etapas de Capacity	Alcance planificación
1	Esquemizador de cortes	Estratégica, táctica y operativa
2	Balance procesos	Estratégica, táctica y operativa
3	Asignación	Estratégica, táctica y operativa
4	Planificación diaria	Operativa

Fuente: Elaboración propia.

La primera etapa consiste en generar patrones de corte de trozos en función de las distintas características de los trozos y, de las características de las máquinas que se encuentran disponibles en las distintas plantas, de esta forma se puede lograr un mejor aprovechamiento de la materia prima disponible. En la etapa de balance de procesos, se evalúa cada proceso de manera independiente para determinar la capacidad de estos, con el fin de integrar en función del diseño de cada planta. Luego, la tercera etapa, tiene como objetivo asignar la producción de los pedidos a las diferentes plantas. Finalmente, ya con una demanda de producción en cada planta se puede pasar a la etapa de planificación diaria, siendo esta última etapa del modelo Capacity, el problema a abordar en esta memoria de título. Esta etapa de planificación diaria tiene como meta, la creación de un modelo que genere una bajada de un plan semanal mediante una carta Gantt para cada una de las plantas.

Las plantas consideran diferentes operaciones que pueden ser procesadas en diversas máquinas, por lo que se puede abordar como un problema de programación de taller de trabajo flexible (Flexible Job Shop Scheduling Problem, FJSSP) (Pinedo, 2016). El FJSSP consiste en asignar y secuenciar un mínimo dado de trabajos y de centros de procesamiento, y cada centro de procesamiento puede tener un número diferente de máquinas o recursos (Chen et al., 2012). El FJSSP pertenece a la clase de problemas denominados NP-hard (Jansen et al., 2005), para los cuales aún no se dispone de algoritmos polinomiales que

encuentren la solución óptima. Se encuentran disponibles distintos algoritmos, heurísticas y metaheurísticas para la resolución de este tipo de problemas. Sin embargo, la gran mayoría de estos resuelven modelos propuestos con supuestos que se alejan de la realidad y excluyen variables importantes a considerar. Además, tienen un gran nivel de complejidad para ser aplicados en la industria, debido al poco detalle que se muestra en los trabajos disponibles. Para la correcta realización de la programación se deben establecer las restricciones, políticas y consideraciones que se deben cumplir. Por tanto, en este trabajo se propone modelar y resolver la problemática de la empresa a partir del FJSSP. La problemática es modelada con programación por restricciones (Constraint Programming, CP) y programación entera mixta (Mixed Integer Linear Programming, MILP). Posteriormente, los modelos son implementados en solvers y testeados en cinco instancias basadas en la planta de remanufactura Los Ángeles de CMPC Maderas.

1.2 Objetivo general

Modelar la planificación diaria de cargas de trabajo en la planta de remanufactura Los Ángeles de CMPC Maderas.

1.3 Objetivos específicos

- Desarrollar una revisión de la literatura para la planificación de cargas de trabajo en plantas forestales o similares.
- Diseñar un modelo para la planificación diaria de cargas de trabajo en la planta de remanufactura Los Ángeles de CMPC Maderas.
- Implementar un modelo de programación lineal entera mixta y de programación por restricciones.
- Comparar y analizar los resultados obtenidos por los modelos implementados.

1.4 Estructura del informe

El resto del documento se organiza de la siguiente manera. En el Capítulo 2, se presenta el problema formalmente, detallando sus componentes y restricciones, junto con una descripción detallada del problema y, una revisión de la literatura. Luego, en el Capítulo 3, se presentan dos modelos matemáticos mediante MILP y CP. En el Capítulo 4, se detallan las instancias utilizadas y como se implementaron los distintos modelos. Además, se

presentan los resultados obtenidos junto con su respectivo análisis. Finalmente, se presentan las conclusiones obtenidas en el Capítulo 5.

Capítulo 2. Problema de asignación de tareas

En este capítulo se muestra el contexto del problema dentro de la empresa y el modelo de programación asociado. Además, se revisa la literatura más importante relacionada al problema.

2.1 Planificación diaria

Como se menciona anteriormente, el objetivo de la etapa de planificación diaria es la creación de un modelo que genere una bajada de un plan semanal mediante una carta Gantt para cada una de las plantas. Además, este modelo debe contener la planificación hora a hora según las órdenes de fabricación, con un ordenamiento lógico de las producciones considerando la manera más eficiente de producir, reduciendo tiempos ociosos entre procesos y minimizando los tiempos de configuración, logrando el producto dentro del plazo.

Para la elaboración del modelo se consideran los siguientes supuestos:

- a. Todas las máquinas están disponibles al inicio del programa.
- b. Todos los trabajos están listos para su procesamiento en el momento cero.
- c. En el flujo de procesamiento de los trabajos no se presenta recirculación, es decir, los trabajos pasan de una estación a la siguiente apenas vayan concluyendo sus respectivas operaciones.
- d. El orden de las operaciones para cada trabajo está predefinido y no se puede modificar. Además, no se puede iniciar una operación sin haber pasado por la operación anterior. Dependiendo si la operación tiene precedencia o no, puede comenzar la operación siguiente una vez iniciada la operación actual o en caso contrario, debe esperar a que la operación actual termine para recién iniciar la siguiente. A modo de simplicidad, en caso de que no haya precedencia, la operación se inicia 30 segundos después de iniciar su operación anterior. Con el fin que no exista conflicto en caso de que la velocidad de la máquina siguiente sea superior a la actual.
- e. No se puede iniciar una operación hasta que existan máquinas disponibles para realizar dicha operación, es decir, cada máquina puede procesar solo una operación a la vez.

- f. Se conocen los tiempos de las máquinas, incluyendo tiempos de holgura y de mantenciones.
- g. Por simplicidad, los tiempos de configuración son un promedio y se incluyen en los tiempos de holgura.
- h. En caso que se requiera un traspaso entre una planta y otra, el tiempo de transporte es conocido. Si la planta recibe un pedido a mitad de semana, se acota la planificación del pedido a partir del día en que llegue a la planta y, en el caso que tenga que trasladarse a otra planta en algún día en específico, se obliga a terminar dentro del plazo establecido.
- i. No existen paradas de emergencia ni necesidades de mantenimiento a lo largo del proceso ya que vienen previamente considerados.
- j. Las máquinas no son homogéneas, es decir, el tiempo de procesamiento del trabajo puede ser menor dependiendo de la máquina seleccionada, para un producto en particular, esto depende de la velocidad de cada máquina.
- k. Existen tres distintos escenarios para la producción con diferentes conjuntos de máquinas:
 - i. Operaciones que contienen sólo una máquina en la que se puede planificar una tarea.
 - ii. Operaciones que contienen más de una máquina y se pueden utilizar en paralelo para la misma tarea.
 - iii. Operaciones en las que hay más de una máquina, pero la tarea se debe realizar en sólo una de ellas.

2.2 Programación de taller de trabajo flexible

Como se menciona anteriormente, se decide abordar el problema como un FJSSP, que es una generalización del problema clásico de taller de trabajo (Job Shop Problem, JSP) y posibilita la solución de dos subproblemas; la asignación de la máquina adecuada a la operación y, determina la secuencia en que se realizan las operaciones dentro de la máquina. Así, el FJSSP permite optimizar el uso de los recursos con mayor flexibilidad, ya que se puede iterar dentro de un conjunto de máquinas disponibles para cada operación, aumentando de esta forma la complejidad de la programación (Jin et al., 2012).

El problema abordado, se puede definir formalmente, dado m máquinas y n pedidos. Cada pedido consiste en una secuencia de operaciones $O_{i,j}$ que indican la j -ésima operación del pedido i , asimismo, J_i es el número de operaciones requeridas para el pedido i . Se denomina A al conjunto de máquinas, donde $A = \{M_{1,1}, M_{1,2}, \dots, M_{i,j}\}$. Para la ejecución de cada operación $O_{i,j}$ se requiere una máquina de un subconjunto de máquinas dado llamado $M_{i,j} \in A$ y un tiempo de proceso $Ts_{i,j}$ que depende de la velocidad de la máquina en la cual se realiza. W_k es el número de operaciones asignadas a la máquina k . El índice k denota las máquinas, el índice i los pedidos, el índice j denota la operación y alternativa de máquina. Finalmente, se asigna un índice w para determinar la secuencia de las operaciones asignadas a cada máquina. Este problema tiene el fin de encontrar el programa de menor tiempo de duración que cumpla con todas las restricciones.

En la **Tabla 2**, se presenta una instancia pequeña del problema con dos pedidos ($n = 2$), donde cada pedido contiene cinco operaciones, $J_1 = J_2 = 5$.

Tabla 2: Instancia de ejemplo.

Pedidos	Operaciones	Máquinas	Paralelo	Precedencia
1	$O_{1,0}$	1	0	0
	$O_{1,1}$	2	0	1
	$O_{1,2}$	3	1	1
		4	1	1
	$O_{1,3}$	5	0	1
	$O_{1,4}$	6	0	1
		7	0	1
		8	0	1
2	$O_{2,0}$	1	0	0
	$O_{2,1}$	2	0	1
	$O_{2,2}$	3	1	1
		4	1	1
	$O_{2,3}$	5	0	1
	$O_{2,4}$	6	0	1
		7	0	1
		8	0	1

Fuente: Elaboración propia.

En la **Tabla 2** se puede ver que las operaciones $O_{1,2}$ y $O_{2,2}$ poseen un subconjunto de máquinas $M_{1,2} = M_{2,2} = \{3,4\}$ y están clasificadas como máquinas en paralelo, esto quiere decir que ambas pueden ser seleccionadas para realizar la misma operación repartiéndose entre ellas la carga de trabajo. Mientras, $O_{1,4}$ y $O_{2,4}$ contienen un subconjunto de máquinas $M_{1,4} = M_{2,4} = \{6,7,8\}$ y no están clasificadas en paralelo, es decir, de este último subconjunto mencionado, se debe escoger solo una máquina para realizar dicha operación. Notar que solo la primera operación de cada pedido no posee precedencia, esto quiere decir que la operación siguiente puede iniciar una vez iniciada la operación actual.

La **Tabla 3** contiene las velocidades del conjunto de máquinas A , la cual se mide en *piezas/seg* y la **Tabla 4**, que muestra la producción total que se debe realizar en cada operación, donde la unidad de medida está en *piezas*. Cabe destacar, que se realiza una previa transformación de los datos, amplificando por 1000 la producción total y la velocidad, esto con el fin de minimizar la pérdida de datos al momento de implementar los modelos, esto se debe a que el solver utilizado para resolver el CP solo trabaja con datos enteros y no reales.

Tabla 3: Máquinas con sus respectivas velocidades.

Máquina	Nombre	Velocidad
1	Waco 1	5633
2	Trozadora 5/4	4683
3	Conception	3616
4	Western	3616
5	H23	1916
6	Leadermak	2160
7	Makor 1	1566
8	Efisa	1850

Fuente: Elaboración propia.

Tabla 4: Producción requerida en cada operación.

Proceso	Producción total
$O_{1,0}$	17214000
$O_{1,1}$	13427000
$O_{1,2}$	12890000
$O_{1,3}$	33333000
$O_{1,4}$	32000000
$O_{2,0}$	7599000
$O_{2,1}$	5928000
$O_{2,2}$	5690000
$O_{2,3}$	2850000
$O_{2,4}$	2850000

Fuente: Elaboración propia.

En la **Figura 1** se presenta una solución factible para la instancia de ejemplo (**Tabla 1**), se puede observar el programa de los pedidos en las diferentes máquinas. Además, ambos pedidos siguen una secuencia de operaciones como se menciona en el supuesto (d) y se ve reflejada la restricción de precedencia desde la operación número dos en adelante. En cuanto a las máquinas en paralelo, se puede ver que el pedido 1, divide su producción en la operación $O_{1,2}$ entre las máquinas Western y Conception. En cambio, el pedido 2 concentra su producción en solo una máquina. Finalmente, para el caso de las operaciones $O_{1,4}$ y $O_{2,4}$, que deben elegir solo una máquina, la producción se realiza en la máquina Leadermak y Efisa, respectivamente. El problema abordado, a diferencia del FJSSP que trabaja con un conjunto de máquinas que pueden ser utilizadas para todas las operaciones del problema, debe distinguir entre los diferentes subconjuntos de máquinas para las respectivas operaciones. Además, el tiempo de procesamiento de las operaciones en cada máquina no viene definido de antemano, este debe ser calculado mediante el modelo. Así, depende de la máquina a la que se asigne, como se distribuya su producción en caso de ser máquinas en paralelo y, de la velocidad de la máquina escogida para procesar la operación.

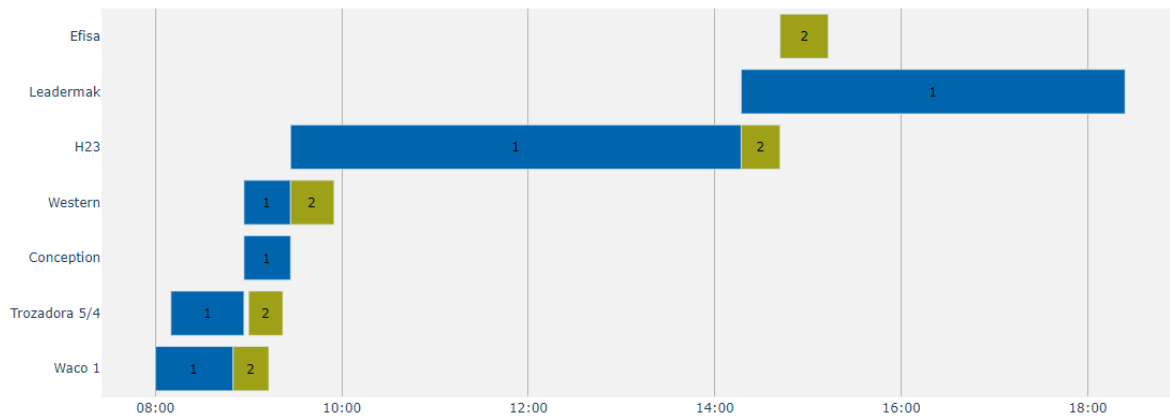


Figura 1: Carta Gantt solución.

Fuente: Elaboración propia.

2.3 Revisión de la literatura

La resolución de problemas de asignación de cargas de trabajo ha sido ampliamente estudiada y se encuentran disponibles distintos algoritmos, heurísticas y metaheurísticas. En esta revisión se destacan los trabajos con aplicación en plantas forestales y similares.

En Goyeneche (2022) se implementó un modelo mediante programación lineal entera mixta para secuenciar las operaciones silvícolas, optimizando los tiempos de las cuadrillas de trabajo de un núcleo forestal. El modelo propuesto es resuelto mediante el solver CBC y mediante un algoritmo de ramificación y acotamiento. El algoritmo de ramificación y corte necesita menor tiempo que el solver CBC para el modelo. Sin embargo, los tiempos que toma el solver son razonablemente moderados.

Campo (2018) propuso un modelo de programación de producción en plantas con configuración basada en FJSSP en condiciones de incertidumbre, tiempos difusos de procesamiento, alistamiento dependiente de la secuencia, y ventanas de tiempo en las entregas. Para resolver el problema usan algoritmos genéticos (GA), minimizando la tardanza ponderada y la prontitud ponderada. El modelo fue aplicado en una empresa del sector textil. Al comparar resultados del GA contra dos heurísticas, se comprobó que la metodología propuesta supera en más de 30% a las otras heurísticas.

Ku & Beck (2016) presentaron y evaluaron cuatro formulaciones MILP para el problema de programación de talleres, la formulación indexada en el tiempo (Bowman, 1959), la

formulación basada en el rango (Wagner, 1959), la formulación disyuntiva tradicional (Manne, 1960) y una segunda formulación disyuntiva propuesta por Liao & You (1992). Realizaron un estudio empírico de los cuatro modelos MILP utilizando CPLEX, GUROBI, y SCIP. Los solvers modernos son capaces de demostrar la optimalidad de problemas de tamaño moderado. La formulación indexada en el tiempo es capaz de rendir mejor que la formulación basada en rangos. Además, la formulación disyuntiva original es más eficiente que los otros tres modelos, tanto para CPLEX como para GUROBI. Finalmente, generaron una comparación del MILP con la programación por restricciones, con CPLEX para el modelo MILP disyuntivo con multihilo y ajuste de parámetros, y CPLEX Optimizer para el modelo CP, entregando comportamientos de forma similar para tamaño moderado y el objetivo de demostrar la optimalidad. Sin embargo, CP domina a MILP para las instancias de mayor tamaño.

En Díaz-Cazaña et al. (2014) se implementó una heurística para la resolución de un JSP considerando tareas de mantenimiento preventivo. El método de solución incluye un modelo de programación lineal entera mixta inspirado en el problema del vendedor viajero, donde el tiempo de iniciación es considerado como métrica de distancia. La heurística propuesta consta de cuatro etapas, búsqueda de solución inicial, obtención de la importancia relativa para cada máquina, obtención de una medida de distancia corregida y la obtención de la solución final. Los resultados muestran una reducción de 423,7 horas en el tiempo de inactividad total.

Birgin et al. (2014) implementó un modelo MILP para una versión extendida del FJSSP. Esta versión permite que un trabajo sea un conjunto de operaciones con una relación de precedencia dadas por un grafo acíclico dirigido, de manera arbitraria, en lugar de un orden lineal, el objetivo es la minimización del tiempo de ejecución. El modelo fue resuelto con el solver CPLEX, con 3600 segundos de tiempo límite, con número de subprocesos igual a 1 y 2048MB para la memoria de trabajo. Se utilizaron diferentes conjuntos de instancias y se obtuvieron soluciones óptimas tanto para las instancias de tamaño pequeño como para las de mayor tamaño.

En Rivera (2012) se diseñó un proceso de programación de la producción en cartulinas CMPC S.A., cuya innovación más importante es que incluye una heurística del tipo búsqueda

en anchura para la resolución del JSP, permitiendo generar la carga de trabajos en las máquinas. En los resultados se destaca que, en el quinto mes de operaciones del nuevo proceso, se obtuvo una producción de 4.000 toneladas adicionales y un aumento estimado de ingresos de MM USD\$1.2.

Osorio & Mota (2008) presentan una propuesta jerárquica para resolver el problema de planificación y control de la producción en una configuración productiva del tipo FJSSP. La solución propuesta, entrega resultados cercanos al óptimo. Además, se aplicó en un problema de flexibilidad parcial que pudo ser resuelto.

En Fattahi et al. (2007) presentan un modelo de programación lineal entera mixta y dos enfoques heurísticos para el FJSSP. El FJSSP es aplicado a una empresa de fabricación de piezas y equipos para la industria petroquímica, es abordado en dos tipos de enfoques: enfoques jerárquicos y enfoques integrados. Concluyen que los algoritmos jerárquicos tienen mejor rendimiento que los algoritmos integrados, aun así, todos los experimentos validan la calidad de los algoritmos propuestos.

En Laroze et al. (1998) desarrollan y comparan cuatro modelos de programación lineal entera mixta para resolver el problema de planificación de las faenas silvícolas de una empresa forestal. Para resolver las distintas exigencias planteadas por la empresa se formularon modelos con diferentes funciones objetivo y conjuntos de restricciones. Los resultados indican que los modelos permiten planificar de manera más eficiente las faenas a nivel operativo.

Dentro de la literatura también se puede encontrar las distintas capacidades de resolución de algunos solucionadores. Por ejemplo, en Da Col & Teppan (2019) comparan las capacidades de resolución de CPLEX Optimizer y Google OR-Tools de Google en instancias del JSP con respecto a la calidad de las soluciones cuyo objetivo es minimizar el makespan, y el tiempo de resolución. Los solucionadores disponen de 20 minutos por instancia de problema, en cuanto a la referencia a gran escala, se dan seis horas para completar la búsqueda de tal manera que se simule un escenario industrial en el que la planificación diaria suele hacerse durante la noche. Aunque ambos utilizan variables de intervalo para expresar las operaciones de trabajo, CPLEX CP Optimizer utiliza tipos básicos para codificar los intervalos, mientras que OR-Tools utiliza tres variables para un único intervalo, lo que ralentiza la propagación

de las restricciones. CPLEX CP Optimizer fue capaz de resolver de forma óptima el 66% de las instancias a gran escala, frente al 29% de OR-Tools.

Capítulo 3. Modelos propuestos

En este capítulo se presenta una breve introducción de los modelos propuestos. Posteriormente, se presentan el modelo de programación lineal entera mixta y el modelo de programación por restricciones.

3.1 Introducción

La CP es un paradigma para resolver problemas combinatorios que se basa en una amplia gama de técnicas de inteligencia artificial, informática e investigación operativa. En la CP, los usuarios declaran las restricciones sobre las soluciones factibles para un conjunto de variables de decisión. Las restricciones difieren de las primitivas comunes de los lenguajes de programación imperativos en que no especifican una secuencia de pasos a ejecutar, sino las propiedades de una solución a encontrar. Además de las restricciones, los usuarios deben especificar un método para resolverlas (Rossi, 2014).

La CP involucra tanto el modelamiento como la resolución de problemas de satisfacción de restricciones (Constraint Satisfaction Problem, CSP) típicamente de complejidad NP-hard. Las etapas básicas para la resolución de un CSP son su modelización y su posterior resolución mediante la aplicación de técnicas CSP específicas.

Un CSP puede ser representado mediante una tupla (X, D, C) donde:

- X es un conjunto de n variables $\{x_1, \dots, x_n\}$.
- $D = \langle D_1, \dots, D_n \rangle$ es una tupla de dominios finitos donde se interpretan las variables X .
- $C = \{c_1, c_2, \dots, c_p\}$ es un conjunto finito de restricciones. Cada restricción k -aria c_i está definida sobre un conjunto de k variables $var(c_i) \subseteq X$ que restringe los valores que dichas variables pueden simultáneamente tomar. Particularmente, una restricción es binaria cuando relaciona únicamente a dos variables x_i y x_j , y se suele denotar como c_{ij} . Todas las restricciones definidas en un CSP son conjuntivas, de manera que una solución debe de satisfacer a todas ellas.

La instanciación de un conjunto de variables es una tupla de pares ordenados, donde cada par ordenado (x_i, a_i) asigna el valor $\{a_i \in D_i\}$ a la variable x_i el cual es un valor consistente si

existe al menos una solución del CSP en la cual $x_i = a_i$. Una solución a un CSP es una asignación (a_1, a_2, \dots, a_n) de valores a todas sus variables, de tal manera que se satisfagan todas las restricciones del CSP (Barber & Salido, 2003).

En tanto los problemas MILP son aquellos en los que hay al mismo tiempo variables continuas y variables que sólo pueden tomar valores enteros, por ejemplo, las variables binarias. Implica la optimización de una función objetivo lineal, sujeta a restricciones lineales de igualdad y desigualdad (Floudas & Lin, 2005).

Finalmente, cabe destacar que la diferencia entre ambos modelos, radica en la necesidad de la CP de que todas las variables decisionales deben ser definidas sobre conjuntos discretos.

3.2 Modelo de programación lineal entera mixta

A continuación, se presenta el modelo MILP para la resolución del problema de asignación de cargas de trabajo, con sus respectivos parámetros, variables de decisión y sus restricciones.

Parámetros:

$PT_{i,j}$: Producción total de la operación $O_{i,j}$.

V_k : Velocidad de la máquina k .

$a_{i,j}$: $\begin{cases} 1, & \text{si las máquinas de la operación } O_{i,j} \text{ trabajan en serie.} \\ 0, & \text{en otro caso.} \end{cases}$

$b_{i,j}$: $\begin{cases} 1, & \text{si la operación } O_{i,j} \text{ debe esperar que la operación previa termine para comenzar.} \\ 0, & \text{en otro caso.} \end{cases}$

L : Número muy grande.

I_i : Inicio del pedido i .

F_i : Fin del pedido i .

Variables de decisión:

C_{max} : Tiempo máximo en el que se completan los trabajos.

$Ts_{i,j,k}$: Tiempo de proceso de la operación $O_{i,j}$ en la máquina k .

$To_{i,j,k}$: Tiempo de inicio de la operación $O_{i,j}$ en la máquina k .

$P_{i,j,k}$: Producción de la operación $O_{i,j}$ en la máquina k .

$Y_{k,j,j'}$: $\begin{cases} 1, & \text{si la operación } O_{i,j} \text{ precede a la operación } O_{i',j'} \text{ en la máquina } k. \\ 0, & \text{en otro caso.} \end{cases}$

A continuación, se presenta el modelo MILP para el problema de asignación de cargas de trabajos.

$$\text{Min } C_{max} \quad (1)$$

Sujeto a:

$$C_{max} \geq To_{i,j,k} + Ts_{i,j,k} \quad \forall i = 1, \dots, n \quad (2)$$

$$To_{i,j,k} \geq I_i \quad \forall i = 1, \dots, n \quad (3)$$

$$To_{i,j,k} + Ts_{i,j,k} < F_i \quad \forall i = 1, \dots, n \quad (4)$$

$$To_{i,j,k} + (Ts_{i,j,k} * b_{i,j}) < To_{i,j+1,k} \quad \forall j = 1, \dots, j_i, \forall j = 1, \dots, j_{i-1} \quad (5)$$

$$Ts_{i,j,k} = \frac{P_{i,j,k}}{V_k} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m \quad (6)$$

$$\sum_k P_{i,j,k} = PT_{i,j} \quad \forall i = 1, \dots, n, \forall j: a_{i,j} = 0 \quad (7)$$

$$\sum_k P_{i,j,k} * Y_{k,i,j} = PT_{i,j} \quad \forall i = 1, \dots, n, \forall j: a_{i,j} = 1 \quad (8)$$

$$To_{i,j,k} \geq To_{i',j',k} + Ts_{i',j',k} - L * Y_{k,j,j'} \quad \forall j, j' \in j = 1, \dots, j_i : j < j', \forall k = 1, \dots, m \quad (9)$$

$$To_{i',j',k} \geq To_{i,j,k} + Ts_{i,j,k} - L * (1 - Y_{k,j,j'}) \quad \forall j, j' \in j = 1, \dots, j_i : j < j', \forall k = 1, \dots, m \quad (10)$$

$$Y_{k,j,j'} \in \{0,1\} \quad \forall j, j' \in j = 1, \dots, j_i, \forall k = 1, \dots, m \quad (11)$$

$$To_{i,j,k} \geq 0 \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m \quad (12)$$

$$Ts_{i,j,k} \geq 0 \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m \quad (13)$$

$$C_{max} \geq 0 \quad (14)$$

La ecuación (1) representa la función objetivo del modelo que consiste en minimizar el makespan, considerado como el intervalo de tiempo en el que se procesa completamente la totalidad de los trabajos. Las restricciones (2) obligan al makespan a tomar un valor mayor o igual al tiempo de término de cada tarea. Las restricciones (3) y (4) son definidas para el caso de los traspasos entre las diferentes plantas, las restricciones (3) limitan a cada pedido a comenzar después de su tiempo de inicio preestablecido y, las restricciones (4) limitan los pedidos a terminar antes de su tiempo de fin preestablecido. Las restricciones (5) obligan a cada trabajo a seguir una secuencia de operaciones especificada. Las restricciones (6) indican que el tiempo de procesamiento de la operación $O_{i,j}$ en dicha máquina sea igual a la producción dividido en la velocidad de la máquina. Las restricciones (7) y (8) limitan la capacidad para satisfacer la demanda de producción de una operación $O_{i,j}$, así, las restricciones (7) son para las máquinas que trabajan en paralelo y las restricciones (8) son para solo una máquina del subconjunto de máquinas capaces para procesar dicha operación. Las restricciones disyuntivas (9) y (10) aseguran que solo se procesa un trabajo en un momento dado en una máquina dada. La constante L debe ser lo suficientemente grande para garantizar la corrección de estas restricciones. Las restricciones (11) definen las variables como binarias. Las restricciones (12), (13) y (14) garantizan la no negatividad de las variables.

3.3 Modelo de programación por restricciones

A continuación, se presenta el modelo de CP para resolver el problema de asignación de cargas de trabajos, con sus respectivos parámetros, variables y restricciones.

Parámetros:

$PT_{i,j}$: Producción total de la operación $O_{i,j}$.

V_k : Velocidad de la máquina k .

L : Número muy grande.

$a_{i,j}$: $\begin{cases} 1, & \text{si las máquinas de la operación } O_{i,j} \text{ trabajan en serie.} \\ 0, & \text{en otro caso.} \end{cases}$

$b_{i,j}$: $\begin{cases} 1, & \text{si la operación } O_{i,j} \text{ debe esperar que la operación previa termine para comenzar.} \\ 0, & \text{en otro caso.} \end{cases}$

I_i : Inicio del pedido i .

F_i : Fin del pedido i .

Variables de decisión:

C_{max} : Tiempo máximo en el que se completan los trabajos.

$To_{i,j}$: Tiempo de inicio de cada operación $O_{i,j}$.

$Ts_{i,j,k}$: Tiempo de proceso de la operación $O_{i,j}$ en la máquina k .

$Tm_{k,w}$: Tiempo de inicio de la máquina k en su prioridad w .

$P_{i,j,k}$: Producción de la operación $O_{i,j}$ en la máquina k .

$X_{i,j,k,w}$: $\begin{cases} 1, & \text{si la operación } O_{i,j} \text{ se procesa en la máquina } k \text{ en la prioridad } w. \\ 0, & \text{en otro caso.} \end{cases}$

$Y_{k,i,j}$: $\begin{cases} 1, & \text{si la máquina } k \text{ se selecciona para la operación } O_{i,j}. \\ 0, & \text{en otro caso.} \end{cases}$

A continuación, se presenta el modelo CP para el problema de asignación de cargas de trabajos.

$$\text{Min } C_{max} \quad (15)$$

Sujeto a:

$$C_{max} \geq To_{i,j} + Ts_{i,j,k} \quad \forall i = 1, \dots, n \quad (16)$$

$$To_{i,j} \geq I_i \quad \forall i = 1, \dots, n \quad (17)$$

$$To_{i,j} + Ts_{i,j,k} < F_i \quad \forall i = 1, \dots, n \quad (18)$$

$$Tm_{k,w} + Ts_{i,j,k} * X_{i,j,k,w} \leq Tm_{k,w+1} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m, \forall w = 1, \dots, w_{k-1} \quad (19)$$

$$Ts_{i,j,k} = \frac{P_{i,j,k}}{V_k} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m \quad (20)$$

$$\sum_k P_{i,j,k} = PT_{i,j} \quad \forall i = 1, \dots, n, \forall j: a_{i,j} = 0 \quad (21)$$

$$\sum_k P_{i,j,k} * Y_{k,i,j} = PT_{i,j} \quad \forall i = 1, \dots, n, \forall j: a_{i,j} = 1 \quad (22)$$

$$Si \ b_{i,j} = 1 \rightarrow To_{i,j} + Ts_{i,j,k} \leq To_{i,j+1} \quad \forall j = 1, \dots, j_i, \forall k = 1, \dots, j_{i-1} \quad (23)$$

$$Si \ b_{i,j} = 0 \rightarrow To_{i,j} < To_{i,j+1} \quad \forall j = 1, \dots, j_i, \forall k = 1, \dots, j_{i-1} \quad (24)$$

$$Tm_{k,w} \leq To_{i,j} + (1 - X_{i,j,k,w}) * L \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m, \forall w = 1, \dots, w_k \quad (25)$$

$$Tm_{k,w} + (1 - X_{i,j,k,w}) * L \geq To_{i,j} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m, \forall w = 1, \dots, w_k \quad (26)$$

$$\sum_i \sum_j X_{i,j,k,w} = 1 \quad \forall k = 1, \dots, m, \forall w = 1, \dots, w_k \quad (27)$$

$$\sum_k Y_{k,i,j} = 1 \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i \quad (28)$$

$$X_{i,j,k,w} \in \{0,1\} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m, \forall w = 1, \dots, w_k \quad (29)$$

$$Y_{k,i,j} \in \{0,1\} \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i, \forall k = 1, \dots, m \quad (30)$$

$$To_{i,j} \geq 0 \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i \quad (31)$$

$$Ts_{i,j,k} \geq 0 \quad \forall i = 1, \dots, n, \forall j = 1, \dots, j_i \quad (32)$$

$$Tm_{k,w} \geq 0 \quad \forall k = 1, \dots, m, \forall w = 1, \dots, w_k \quad (33)$$

$$C_{max} \geq 0 \quad (34)$$

La ecuación (15) representa la función objetivo del modelo que consiste en minimizar el makespan, considerado como el intervalo de tiempo en el que se procesa completamente la totalidad de los trabajos. Las restricciones (16) obligan al makespan a tomar un valor mayor o igual al tiempo de término de cada tarea. Las restricciones (17) y (18) son definidas para el caso de los trasposos entre las diferentes plantas. Así, las restricciones (17) limitan a cada pedido a comenzar después de su tiempo de inicio preestablecido y, las restricciones (18) limitan los pedidos a terminar antes de su tiempo de fin preestablecido. Las restricciones (19) indican que cada máquina debe procesar solo una operación a la vez. Las restricciones (20) indican que el tiempo de procesamiento de la operación $O_{i,j}$ en dicha máquina sea igual a la producción dividido en la velocidad de la máquina. Las restricciones (21) y (22) limitan la capacidad para satisfacer la demanda de producción de una operación $O_{i,j}$, donde las restricciones (21) son para las máquinas que trabajan en paralelo y las restricciones (22) son

para solo una máquina del subconjunto de máquinas capaces para procesar dicha operación. Las restricciones (23) y (24) obligan a cada trabajo a seguir una secuencia de operaciones especificada, además dependiendo de su subconjunto se restringen según sus tiempos de inicio en el caso de que tenga precedencia o no. Las restricciones (25) y (26) fuerzan a que cada operación $O_{i,j}$ se pueda iniciar después de que su máquina asignada esté disponible, y que su operación anterior $O_{i,j-1}$ esté completada. Las restricciones (27) y (28) aseguran que cada operación de cada pedido sea asignada a una máquina y en una sola posición dentro de la máquina. Las restricciones (29) y (30) definen las variables como binarias. Las restricciones (31)-(34) garantizan la no negatividad de las variables.

Capítulo 4. Implementación y resultados

En este capítulo se presentan los detalles computacionales, la descripción de las instancias y los resultados.

4.1 Detalles computacionales

Los modelos propuestos fueron implementados usando Python 3.8 y los experimentos fueron ejecutados en un procesador Intel(R) Core(TM) i3-5005U (2.00GHz CPU), y 4 GB de RAM cuyo sistema operativo es Windows 10. Los modelos fueron ejecutados utilizando solo un hilo de procesamiento.

Para resolver el problema de CP se utiliza el solucionador CP-SAT, específicamente el módulo `cp_model` del paquete de herramientas Google OR-Tools. Para el caso del problema MILP se utiliza el solver CPLEX.

4.2 Instancias de prueba

Se seleccionan instancias con datos actuales de la empresa basados en la planta de remanufactura Los Ángeles. Se determinan cinco distintos escenarios.

1. El primer escenario posee 67 trabajos y un número de órdenes de producción de 10 pedidos. Dentro de estas órdenes hay seis tipos de productos distintos y los productos requieren entre tres y cinco operaciones en su fabricación. La planta cuenta con siete máquinas activas de las cuales dos pertenecen a un subconjunto de máquinas que trabajan con máquinas en paralelo, y otras dos que pertenecen a un subconjunto de máquinas que no trabajan en paralelo. El horizonte de planificación es de 552.498 segundos.
2. El segundo escenario contiene 187 trabajos y un número de órdenes de producción de 33 pedidos. Al igual que en el escenario anterior, las órdenes contienen seis tipos de productos distintos y los productos requieren entre tres y cinco operaciones en su fabricación. La planta cuenta con siete máquinas activas de las cuales dos pertenecen a un subconjunto de máquinas que trabajan con máquinas en paralelo, y otras dos que pertenecen a un subconjunto de máquinas que no trabajan en paralelo. El horizonte de planificación es de 1.992.092 segundos.

3. El tercer escenario contiene una demanda media que posee 327 trabajos y un número de órdenes de producción de 53 pedidos. Existe una variedad de 13 tipos de productos, dependiendo de este su secuencia varía entre tres y cuatro operaciones. Para este caso, todas las operaciones trabajan en serie, solo existen subconjuntos de máquinas que no trabajan en paralelo, de las cuales se debe escoger solo una para realizar la operación correspondiente. El horizonte de planificación es de 1.983.580 segundos.
4. El cuarto escenario posee 638 trabajos y un número de órdenes de producción de 133 pedidos, dentro de estos pedidos existe una variedad de 35 tipos de productos, dependiendo del producto su secuencia puede variar entre tres y cinco operaciones. Ninguna máquina trabaja en paralelo y todos los subconjuntos de máquinas contienen solo una máquina disponible para realizar la operación. El horizonte de planificación es de 5.600.354 segundos.
5. El quinto escenario contiene una demanda alta, posee 1083 trabajos y un número de ordenes de producción de 215 pedidos. Hay una variedad de 35 tipos de productos distintos, dependiendo del producto su secuencia puede variar entre tres a seis operaciones. Existen subconjuntos de máquinas en paralelo y subconjuntos de máquinas en serie. El horizonte de planificación es de 10.663.881 segundos.

Las instancias descritas anteriormente se encuentran en el siguiente link¹.

4.3 Resultados computacionales

El rendimiento de cada modelo se basa en su capacidad para encontrar buenas soluciones durante un tiempo establecido de 3600 segundos, sin considerar en donde fueron encontradas, la calidad de la solución es determinada por el valor del makespan y el gap. El gap es calculado según la Ecuación (35). En caso de tener el mismo valor de gap la solución se puede evaluar por su tiempo de ejecución.

$$gap = \frac{makespan - best\ bound}{makespan} \times 100 \% \quad (35)$$

1

https://docs.google.com/spreadsheets/d/1ffBn8IJG11rkqQ0jOOngHfjs44D_2zbmZOgrixom5jM/edit?usp=sharing

La **Tabla 5** muestra los resultados computacionales de las diferentes instancias descritas anteriormente. Donde se presenta el tipo de solución (óptima o factible), la mejor cota inferior (best bound), el makespan, el gap, y finalmente, el tiempo en el cual se encuentra la solución. Además, en la **Tabla 5** se demuestra una solución al menos factible para ambos modelos en los distintos escenarios.

Cabe destacar que los resultados se encuentran expresados en segundos.

Tabla 5: Tabla de resultados con los cinco escenarios.

		Modelo CP	Modelo MILP
Primer escenario	Solución	Óptima	Óptima
	Best Bound	166368	166354,12
	Makespan	166368	166367,99
	GAP	0%	0%
	Tiempo	0,034910	244,76
Segundo escenario	Solución	Óptima	Factible
	Best Bound	541422	266878,25
	Makespan	541421	541421
	GAP	0%	51%
	Tiempo	0,167244	3600,2
Tercer escenario	Solución	Factible	Factible
	Best Bound	467967	211393,77
	Makespan	503939	519790,85
	GAP	7%	59%
	Tiempo	3600,64	3600,12
Cuarto escenario	Solución	Factible	Factible
	Best Bound	1040745	173301
	Makespan	1047501	1148390,99
	GAP	1%	85%
	Tiempo	3600,52	3600,15
Quinto escenario	Solución	Factible	Factible
	Best Bound	1422570	147173,99
	Makespan	2245379	2347260,99
	GAP	37%	94%
	Tiempo	3600,42	3600,5

Fuente: Elaboración propia.

De los resultados obtenidos en la **Tabla 5**, es posible evidenciar que el CP entrega los mejores resultados en todos los escenarios. Para el primer escenario, entrega el mismo valor de gap, pero se puede notar que la solución óptima se encuentra en un tiempo notoriamente reducido. Si bien, ambos modelos encuentran la solución óptima, la diferencia de valores se debe a que

el CP solo trabaja con datos enteros, a diferencia del MILP que trabaja con datos reales. Además, se puede apreciar a lo largo de la **Tabla 5** que los resultados entregados por MILP, a diferencia de los resultados entregados por el modelo CP, son números reales. En el segundo escenario se puede ver que el modelo CP alcanzó a encontrar una solución óptima en un tiempo de 0,167244 segundos. Sin embargo, el modelo MILP no encuentra una solución óptima dentro del tiempo límite de 3600 segundos. Por último, en el tercer, cuarto y quinto escenario el valor de makespan encontrado es mejor para el caso del modelo CP. Además, el valor del gap es considerablemente menor para el modelo CP, dando menor al 10% en cuatro de las cinco instancias. Mientras, para el modelo MILP, solo el primer escenario obtiene un gap 0%. Para el resto de las instancias los valores son sobre el 50%, esto deja reflejado que el modelo MILP no alcanza gran parte de la máxima mejora posible.

Estos resultados muestran que el problema es correctamente modelado en la planta. A modo de comparación, las Cartas Gantt generadas de la instancia pequeña se pueden observar en las **Figuras 2 y 3**, donde se puede notar que la planificación entregada es distinta, pero ambas terminan en el mismo instante, demostrando ser de buena calidad. Para el caso de las instancias 4 y 5, que entregan una planificación que supera los 12 y 25 días respectivamente, se consideran significativas ya que se puede sacar un estimado de pedidos a programar por semana. Se puede priorizar que pedidos se realizan dentro de cada semana en caso de tener fecha límite de entrega, o en caso contrario, se da a conocer al cliente una fecha estimada de entrega.

Actualmente, en la empresa se planifica la producción operación por operación. Esto genera que se trabaje en base a datos históricos de la empresa y se limitan a recibir una mayor cantidad de pedidos. En particular, en la planta de remanufactura Los Ángeles, en un plazo de seis días se programan aproximadamente entre dos y cinco millones de piezas dependiendo de las máquinas que se habiliten. Según los resultados, la instancia 2 y 3 reflejan una planificación que rondan los seis días, pero con distintas condiciones dentro de la planta, la instancia 2 planifica casi seis millones de piezas y la instancia 3 planifica casi 10 millones de piezas, siendo esto favorable para la empresa ya que se demuestra que con el modelo aplicado se puede mejorar considerablemente la asignación de cargas de trabajos.

En anexos se puede encontrar la Carta Gantt correspondiente a cada instancia para cada modelo.

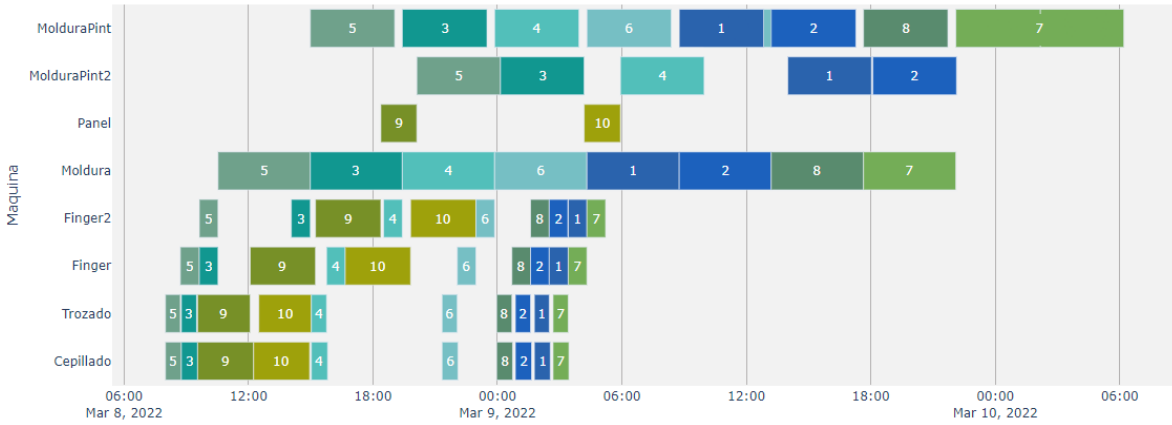


Figura 2: Resultado Carta Gantt modelo CP escenario 1.

Fuente: Elaboración propia.

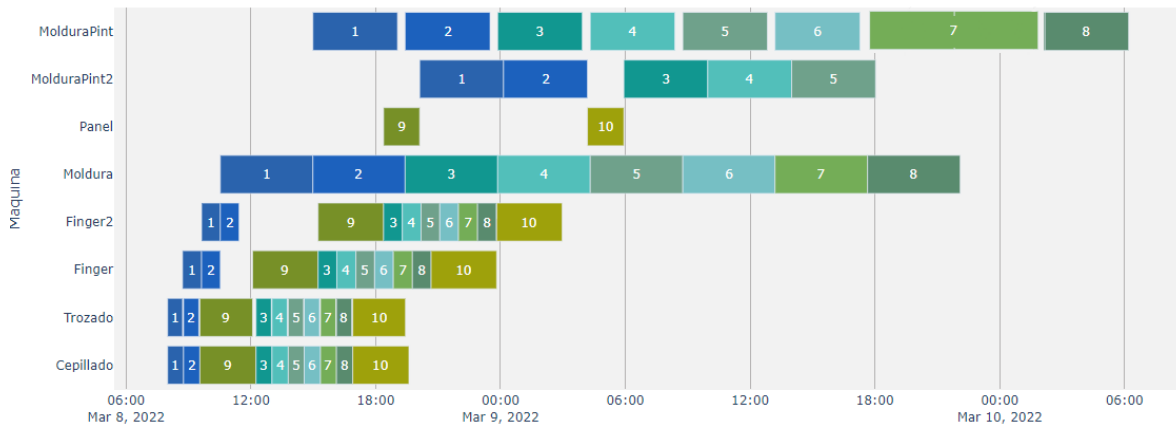


Figura 3: Resultado Carta Gantt modelo MILP escenario 1.

Fuente: Elaboración propia.

Capítulo 5. Conclusiones

En esta memoria de título se propusieron dos modelos para abordar el problema de la planificación diaria de cargas de trabajo en la planta de remanufactura Los Ángeles de CMPC Maderas, un modelo de programación lineal entera mixta y un modelo de programación por restricciones.

Se probaron diferentes instancias desde complejidad baja, con muy pocos pedidos, hasta una sobrecarga de pedidos. Ambos modelos entregan soluciones factibles dentro de los límites de tiempo para todas las instancias, incluso para las más grandes, pero evidentemente el modelo de programación por restricciones entrega mejores resultados. A medida que va aumentando la cantidad de pedidos, se va obteniendo peores resultados. Además, la solución al problema es más lenta debido a la gran cantidad de variables y restricciones que se van agregando a los modelos. La programación por restricciones como herramienta de modelado es más flexible que la programación lineal entera mixta, las funciones objetivo, así como las restricciones pueden ser de un carácter más general. Mientras que las variables de decisión pueden representar valores enteros y reales, también elementos de conjuntos e incluso subconjuntos.

Una futura mejora es aplicar una heurística o metaheurística al modelo propuesto. El uso de estas podría resultar positivo al momento de obtener buenos resultados con tiempos de ejecución considerablemente menores que los modelos.

Cabe mencionar que el modelo de programación por restricciones con la herramienta Google OR-Tools utiliza tres variables para un único intervalo, lo que ralentiza la propagación de las restricciones, siendo un problema insignificante al momento de comparar con el modelo lineal entero mixto. Además, en el modelo por restricciones se encuentra la complejidad que las herramientas de Google OR-Tools únicamente trabaja con datos enteros. Por tanto, se debe realizar una previa transformación de datos de tal forma que se pierda lo mínimo de información al momento de ejecutar los experimentos. Además, se puede deducir que el solver CP-SAT puede aplicarse con éxito al problema de planificación diaria de cargas de trabajo en la industria forestal.

Si bien, ambos modelos entregan soluciones factibles, a la empresa le conviene utilizar el modelo de programación por restricciones, esto debido a que entrega mejores soluciones en tiempos más acotados. Además, la herramienta que ofrece Google para la implementación del modelo tiene funciones interesantes que ayudan a mejorar el tiempo de ejecución del programa. Una de estas es la función no-overlap que simplifica las restricciones que prohíben que dos tareas se superpongan dentro de una máquina. Otra función que podría mejorar el modelo es la que agrega circuitos como restricciones, mediante esta función se podría mejorar la asignación de cargas de trabajo considerando los tiempos de configuración específicos por cada máquina y no aproximándolos. Cabe destacar que OR-Tools es un software de código abierto, y de acceso gratuito. Mientras que CPLEX requiere licencia comercial y alto costo, además de ser software privativo.

Referencias

- Andison, D. W., Burton, P. J., Messier, C., Smith, D. W., & Adamowicz, W. L. (2003). Tactical forest planning and landscape design. Towards sustainable management of the boreal forest, 433-480.
- Barber, F & Salido M. A. (2003) Introducción a la Programación de Restricciones. Departamento de Sistemas Informáticos y Computación.
- Birgin, E., Feofiloff, P., Fernandes, C., Melo, E., Oshiro, M., Ronconi, D. (2014). A MILP model for an extended version of the Flexible Job Shop Problem. *Optimization Letters* 8, 1417–1431.
- Bowman, E. H. (1959). The Schedule-Sequencing Problem. *Operations Research*, 7(5), 621–624.
- Campo, E.A. (2018) con tiempos difusos de procesamiento y alistamiento dependiente de la secuencia, y ventanas de tiempo en las entregas, mediante algoritmos genéticos. Trabajo de investigación para optar al título de magister en Ingeniería Industrial. Facultad de Minas-Ingeniería de la Organización, Universidad Nacional de Colombia, Colombia.
- Chen, J.C., Wu C.C., Chen C.W., & Chen K.H. (2012). Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm. *Expert Systems with Applications*, 39, 10016-10021.
- CMPC (2022). Página oficial de CMPC. URL: <https://www.cmpc.com/>
- Da Col G. & Teppan E.C. (2019). Google vs IBM: A Constraint Solving Challenge on the Job-Shop Scheduling Problem. *Universitaet Klagenfurt, Austria*. 1-7.
- Díaz-Cazaña, R., Costa-Salas, Y. J., & Sarache-Castro, W. A. (2014). Heuristic for production scheduling on job-shop plants considering preventive maintenance tasks. *DYNA*, 81(187), 215–222.
- Fattahi P, Saidi M. M. & Jolai F. (2007) Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *Journal of Intelligent Manufacturing* (2007) 18:331–342.

- Floudas, C.A. & Lin, X. (2005). Mixed Integer Linear Programming in Process Scheduling: Modeling, Algorithms, and Applications. *Annals of Operations Research* 139, 131–162.
- Goyeneche, A. (2022). Planificación de actividades silvícolas en plantaciones forestales usando programación lineal entera. Tesis de Magister. Universidad Tecnológica de Pereira, Pereira, Risaralda, Colombia.
- Jansen, K., Mastrolilli, M., & Solis-Oba, R. (2005). Approximation schemes for job shop scheduling problems with controllable processing times. *European Journal of Operational Research*, 167(2), 297–319.
- Jin X., Liang G., Kunkun P., Xinyu L., Haoran L. (2019). Review on flexible job shop scheduling. *IET Collaborative Intelligent Manufacturing*. School of Mechanical Science and Engineering, Huazhong University of Science and Technology, China. 2-19.
- Ku, W.Y. & Beck C. (2016) Mixed Integer Programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73, 165-173.
- Larozé B., A., Muñoz S., F., Pinto, P. (1998). Evaluación de diferentes modelos de optimización para la planificación de faenas silvícolas. *Bosque*, 19(1), 13–26.
- Liao C. & You C.T. (1992). An improved formulation for the job-shop scheduling problem, *Journal of the Operational Research Society*. 1047–1054.
- Manne A. S. (1960). On the job-shop scheduling problem, *Operations Research*. 8(2), 219–223.
- Osorio, J.C. & Motoa, T.G. (2008). Planificación jerárquica de la producción en un job shop flexible. Universidad de Antioquia, Medellín, Colombia. *Revista Facultad de Ingeniería Universidad de Antioquia*, núm. 44, junio, 2008, pp. 158-171.
- Pinedo, M. L. (2016). *Scheduling: Theory, algorithms, and systems* (5a ed.). Springer International Publishing. New York, NY, USA.
- Prieto, A & Díaz-Balteiro, L. (1999). Modelo de planificación forestal en la programación lineal: Aplicación al Monte Pinar de Navafría (Segovia). *Investigación agraria. Sistemas y recursos forestales*, ISSN 1131-7965, Vol. 8, N° 1, 1999, pags. 63-92.

Rivera, R. T. (2012). Programación de la producción en cartulinas CMPC S.A. Proyecto de grado para optar al grado de magíster en Ingeniería de negocios con tecnologías de información. Universidad de Chile.

Rossi, F. Van-Beeck P. & Walsh T. (2014). Handbook of constraint programming. Elsevier Science, USA.

Wagner H. M. (1959) An integer linear-programming model for machine scheduling, Naval Research Logistics Quarterly 6 (2). 131–140.

Anexos

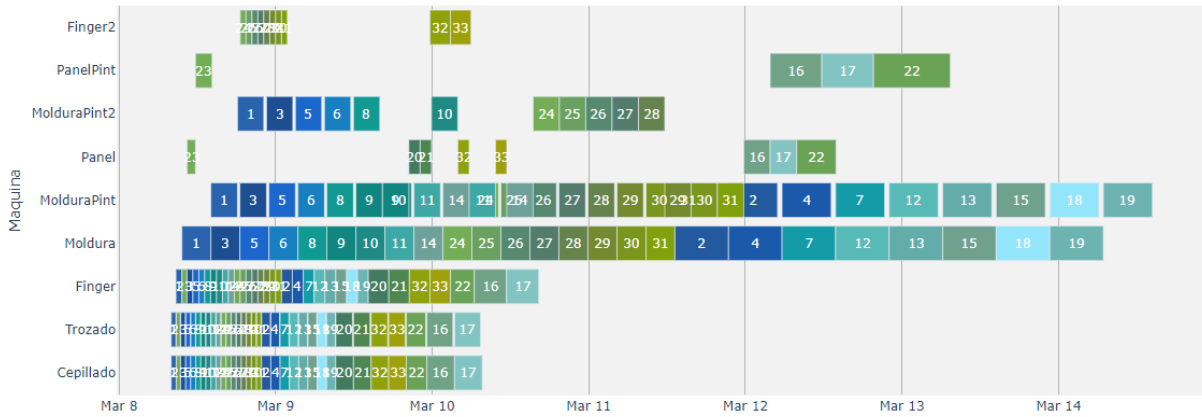


Figura A.4: Resultado Carta Gantt modelo CP escenario 2.

Fuente: Elaboración propia.

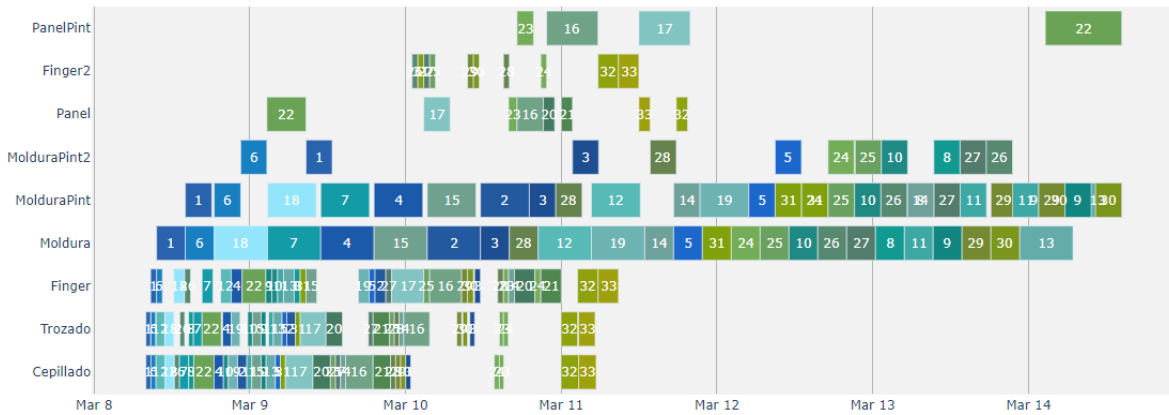


Figura A.5: Resultado Carta Gantt modelo MILP escenario 2.

Fuente: Elaboración propia.

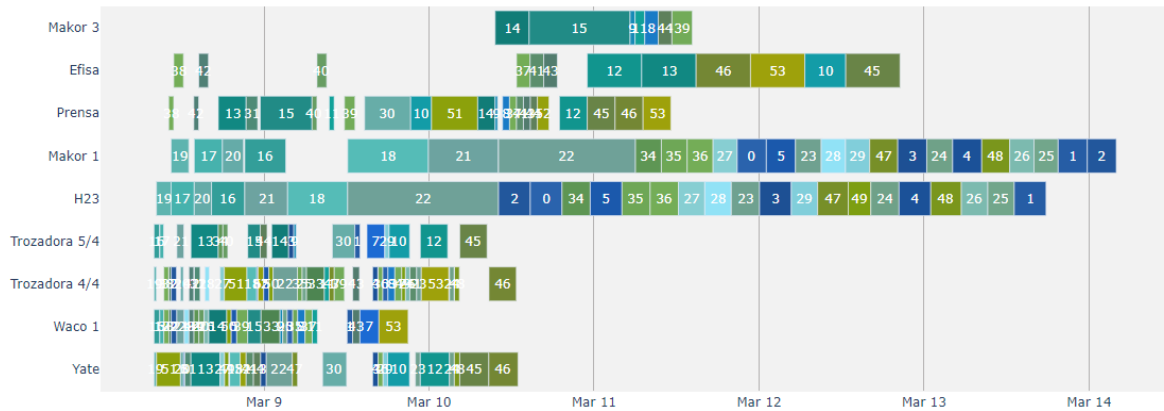


Figura A.6: Resultado Carta Gantt modelo CP escenario 3.

Fuente: Elaboración propia.

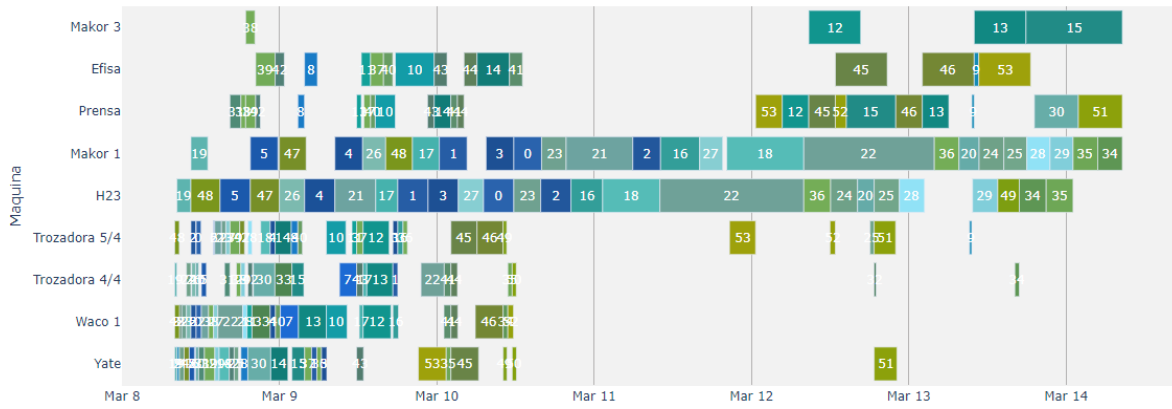


Figura A.7: Resultado Carta Gantt modelo MILP escenario 3.

Fuente: Elaboración propia.

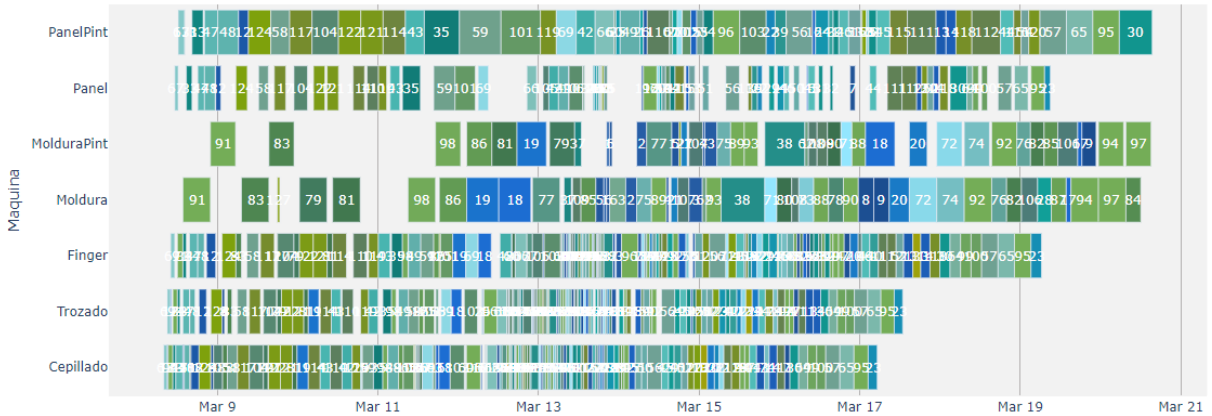


Figura A.8: Resultado Carta Gantt modelo CP escenario 4.

Fuente: Elaboración propia.

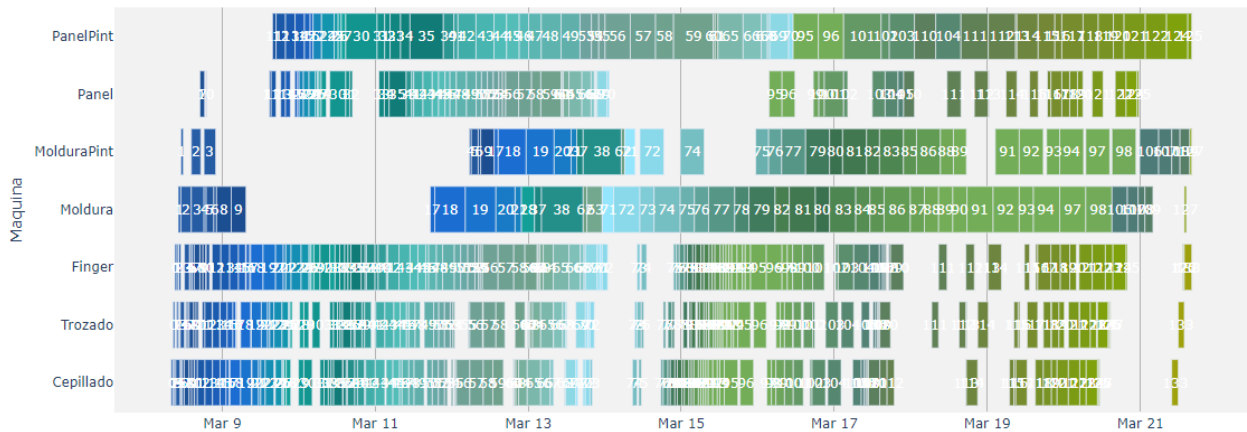


Figura A.9: Resultado Carta Gantt modelo MILP escenario 4.

Fuente: Elaboración propia.

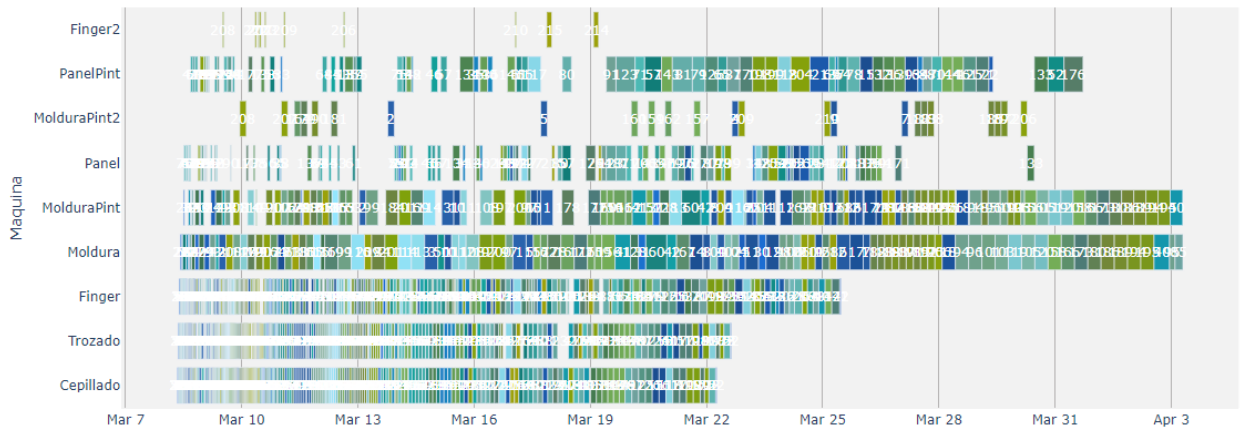


Figura A.10: Resultado Carta Gantt modelo CP escenario 5.

Fuente: Elaboración propia.

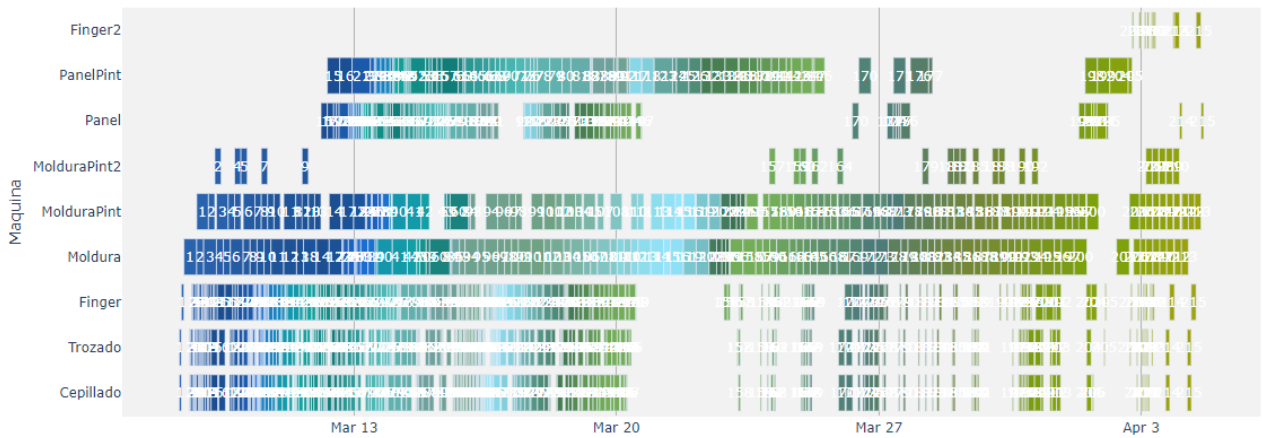


Figura A.11: Resultado Carta Gantt modelo MILP escenario 5.

Fuente: Elaboración propia.

UNIVERSIDAD DE CONCEPCIÓN – FACULTAD DE INGENIERÍA

RESUMEN DE MEMORIA DE TÍTULO

Departamento: Departamento de Ingeniería Civil Industrial

Carrera: Ingeniería Civil Industrial

Nombre del memorista: Sofía Paulina Rodríguez Retamales

Título de la memoria: Planificación diaria para la asignación de cargas de trabajo en plantas de madera en CMPC Maderas spa.

Fecha de la presentación oral: 23-03-2023

Profesor(es) Guía: Carlos Contreras Bolton

Profesor(es) Revisor(es): Carlos Herrera Lopez

Concepto:

Calificación:

Resumen
<p>La presente memoria de título tiene como objetivo abordar el problema de planificación diaria de cargas de trabajo en la planta de remanufactura Los Ángeles de CMPC Maderas. El problema es modelado como un problema de programación de taller de trabajo flexible, que consiste en asignar y secuenciar un mínimo dado de trabajos en diferentes centros de procesamiento con diferentes recursos. Adicional al FJSSP tradicional, se deben distinguir los diferentes subconjuntos de máquinas para las respectivas operaciones de cada pedido y calcular el tiempo de procesamiento de cada operación dependiendo de la(s) máquina(s) que le sean asignadas. El problema modelado es implementado mediante programación lineal entera mixta y programación por restricciones, los cuales son resueltos mediante el solucionador IBM ILOG CPLEX Optimization Studio y el solucionador CP-SAT OR-tools de Google respectivamente. Se presentan cinco instancias para el análisis, abarcando pedidos con muy baja demanda hasta pedidos con sobrecarga. Los resultados presentan soluciones óptimas para las instancias con muy baja demanda y soluciones factibles para las instancias con mayor cantidad de pedidos. Los modelos difieren en cuanto a tiempos de ejecución, valor objetivo encontrado y error relativo, siendo el modelo por restricciones el que entrega mejores resultados.</p>

