

Estado del arte del problema de secuenciación de tareas implementando reglas de despacho



Valentina Salazar Álvarez

Job scheduling problem using dispatching rules: State of the Art

Directora: Eliana Mirledy Toro Ocampo, Ph.D.

Universidad Tecnológica de Pereira
Facultad de Ciencias Empresariales
Programa de Ingeniería Industrial
Pereira, 2019

Estado del arte del problema de secuenciación de tareas implementando reglas de despacho

Valentina Salazar Álvarez

Job scheduling problem using dispatching rules: State
of the Art

Directora: Eliana Mirledy Toro Ocampo, Ph.D.

Universidad Tecnológica de Pereira
Facultad de Ciencias Empresariales
Programa de Ingeniería Industrial
Pereira, 2019

*A mis padres por apoyarme cuando todo se pone difícil
y por enseñarme lo que es el amor incondicional.*

Agradecimientos

Quiero agradecer a mi familia por siempre haber sido un apoyo en el camino, por enseñarme que el amor lo puede todo, por supuesto a mi profesora Eliana que ha sido más que una maestra, una amiga, que ha puesto toda su fe en mí y me ha demostrado que con esfuerzo se ven todos los logros.

Índice

1. Resumen	5
2. Abstract	5
3. Introducción	6
3.1. Introducción	6
3.2. Objetivos	8
3.3. Metodología	8
3.4. Justificación	8
4. Descripción del problema	10
4.1. El problema de secuenciación (<i>Scheduling</i>)	10
4.2. Clases de problemas de secuenciación	11
5. Conceptos y notación	12
5.1. <i>Scheduling</i>	12
5.1.1. Descripción de un problema de <i>Scheduling</i>	13
5.1.2. Campo α	13
5.1.3. Campo β	14
5.1.4. Campo γ	15
5.2. Sistemas de producción	16
5.2.1. Job Shop	16
5.2.2. Flow Shop	16
5.3. Taller	17
5.4. Tipos de talleres	17
5.4.1. Talleres de una sola máquina	17
5.4.2. Talleres de máquinas en paralelo	18
5.4.3. Talleres de flow shop básico	18
5.4.4. Talleres flexibles de flow shop básico	19
5.4.5. Talleres de job shop básico	19
5.4.6. Talleres flexibles de job shop básico	20
5.5. Medidas de desempeño	22
5.5.1. Relacionadas con los trabajos	22
5.5.2. Relacionadas con las máquinas	23
5.5.3. Relación entre las medidas de desempeño	24
6. Reglas de despacho	25
6.1. Reglas de prioridad simple	25
6.1.1. Relacionadas con el tiempo de proceso	25
6.1.2. Relacionadas con el tiempo de vencimiento del pedido	26
6.1.3. Relacionadas con el número de operaciones	26
6.1.4. Relacionadas con los costos	26

6.1.5.	Relacionadas el tiempo de <i>Setup</i>	26
6.1.6.	Relacionadas al tiempo de llegada y al azar	27
6.1.7.	Relacionadas con la máquina	27
6.1.8.	Información variada	28
6.2.	Combinación de reglas de prioridad simple	28
6.3.	Índices ponderados de prioridad	30
6.4.	Reglas de programación heurística	31
6.5.	Otras reglas	33
7.	Marco teórico	35
7.1.	Tipos de Diagrama de Gantt	36
7.2.	Modelo matemático del problema	40
7.3.	Heurísticas y algoritmos	43
7.3.1.	Algoritmo de Moore	43
7.3.2.	Algoritmo de Lawler	43
7.3.3.	Giffler y Thomson	43
7.3.4.	Algoritmo de Johnson	44
7.3.5.	Heurística de cambios de cuello de botella	44
7.3.6.	Algoritmo de Page/Palmer	45
7.3.7.	Método de Gupta	45
7.3.8.	Heurística CDS	45
7.3.9.	Heurística NEH	45
7.3.10.	Método de ruta crítica CPM (<i>Critical Path Method</i>)	46
7.3.11.	Algoritmo de Johnson	46
7.3.12.	Branch and Bound (Secuenciación)	46
8.	Casos de implementación	55
9.	Análisis de resultados	104
10.	Conclusiones	105

Índice de figuras

1.	Cadena de suministro al interior de la empresa	10
2.	Taller de una sola máquina	17
3.	Taller de máquinas en paralelo	18
4.	Taller de flow shop básico	18
5.	Taller flexible de flow shop básico	19
6.	Taller de job shop básico	19
7.	Taller flexible de job shop básico	20
8.	Técnicas de solución para secuenciación de tareas	35
9.	Gantt orientado a máquinas	36

10.	Gantt orientado a trabajos	36
11.	Reglas de despacho LEKIN	39
12.	Heurísticas LEKIN	39
13.	Gráfica del problema PLEM	40
14.	Gráfica del problema PL	40
15.	Restricción disyuntiva 1	41
16.	Restricción disyuntiva 2	41
17.	Branch & Bound para flow shop (1)	47
18.	Branch & Bound para flow shop (2)	48
19.	Branch & Bound para flow shop (3)	49
20.	Información caso flow shop	55
21.	Implementación regla de despacho FCFS y LCFS	56
22.	Implementación regla de despacho SPT y LPT	57
23.	Implementación regla de despacho RANDOM	57
24.	Resumen medidas de desempeño	58
25.	Diagrama de Gantt para flow shop	58
26.	Evaluación de medidas de desempeño	59
27.	Secuencia 1	60
28.	Secuencia 2	61
29.	Secuencia 3	61
30.	Primera iteración	65
31.	Segunda iteración	66
32.	Tercera iteración	67
33.	Cuarta iteración	68
34.	Quinta iteración	69
35.	Sexta iteración	70
36.	Séptima iteración	71
37.	Octava iteración	72
38.	Novena iteración	73
39.	Décima iteración	74
40.	Primera iteración	75
41.	Segunda iteración	76
42.	Tercera iteración	77
43.	Cuarta iteración	78
44.	Quinta iteración	79
45.	Sexta iteración	80
46.	Séptima iteración	81
47.	Octava iteración	82
48.	Novena iteración	83
49.	Décima iteración	84
50.	Interfaz LEKIN con implementación	85
51.	Interfaz LEKIN con implementación y medidas de desempeño	86
52.	Gantt con la regla de despacho SPT	87

53.	Gantt con la regla de despacho LPT	88
54.	Secuencia Job shop 1-2-3 y 2-1-3	89
55.	Secuencia Job shop 3-2-1 y 1-3-2	90
56.	Secuencia Job shop 2-3-1 y 3-1-2	91
57.	Descripción del grafo disyuntivo	92
58.	Ruta crítica inicial	93
59.	B&B máquina 2	95
60.	Recalculo de ruta crítica con máquina 2	96
61.	B&B máquina 1	97
62.	Recálculo de ruta crítica con máquina 1	98
63.	B&B máquina 3	98
64.	Recalculo de ruta crítica con máquina 3	99
65.	B&B máquina 4	100
66.	Diagrama de Gantt job shop	102

Índice de cuadros

1.	Branch & Bound aplicado a flow shop	46
2.	Resumen de la literatura NPFS	52
3.	Resumen de resultados heurística cuello de botella	101
4.	Tabla comparativa de métodos	103

1. Resumen

Este proyecto de investigación es un estado del arte que busca ampliar la información sobre el problema de secuenciación de tareas, teniendo en cuenta las distintas medidas de desempeño y las reglas de despacho usadas actualmente y también a lo largo de las décadas. [1]

Con el pasar del tiempo se ve como la tecnología y las teorías frente a la producción evolucionan, cada día se buscan nuevas maneras de optimizar los procesos y mejorar cada una de las áreas que conforman una organización. Este proyecto busca darle una cobertura amplia al problema de secuenciación de tareas usando reglas de despacho, que pueden ser aplicadas mediante distintos métodos como heurísticas y algoritmos, todos ellos encaminados a la optimización de distintos recursos como por ejemplo el tiempo y el costo.

La optimización del proceso de producción es una tarea importante que debe resolverse en la planificación estratégica y/u operativa de cada empresa industrial. [2] Todo este concepto lleva a la toma de decisiones, que es vital en cualquier compañía ya que de esta manera se vincula la parte productiva a la gerencial como se evidencia en este proyecto. Las decisiones van ligadas de la mano de distintas áreas para lograr la optimización.

2. Abstract

The following research project is a state of the art that seeks to expand the information on the problem of job scheduling, taking into account the different performance measures and the dispatching rules currently used, also over the years.

Over the decades we see that technology and theories regarding production evolve, every day new ways are sought to optimize processes and improve each of the areas that make up an organization. This project seeks to give a wide coverage to the problem of job scheduling using dispatching rules, which can be applied through different methods such as heuristics and algorithms, all of them aimed at optimizing different resources such as time and cost.

The optimization of the production process is an important task that must be solved in the strategic and operational planning of each industrial company. This whole concept leads us to decision making, which is vital in any company since in this way the productive part is linked to management as evidenced in this project. The decisions are linked by the hand of different areas to achieve optimization.

3. Introducción

3.1. Introducción

La secuenciación según el Diccionario de la lengua española (DLE) hace referencia a una continuidad o sucesión ordenada de algo. Por lo tanto, cuando se habla de secuenciación de tareas se hace referencia a una serie de sucesiones que van a guardar una relación entre sí y con un orden especial. La secuenciación de tareas (*scheduling*) surge en la época de la revolución industrial con la producción en serie, pues se buscaba la manera de volver más eficiente el flujo de producción. A pesar de que se había producido mucho antes de la revolución, este fue un factor determinante para continuar con la evolución de la secuenciación. La toma de decisiones es muy importante dentro de una organización, pues con una secuenciación idónea se realizan mejoras a nivel productivo. “La secuenciación y la programación son una forma de toma de decisiones que desempeña un papel crucial en las industrias manufactureras y de servicios” [3] En la actualidad la necesidad de programar y ordenar de manera eficiente la producción es un problema diario, pero esta planeación no sólo se da en la esfera industrial sino en nuestra vida cotidiana, tomar decisiones de programación se llevan a cabo todos los días. Los problemas que implican la secuenciación de tareas suelen ser complejos y extensos, y desde los años 50 han tenido a los científicos creando distintos métodos de solución. Pero uno de sus pioneros Henry Gantt aportó un diagrama sencillo con el cual se puede facilitar la visualización de los problemas de secuenciación, hoy en día este diagrama lleva su apellido, diagrama de Gantt “el más pronto y mejor organizado tipo de diagrama de control para mostrar las relaciones entre lo planificado y lo realmente acontecido” [4]. Este diagrama contribuyó en gran medida a este tipo de problemas, pero no fue hasta la aparición de la investigación de operaciones que se consolidó aún más la secuenciación de tareas. “Las primeras actividades formales de investigación de operaciones (IO) se iniciaron en Inglaterra durante la Segunda Guerra Mundial, cuando un equipo de científicos empezó a tomar decisiones con respecto a la mejor utilización del material bélico. Al término de la guerra, las ideas formuladas en operaciones militares se adaptaron para mejorar la eficiencia y productividad en el sector civil.” [5]

Con el paso de los años la secuenciación se volvió más estudiada y en los años 50 se comenzaron a crear algoritmos que buscaban secuenciar las tareas; en este caso se tiene el algoritmo de Johnson, *Short Time Process (SPT)* y *Earliest Due Date (EDD)* [6]. Una década después, la programación lineal contribuyó creando alternativas de solución a los problemas y generó la innovación tecnológica de softwares que los resolvieran.

Para finales 1970 Michael Garey y David Johnson crearon la teoría de la complejidad computacional que es una rama de la teoría de la computación donde clasifican los problemas computacionales dependiendo de su dificultad y su estructura. Años más tarde publicaron su libro sobre los problemas NP-Difícil [6] dentro de los que se denominan el problema de secuenciación de tareas.

La teoría de la complejidad computacional analiza el tiempo y recursos que serán consumidos en la solución de un problema y dependiendo de ellos los categoriza en distintas

clases de complejidad como lo son, P y NP. [7]

En la actualidad los modelos matemáticos, las heurísticas, los algoritmos y los softwares permiten un acercamiento a la solución de este tipo de problemas.

Este proyecto de investigación se centra en el estado del arte del problema de secuenciación de tareas, analizando la literatura actual y revisando metodologías para resolver problemas de Flow shop y Job shop para posteriormente validarse los resultados con el software académico disponible LEKIN. [8]

3.2. Objetivos

Objetivo general

Realizar una revisión del estado del arte del problema de secuenciación de tareas donde se identifiquen sus variantes y estrategias de solución.

Objetivos específicos

- Identificar las medidas de desempeño asociadas a los problemas de secuenciación de tareas.
- Describir las reglas de despacho de la literatura actual.
- Identificar los elementos esenciales de un problema de programación secuencial.

3.3. Metodología

- Construcción del estado del arte del problema de secuenciación de tareas a partir de la literatura.
- Análisis de conceptos y notación para el problema.
- Revisión de modelos matemáticos que permitan la abstracción a la realidad del problema.
- Identificación de las reglas de despacho y heurísticas existentes en la literatura con sus aplicaciones.
- Implementación de distintos problemas reales utilizando reglas de despacho.
- Verificación con software de los resultados obtenidos por los distintos métodos (Heurísticas y reglas).

3.4. Justificación

El problema de secuenciación de tareas es un tema imprescindible en las empresas productoras de bienes, pues afecta directamente el flujo de producción, que es un área de vital importancia en la toma de decisiones estratégicas y tácticas. Con un flujo de producción optimizado, los procesos se realizarán de una manera rápida minimizando distintas variables como el tiempo total de las tareas, retrasos, tardanzas y demás. Todo esto cumpliendo con las fechas de vencimiento (*Due Dates*).

Esta fecha de vencimiento puede ser un parámetro dado del problema o puede ser una variable, es decir, estamos interesados en una fecha de vencimiento óptima con respecto a la función objetivo. [9] Esto podrá ayudar a la organización a maximizar su valor en el mercado y mejorar de una manera eficaz el sistema interno.

Se pretende realizar una revisión del estado del arte de la temática para identificar líneas de investigación teórica y aplicada, de esta manera ampliar la información sobre las reglas de despacho existentes pues muchas empresas al día de hoy aún no cuentan con la información necesaria para llegar a implementar mejoras en sus procesos. Las compañías siguen haciendo uso de métodos empíricos, que podrían ser optimizados por las distintas técnicas de solución. Por lo cual el estado del arte planteado ayudaría a mejorar procesos existentes para ser implementados por los trabajadores y la empresa.

4. Descripción del problema

4.1. El problema de secuenciación (*Scheduling*)

La secuenciación (*Scheduling*) es un proceso de gestión donde se tiene una serie de tareas y unas máquinas o procesos, donde se van a llevar a cabo cada una de las actividades. No son solamente tareas y máquinas, pueden ser distintas variables que involucren una actividad y un proceso, con esto se realizará una planeación de cada una de las tareas.

Como se muestra en la **figura 1** se logra ver que las decisiones a tomar con la secuenciación están ligadas a nuestra producción (fabricación) y se hacen en un lapso corto de tiempo.



Figura 1: Cadena de suministro al interior de la empresa

Los problemas de programación se pueden entender de manera muy amplia como los problemas de asignación de recursos a lo largo del tiempo para realizar un conjunto de tareas que forman parte de algunos procesos, entre los cuales los más importantes son los de computación y fabricación [10].

La programación de producción tiene como objetivo maximizar la eficiencia de la operación y reducir los costos. Muchos problemas de programación tienen clases isomórficas de soluciones, es decir, clases de soluciones de costo idéntico que difieren solo intercambiando un valor de recurso particular por otro a lo largo del programa [11].

El *scheduling* para todos los sistemas de producción [12]:

- Son decisiones complejas y su complejidad seguirá aumentando debido a la tendencia actual de incrementar la flexibilidad en los pedidos de los clientes, lo que deriva en aumentar la complejidad del proceso productivo.
- Tiene horizonte temporal corto, el ciclo de vida de un ordenamiento o schedule suele ser muy corto.
- A pesar de ser decisiones de horizontes cortos, suelen ser muy importantes para la línea de producción de una organización ya que impactan directamente sobre los costos de producción y los tiempos de entrega, lo que afecta a la competitividad de la empresa (tiempo de entrega y costos).
- Por estar situados en el centro de las operaciones de producción, las restricciones y objetivos del scheduling son muy dependientes del tipo de empresa.
- Suelen ser decisiones estructuradas, lo que permite que la información, los criterios y las limitaciones sean fáciles de formalizar

4.2. Clases de problemas de secuenciación

Los problemas de la secuenciación de tareas pueden ser de distintas clases como lo son [13]:

- Problemas estáticos: El número de tareas y sus correspondientes llegadas ya se conocen y no varían.
- Problemas dinámicos: La llegada de las tareas es desconocida y aleatoria.
- Problemas determinísticos: Los valores de las variables independientes son ya conocidos.
- Problemas aleatorios: Las variables independientes son de carácter aleatorio.

Los problemas estáticos y dinámicos están relacionados con la llegada de las tareas mientras los problemas determinísticos y aleatorios se relacionan con la naturaleza de las variables independientes del problema.

5. Conceptos y notación

5.1. *Scheduling*

El *scheduling* se encarga de la asignación de recursos a las tareas que deben realizarse durante un horizonte de tiempo para concretar un plan de producción, y los horizontes temporales suelen ser menores a una semana. [12]

La secuenciación es en esencia resuelta con un Problema de Optimización con Restricciones (COP, según sus siglas en inglés) que en el entorno de la fabricación significa encontrar la secuencia de colocación de trabajos, Π_j en cada uno de los recursos que optimicen una función objetivo. [14]

Como lo plantean los autores Rossit, Tohmé y Frutos [15] en toda esta sección a tratar los problemas de programación de los sistemas de producción se han analizado extensamente y se han resuelto bajo diferentes enfoques. [16; 17; 18; 19; 20; 21] Los resultados en este campo han contribuido a la mejora de los sistemas de fabricación [22]

El sistema flow shop es común en la fabricación. Configuraciones donde un conjunto de trabajos $N = \{1, 2, \dots, n\}$ son procesados por una conjunto de máquinas $M = \{1, 2, \dots, m\}$. Cada trabajo pasa por las máquinas en el mismo orden tecnológico, es decir, comienza en la máquina 1, luego va a la máquina 2, ... hasta la máquina m . La decisión a tomar es elegir el orden en que los diferentes trabajos pasarán a través de las máquinas. Si la secuencia de trabajo es la misma para todas las máquinas, la programación se denomina permutación y el problema de elegir la mejor se conoce como el problema de flow shop de permutación (*PFS*). Si, por el contrario, la secuencia de procesamiento puede cambiar de una máquina a otra, la condición de permutación se relaja y el problema se conoce como flow shop sin permutación (*NPFS*). La descripción estándar del problema NPFS considera lo siguiente:

1. Cada máquina puede procesar solo un trabajo a la vez.
2. Cada trabajo j tiene un tiempo de procesamiento p_{ij} en la máquina $i = \{1, 2, \dots, m\}$
3. La capacidad de los buffers intermedios debe ser lo suficientemente grande como para permitir la reordenación de la secuencia de trabajo.

La configuración estándar de *NPFS* y *PFS* es muy similar, siendo el tercer elemento la diferencia potencial más relevante entre ellos. En algunos casos, los problemas de *PFS* suponen también buffers intermedios con capacidad ilimitada, siendo perfectamente compatibles con *NPFS*. Por otro lado, en ausencia de búferes intermedios, el enfoque *NPFS* no es aplicable para obtener un esquema de programación factible (lo mismo ocurre con el caso de la flow shop "sin espera" (*non-wait flow shop*)). [23]

Además de las tres especificaciones principales mencionadas en la forma estándar *NPFS*, existen otros requisitos: todos los trabajos y máquinas deben estar disponibles desde el principio; no se permite la preferencia; Las máquinas pueden estar inactivas durante el horizonte de planificación; cada trabajo puede ser procesado por una sola máquina a la vez; y los datos del problema son deterministas y conocidos de antemano. Esta descripción

no abarca todo el ámbito de los problemas de *NPFS*, pero sirve como una plantilla para ellos. Con cambios menores (como agregar o eliminar restricciones), se pueden obtener todas las diferentes variantes de *NPFS*.

5.1.1. Descripción de un problema de *Scheduling*

Las siguientes secciones se basan en la documentación presentada por la autora Salto [24], un problema de *scheduling* se describe por α, β, γ . El campo α describe el ambiente de máquina y contiene una única entrada. El campo β provee detalles de las características de procesamiento y restricciones; puede tener una única entrada, múltiples entradas o ninguna entrada. El campo γ identifica el objetivo a ser minimizado y usualmente contiene una única entrada.

5.1.2. Campo α

En este campo se encuentran distintos ambientes de máquina que pueden ser [25]:

- Máquina única (1). El caso de máquina única es el ambiente de máquina más simple y es el caso especial de todos los demás ambientes de máquina.
- Máquinas idénticas en paralelo (P_m). Hay m máquinas idénticas en paralelo. El trabajo j necesita una única operación y se puede procesar en cualquiera de las m máquinas o sobre alguna de un conjunto especificado. Si no se permite el procesamiento del trabajo j sobre cualquier máquina o sobre sólo alguna perteneciente a un subconjunto dado (el subconjunto M_j), entonces en el campo β aparece la entrada M_j .
- Máquinas en paralelo con diferentes velocidades (Q_m). Hay m máquinas en paralelo con diferentes velocidades; v_i indica la velocidad de la máquina i . El tiempo de permanencia p_{ij} del job j en la máquina i es p_j/v_i , asumiendo que sólo se procesa en la máquina i . Este ambiente también se lo conoce como máquinas uniformes. Si todas las máquinas tienen la misma velocidad, es decir $v_i=1$ para todo i y $p_{ij}=p_j$, entonces este ambiente es idéntico al previo.
- Máquinas no relacionadas en paralelo (R_m). Este ambiente es una generalización del anterior. Hay m máquinas diferentes en paralelo. La máquina i puede procesar el trabajo j a una velocidad v_{ij} . El tiempo de permanencia p_{ij} del trabajo j sobre la máquina i es p_j / v_{ij} , asumiendo que sólo se procesa sobre la máquina i . Si la velocidad de las máquinas son independientes de los trabajos, es decir $v_i=v_{ij}$ para todo i y j , entonces el ambiente es idéntico al anterior.
- Flow shop (F_m). Hay m máquinas en serie, cada trabajo se procesa en cada una de ellas. Todos los trabajos tienen la misma trayectoria, es decir, primero se procesan sobre la máquina 1, luego sobre la máquina 2, y así sucesivamente. Cuando un trabajo deja de usar una máquina se agrega a la cola de la próxima máquina. Usualmente,

se asume que todas las colas trabajan bajo la disciplina primero en entrar primero en salir (FIFO). Si se aplica la regla FIFO, al flow shop se lo denomina como flow shop con permutación y el campo β incluye la entrada *prmu*.

- Flexible flow shop (*FFs*). Es una generalización del flow shop y del ambiente de máquinas paralelas. En lugar de m máquinas en serie, hay s etapas en serie con una determinada cantidad de máquinas en paralelo en cada una de ellas. Cada trabajo se procesa primero en la etapa 1, luego en la etapa 2, y así siguiendo. Cada etapa funciona como un banco de máquinas paralelas; en cada etapa el trabajo j necesita sólo una máquina y cualquier máquina puede procesar cualquier trabajo. Las colas entre etapas distintas trabajan bajo la regla FIFO.
- Open Shop (O_m). Hay m máquinas y n trabajos. Cada trabajo se debe procesar en cada una de las m máquinas. Sin embargo, algunas de esos tiempos de procesamiento pueden ser cero. No hay restricción en relación al ruteo de cada trabajo a través del ambiente de máquinas. El programador determina la ruta de cada trabajo, diferentes trabajos pueden tener distintas rutas.
- Job shop (J_m). Hay m máquinas y n trabajos, cada trabajo tiene predeterminada su ruta. Se hace una distinción entre job shops donde cada trabajo puede visitar alguna máquina al menos una vez y aquellos donde un trabajo puede visitar una máquina más de una vez. En el último caso, el campo β contiene la entrada *recre* para indicar *recirculación*.

5.1.3. Campo β

Las restricciones de procesamiento que se pueden especificar en este campo pueden incluir múltiples entradas. Las posibles entradas son:

- Release time (r_j). Si este símbolo está presente en el campo β , el trabajo j no puede empezar su procesamiento antes de su *release date* r_j . Si r_j no aparece en el campo β , el procesamiento del trabajo j puede comenzar en cualquier momento. En contraste con el *release dates*, los tiempos de entrega no se especifican en este campo. El tipo de función objetivo da información suficiente si se consideran tiempos de entrega o no.
- Tiempos de *setup* dependientes de la secuencia (s_{jk}). s_{jk} representa el tiempo de *setup* dependiente de la secuencia de los trabajos j y k ; s_{0k} indica el tiempo de *setup* para el trabajo k si es el primero en la secuencia y s_{j0} hace referencia al tiempo de *clean-up* luego del trabajo j si este es el último de la secuencia (tanto s_{0k} como s_{j0} pueden ser 0). Si el tiempo de *setup* entre los trabajos j y k depende de la máquina, entonces se debe incluir el subíndice i , es decir s_{ijk} . Si s_{0k} no aparece en el campo β , se asume que todos los tiempos de *setup* son cero o independientes de la secuencia, en cuyo caso se pueden simplemente incorporar a los tiempos de procesamiento.

- Restricciones de precedencia (*prec*). Las restricciones de precedencia pueden aparecer en un ambiente de máquina única o máquinas paralelas, exigiendo que uno o más trabajos tengan que finalizar antes que otros comiencen su procesamiento. Hay varias formas de restricciones de precedencia. Si cada trabajo tiene al menos un antecesor y un sucesor, las restricciones se denominan *cadena*. Si cada trabajo tiene al menos un sucesor, las restricciones se denominan *intree*. Si cada trabajo tiene al menos un antecesor, las restricciones se denominan *outtree*. Si en el campo β no aparece *prec*, los trabajos no están sujetos a restricciones de precedencia.
- Permutaciones (*prmu*). Una restricción que puede aparecer en el ambiente flow shop es que la cola de cada máquina trabaje respetando la regla FIFO. Esto implica que el orden (o permutación) en la cual se asignan los trabajos a la primera máquina se debe respetar en todas las demás.
- Recirculación (*recrc*). Puede ocurrir en job shop, cuando un trabajo puede visitar más de una vez una máquina.

5.1.4. Campo γ

El objetivo es minimizar una función de tiempos de finalización de los trabajos, los cuales dependen del *schedule*. El tiempo de finalización de la operación de un trabajo j sobre la máquina i se denota como C_{ij} . El tiempo en el que el trabajo j sale del sistema se indica como C_j . La función objetivo puede ser una función de tiempos de entrega. La *lateness* de un trabajo j se define como se muestra en la ecuación número 1:

$$L_j = C_j - d_j \quad (1)$$

la cual es positiva cuando el trabajo j finaliza tarde y negativo cuando se completa en forma temprana. La *tardiness* de un trabajo j se define como se muestra en la ecuación número 2:

$$T_j = \max(C_j - d_j, 0) = \max(L_j, 0) \quad (2)$$

La diferencia entre *tardiness* y *lateness* radica en que la *tardiness* nunca es negativa. Y la unidad de penalidad del trabajo j se define como se muestra en la ecuación 3:

$$U_j = \begin{cases} 1 & \text{Si } C_j > d_j \\ 0 & \text{Otro caso} \end{cases} \quad (3)$$

Lateness, *tardiness* y unidad de penalidad son las tres funciones de penalidad básicas relacionadas con tiempos de entrega.

Los siguientes son algunos ejemplos de funciones objetivos a ser minimizadas:

- *Makespan* (C_{max}). Se define como el $\max(C_1, C_2, \dots, C_n)$. Es equivalente al tiempo de finalización del último trabajo en dejar el sistema. Un *makespan* mínimo usualmente implica una alta utilización de las máquinas.

- *Lateness máxima* (L_{max}). Se define como el $\max(L_1, L_2, \dots, L_n)$. Mide el peor incumplimiento de los tiempos de entrega.

5.2. Sistemas de producción

Existen distintos sistemas de producción en este trabajo de investigación se revisarán los sistemas job shop y flow shop.

5.2.1. Job Shop

Mucha variedad y poco volumen en la producción. Se caracteriza porque no todas sus tareas tienen el mismo proceso de fabricación.

5.2.2. Flow Shop

Poca variedad y mucho volumen en la producción. Se caracteriza porque todas sus tareas tienen la misma secuencia de producción.

5.3. Taller

Un taller es un espacio destinado a la realización del proceso productivo, en el se encuentran todos los recursos necesarios para llevarlo a cabo. Existen distintos tipos de talleres los cuales tienen una disposición de planta diferente.

5.4. Tipos de talleres

Se encuentran talleres de varias clases y dependiendo del tipo de producción que se quiera realizar. Se tienen los siguientes tipos de taller:

5.4.1. Talleres de una sola máquina

Como se referencia en la **figura 2** se aprecia un taller con n trabajos y una única máquina en el proceso.

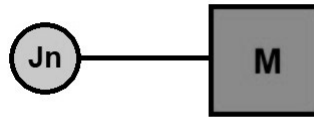


Figura 2: Taller de una sola máquina

5.4.2. Talleres de máquinas en paralelo

En la **figura 3** se muestra un taller de máquinas en paralelo las cuales pueden operar simultáneamente realizando trabajos y sin una restricción de precedencia como en el caso de un flow shop básico.

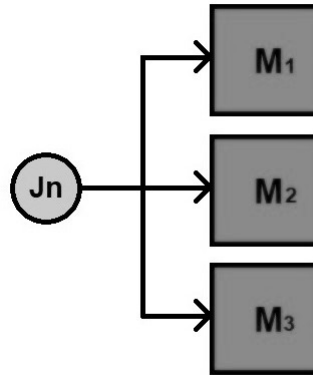


Figura 3: Taller de máquinas en paralelo

5.4.3. Talleres de flow shop básico

Se muestra en la **figura 4** un taller flow shop básico, se caracteriza porque los trabajos realizan la misma secuencia en todo el proceso.



Figura 4: Taller de flow shop básico

5.4.4. Talleres flexibles de flow shop básico

En la **figura 5** se relaciona un taller flow shop básico flexible, el cual relaciona máquinas en paralelo dispuestas en varias etapas del proceso. En cada una de las etapas se encuentran máquinas idénticas en paralelo.

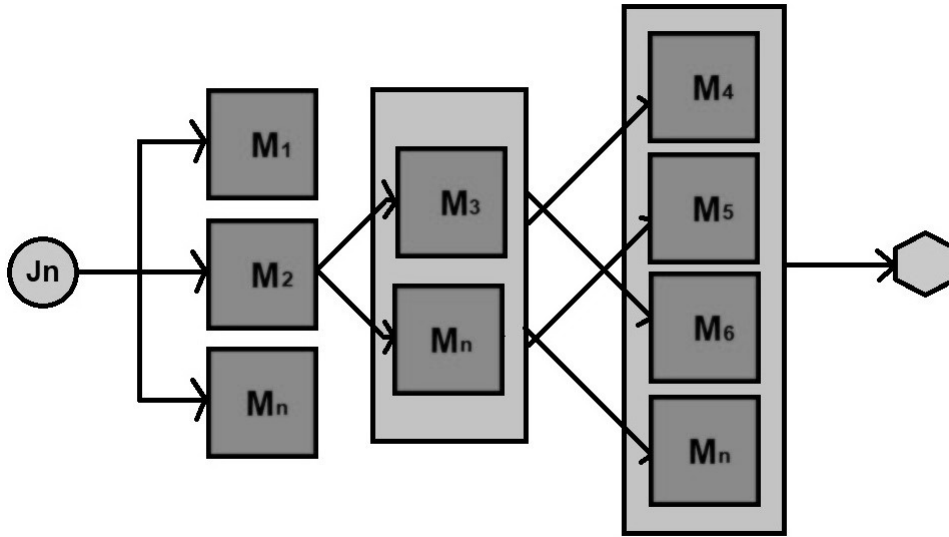


Figura 5: Taller flexible de flow shop básico

5.4.5. Talleres de job shop básico

En la **figura 6** se muestra la distribución de un taller job shop básico en el cual los trabajos no tienen la misma secuencia en sus máquinas como es el caso del flow shop.

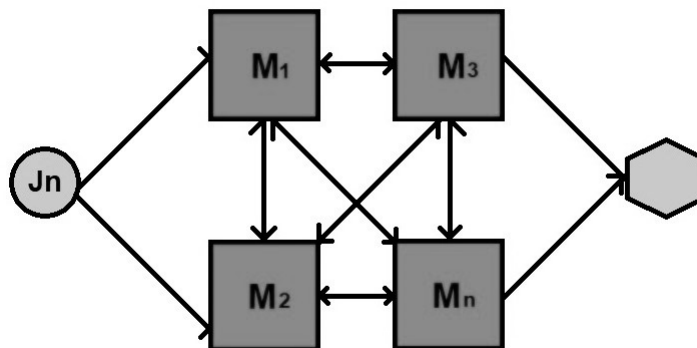


Figura 6: Taller de job shop básico

5.4.6. Talleres flexibles de job shop básico

Para la **figura 7** se referencia un taller job shop flexible en el cual se tiene un grupos de máquinas y se debe decidir sobre cuáles grupos se van a realizar las operaciones.

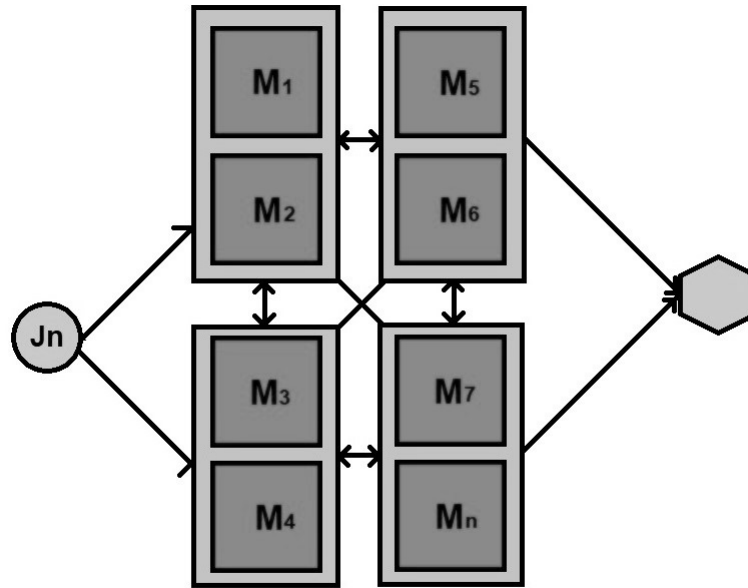


Figura 7: Taller flexible de job shop básico

Donde $J_1 \dots J_n$ representa el trabajo n a desempeñar.

De manera general para este tipo de problemas se denotarán los conjuntos de máquinas y trabajos.

M_i : Conjunto de todas las máquinas ($i=1\dots m$)

J_j : Conjunto de todas las trabajos ($j=1\dots n$)

Una secuenciación se realiza cuando se tienen n trabajos con sus unidades temporales y deben ser procesados por una o varias máquinas m .

A cada trabajo se le realiza una operación por lo cual se obtendrá la siguiente información de cada uno de ellos:

- Tiempo de inicio: SO_{ij}
- Tiempo de fin: EO_{ij}
- Ruta de procesado: R_j
- Tiempo de proceso de cada operación (tiempo de ciclo): P_{ij} siendo dependiente de la máquina, de lo contrario su notación es P_j
- Tiempo de llegada al sistema: r_{ij} , si no es diferente para cada máquina entonces su notación será r_j
- Fecha de entrega (trabajo finalizado): d_j
- Prioridad: w_j
- Disponibilidad de maquinaria: $rm_i \leq SO_{ij}$

5.5. Medidas de desempeño

Son herramientas de gestión que permiten comparar nuestras metas con el desempeño real que se ha obtenido, se utilizan en todas las áreas. En producción se tienen medidas de desempeño relacionadas tanto a máquinas como a trabajos.

5.5.1. Relacionadas con los trabajos

- Tiempo total de espera de los trabajos: W
- Tiempo total del trabajo: W_j
- Tiempo de terminación del trabajo: C_j
- Tiempo de flujo del trabajo: F_j

Siendo calculado de la siguiente manera: $F_j = C_j - R_j$

- Tiempo de terminación del conjunto de trabajos (lapso de producción o makespan): C_{max}
- Retraso del trabajo: L_j

Siendo calculado de la siguiente manera: $L_j = C_j - d_j$

- Tardanza del trabajo: T_j priman los valores positivos de L_j , por lo tanto: $T_j = \max(0, L_j)$
- Adelanto del trabajo: E_j , priman los valores negativos de L_j , por lo tanto: $E_j = \max(0, -L_j)$
- Condición tardía del trabajo: U_j de carácter binario que toma el valor de 1 si un trabajo llega tarde y 0 si no.

$U_j = 1$ si $T_j > 0$ y 0 si no.

- Número de trabajos tardíos: NT

5.5.2. Relacionadas con las máquinas

- $I^B(m)$: Tiempo ocioso de la máquina m antes de ser asignado el primer trabajo.
- I^B : Tiempo ocioso de todas las máquinas antes de ser asignado el primer trabajo.
- $I(m)$: Tiempo ocioso de la máquina m desde su disponibilidad hasta la terminación de todos los trabajos.
- I : Tiempo ocioso de todas las máquinas desde su disponibilidad hasta la terminación de todos los trabajos.

Nota: La terminación de todos los trabajos equivale al *makespan* (C_{max}).

Para este caso, se puede tener un modelo matemático **multiobjetivo** donde normalmente las principales variables a tratar son:

- $\sum C_j$: Tiempo total de todos los trabajos (Makespan)
- $\sum L_j$: Tiempo total de todos los retrasos
- $\sum T_j$: Tiempo total de todas las tardanzas
- $\sum E_j$: Tiempo total de todos los adelantos
- $\sum U_j = NT$: Número total de trabajos tardíos

5.5.3. Relación entre las medidas de desempeño

Cuando se realiza la optimización de estas variables se puede ver que existen distintos resultados como [26]:

- **Equivalentes:** Se tienen dos casos:
El primero es cuando \bar{C} , \bar{F} , \bar{W} y \bar{L} dan la misma solución. El segundo cuando número medio de trabajos siendo realmente procesados C_{max} es equivalente al $I_{máx}$ (desocupación media de las máquinas).
- **No equivalentes:** Medidas análogas a las anteriores pero para su valor máximo. Exceptuando: si la fecha de lanzamiento=0 (*released date*) para todos los trabajos y la fecha de vencimiento (*due date*) para todos los trabajos es la misma.

6. Reglas de despacho

En la literatura de secuenciación de tareas los términos reglas de despacho, reglas de secuenciación, regla de prioridad o heurísticas, son sinónimos. Especifican el orden en que deben realizarse los trabajos en cada centro. [27] Se pueden clasificar estas reglas en unas categorías como lo son:

6.1. Reglas de prioridad simple

Se basan en información relacionada con un trabajo específico (tiempo de procesamiento, fecha de vencimiento, número de operaciones, etcétera). En algunos casos, información como la cola, la longitud en la máquina donde el trabajo irá después, estas reglas se consideran lo suficientemente simples de modo que las reglas basadas en dicha información también se incluyen en esta categoría. Reglas como la selección aleatoria que no dependen de la información relacionada a un trabajo específico también se consideran simples.

6.1.1. Relacionadas con el tiempo de proceso

1. SI (“*Shortest imminent operation time*“): Seleccionar el trabajo con el menor tiempo de operación. También llamado (SIO, SPT)
2. SIS (“*Shortest gross imminent operation time*“): Seleccionar el trabajo con el menor tiempo de operación incluyendo *setup*.
3. LI (“*Largest imminent operation time*“): Seleccionar el trabajo con el mayor tiempo de operación. También llamado (LPT)
4. LIS (“*Largest gross imminent operation time*“): Seleccionar el trabajo con el mayor tiempo de operación incluyendo *setup*.
5. SR (“*Shortest remaining processing time*“): Seleccionar el trabajo con el tiempo de procesamiento restante más corto (para operaciones no realizadas).
6. LR (“*Largest remaining processing time*“): Seleccionar el trabajo con el tiempo de procesamiento restante más largo (para operaciones no realizadas).
7. LRM (“*Longest remaining processing time*“): Seleccionar el trabajo con el tiempo de procesamiento restante más largo (excluyendo la operación bajo consideración).
8. SIRIP: Seleccionar el trabajo con el tiempo de operación más corto donde el tiempo de procesamiento no se conoce de antemano y se determina mediante el uso del valor de error distribuido normalmente en la estimación.
9. LSPON (“*Longest subsequent operation*“): Seleccionar el trabajo con la operación posterior más larga.

6.1.2. Relacionadas con el tiempo de vencimiento del pedido

10. EDD: Seleccionar el trabajo con la fecha de vencimiento más próxima.
11. OPNDD: Seleccionar el trabajo con la fecha de vencimiento más temprana, se asignan fechas de vencimiento con el mismo espacio de tiempo entre cada operación cuando el trabajo entra al taller.

6.1.3. Relacionadas con el número de operaciones

12. FOPNR: Seleccionar el trabajo con menos operaciones restantes.
13. MOPNR: Seleccionar el trabajo con más operaciones restantes.
14. LHALF ("*Last half preference* "): Otorgue una mayor prioridad al trabajo con el cual menos de la mitad de todas las operaciones quedan por realizar.
15. FHALF ("*First half preference* "): Otorgue una mayor prioridad al trabajo con el cual más de la mitad de todas las operaciones quedan por realizar.

6.1.4. Relacionadas con los costos

16. Valor (Value): Seleccionar el trabajo con el mayor costo.
17. 1/C: Seleccionar el trabajo que tenga el costo de penalización por unidad más alto debido a la tardanza.

6.1.5. Relacionadas el tiempo de *Setup*

El tiempo de *setup* hace referencia al tiempo de alistamiento o ajuste de la máquina en una línea de producción.

18. NSUT: Seleccionar el trabajo que no requiere tiempo de *setup*.
19. MINSEQ: Seleccionar el trabajo o clase de trabajos con el tiempo de *setup* mínimo.
20. NB: Seleccionar el trabajo sin asignar que tenga el menor tiempo de *setup* relativo al trabajo recién completado.
21. NB': Utilizar la regla anterior (NB) pero comenzar con un trabajo distinto al primer trabajo en programar.
22. NB'': Aplicar la regla NB a la matriz de tiempos de *setup* después de restar el mínimo tiempo de *setup* a todos los valores de la columna de los tiempos de *setup*.

6.1.6. Relacionadas al tiempo de llegada y al azar

23. FIFO: Seleccionar el primer trabajo en llegar para ser el primero en salir.
24. Random: Seleccionar los trabajos al azar.
25. FASFO: Seleccionar el primer trabajo en llegar al taller para ser el primero en salir.
26. LIFO: Seleccionar el último trabajo para que salga primero del taller.
27. S-1: Seleccionar el trabajo con la menor holgura (tiempo disponible antes de la fecha de vencimiento para las demás operaciones).
28. S-2: Seleccionar el trabajo con la menor holgura “estática” (diferencia entre fecha de vencimiento y hora de llegada).
29. S-1/OP: Seleccionar el trabajo con la menor relación entre el tiempo de holgura y el número de operaciones restantes.
30. S-2/OP: Seleccionar el trabajo con la menor proporción de holgura a la cantidad de operaciones restantes.
31. JSR: Seleccionar el trabajo con la “holgura de trabajo mínima” (Relación de la holgura del trabajo al tiempo total disponible hasta la fecha de vencimiento).
32. RSPT1: Seleccionar el trabajo con la menor proporción de holgura “estática” al tiempo de procesamiento restante.

Nota: Se deben diferenciar los tipos de holgura (*Slack*) donde existe la **holgura estática** que se define como el tiempo disponible menos el de proceso, pero también se tiene la **holgura dinámica** que es la diferencia entre la fecha de entrega, el momento actual y el tiempo de proceso restante.

6.1.7. Relacionadas con la máquina

Estas reglas se pueden categorizar en la sección 3 pues tienen parecido a las heurísticas “mirar hacia adelante” (*look ahead*).

33. NINQ: Seleccionar el trabajo que pasará a su próxima operación donde la máquina tiene la cola más corta.
34. WINQ: Seleccionar el trabajo que pasará a su próxima operación donde la máquina tiene menos trabajo.

6.1.8. Información variada

35. ESD: Seleccionar el trabajo que tiene la fecha de inicio planificada más temprana para su operación.

6.2. Combinación de reglas de prioridad simple

En muchos casos estas reglas trabajan dividiendo la cola en dos o más grupos con distintas reglas de prioridad. En otros casos se aplican a la misma cola pero bajo diferentes circunstancias.

36. FIFO/SI: Seleccionar los trabajos que esperan más de un tiempo específico, usar la regla FIFO (23), si los trabajos en espera están en cola para una menor duración, seleccionar de acuerdo a la regla SI (1).
37. \$: Esta regla se encuentra definida en la literatura como el signo de pesos (“\$”). Dividir los trabajos en dos clases; alto y bajo valor en dólares. Seleccionar entonces el trabajo con un alto valor en dólares con FIFO, luego desde “otras reglas” (Categoría 3, seleccionar utilizando el parámetro p , que equivale a la proporción de trabajos en la clase de bajo valor en dólares).
38. SMOVE: Seleccionar el trabajo que pasará a la próxima operación donde la máquina tiene la cola “crítica” más corta (lo que significa menos tiempo de procesamiento). Si no hay colas críticas, utilizar la regla FIFO (Con un parámetro Q , que cambia el estado crítico de cualquier máquina).
39. ASIFIFO: Usar la regla SI (1) por cierto tiempo, luego la regla FIFO (23). Posteriormente repetir el ciclo.
40. SI + JSR: Si al menos un trabajo tiene una holgura negativa, la prioridad de todos los trabajos es igual a su valor de holgura y se usa la regla S-1 (27). Si todos los trabajos tienen holgura positiva, los trabajos que tienen una relación de holgura inferior, al doble de la relación mínima, obtienen la prioridad con la regla SI (1); otros trabajos en cola no son considerados.
41. 3CL-FIFO: Dividir los trabajos en tres clases de prioridad y usar la regla FIFO (23).
42. SEQ: Considerar el valor del trabajo en proceso, el tiempo de espera reducido y el número de operaciones (se puede modificar con diferentes proporciones).
43. OPNDDP: Seleccionar el trabajo con la regla SI (1) a menos que el trabajo esté detrás de su fecha de vencimiento.
44. DDNINQ: Los trabajos que salen del taller después de la operación o que van a una máquina con una cola pequeña ($< Q$) obtienen una mayor prioridad. Luego seleccionar según la regla DD (10).

45. FCFS*S: Utilizar la regla FIFO (23) si el número de trabajos en espera es menor que Q ; de lo contrario, usar la regla SI (1).
46. LOPN * S: Igual que la regla anterior (45), excepto que se debe usar la regla LI (3) en lugar de FIFO.
47. NINA*S: Seleccionar el trabajo que se unirá a la cola más corta, si todos los trabajos se unirán a las colas con más de Q trabajos, use la regla SI (1).
48. 2 CLASS: Dividir los trabajos en dos colas al azar de modo que, según en el tiempo requerido para la operación, cierto porcentaje de trabajos se una a la cola de alta prioridad; luego utilizar la regla FIFO (23).
49. 2C-SI: Formar dos colas de prioridad basadas en el tiempo de operación; luego usar la regla FIFO (23).
50. 2C-TSI: Esta regla prefiere un trabajo con holgura negativa; y luego aplicar la regla SI (1).
51. Costo/Tiempo (Cost/Time): Primero seleccionar cualquier trabajo “crítico” usando la regla SI (1), y luego seleccionar un trabajo que se atrasa (pero no críticamente tarde) con la mayor proporción entre el costo de la tardanza y el tiempo de operación; luego seleccionar un trabajo entre los restantes con la regla SI.
52. DDSU: Seleccionar los trabajos con la fecha de vencimiento más temprana; en caso de empate, seleccionar el trabajo que no requiere tiempo de *setup* (tiempo de configuración).
53. SI/Q: Utilizar la regla SI (1), pero si el trabajo seleccionado se unirá a la siguiente máquina con una cola de cierta longitud (o mayor), pruebe el siguiente trabajo en cola con SI (deje el trabajo seleccionado previamente en la cola).
54. SI/ Q_1Q_2 : Formar tres colas, una con trabajos yendo a una cola “corta” para su próxima operación. Otra con una cola “media”, y la tercera con una cola “larga”. Seleccionar con la regla SI con prioridad en la primera cola, luego la segunda y por último la tercera.
55. SOR: Utilizar la regla SI (1), pero dé preferencia a los trabajos que irán a las colas “críticas”. (Cola con solo una pequeña cantidad de trabajos esperando).
56. SI/S-1: Utilizar la regla SI (1) si una consecuencia inmediata no es la holgura negativa para trabajos en cola. De lo contrario, elegir el trabajo con el menor valor de tiempo de procesamiento de operación y holgura.

6.3. Índices ponderados de prioridad

Es la combinación de las dos categorías anteriores. Muchas investigaciones incluyen parametrización de estos pesos en un rango especificado.

Generalmente, en las reglas se elige el trabajo con el valor más pequeño del índice de prioridad.

57. P11: Igual que la regla S-1 (27) excepto que adjunta un peso α al tiempo disponible hasta la fecha de vencimiento.
58. P + S-1/OP: Seleccionar el trabajo con la suma ponderada más pequeña, del siguiente tiempo de procesamiento y el tiempo de holgura por operación restante.
59. P + WKR: Seleccionar el trabajo con la suma ponderada más pequeña del próximo tiempo de procesamiento y trabajo restante.
60. P/WKR: Seleccionar el trabajo con la relación ponderada más pequeña del siguiente tiempo de procesamiento para el trabajo restante.
61. P/TWK: Seleccionar el trabajo que tiene la proporción más pequeña entre el siguiente tiempo de procesamiento y el trabajo total.
62. P + WQ: Seleccionar el trabajo con la suma ponderada más pequeña del siguiente tiempo de procesamiento y trabajar en la próxima cola.
63. MSR: Seleccionar el trabajo con el valor más pequeño de fecha de vencimiento, sumando el valor ponderado de la siguiente operación y restando el valor ponderado del trabajo restante.
64. RPT/RT: Seleccionar el trabajo con la relación mínima entre el tiempo de procesamiento restante y la holgura de trabajo actual.
65. P1M: Seleccionar el trabajo con el valor más alto de relación entre la diferencia de hora actual y hora en que se planifica la operación, menos un factor basado en la asignación de flujo al valor ponderado del número de operaciones restantes.
66. P14: Es un índice ponderado basado en los parámetros de la regla P/WKR (60) y que también usa el parámetro de trabajo restante.
67. P/OPNR: Seleccionar el trabajo con la relación más baja entre el tiempo de procesamiento y el valor ponderado del número de operaciones restante.
68. P-SP: Seleccionar el trabajo con el valor más bajo de la diferencia entre el tiempo de procesamiento ponderado de la operación y el valor ponderado para la siguiente operación.
69. P + NINQ: Seleccionar el trabajo con el valor más pequeño de la suma ponderada de la operación y el valor ponderado del número de trabajos en la siguiente cola.

70. $(P + WQ)/P$: Seleccionar el trabajo con el índice más pequeño calculado como la suma de los valores ponderados del tiempo de funcionamiento y el trabajo en la próxima cola, dividido por el tiempo de procesamiento ponderado de la siguiente operación.
71. SHOPNE: Similar a la regla SI (1), excepto que el tiempo para la operación se estima a partir del valor promedio por una función ponderada.
72. PI6: Es un índice basado en la fecha de vencimiento, el tiempo de transporte, el tiempo de procesamiento ponderado para las operaciones restantes y el tiempo ponderado para la operación inminente.
73. PI7: Es un índice basado en el tiempo disponible, la media y la varianza del tiempo de procesamiento para la operación inminente.
74. PI8: Es un índice basado en la fecha de vencimiento, el tiempo de transporte y el valor ponderado del tiempo de procesamiento restante.
75. PI11: Es un índice que selecciona el trabajo con el valor más alto de penalización esperada (los trabajos que terminan temprano tienen penalidad negativa o bonificación).
76. PI12: Es un índice basado en PI6 (72) y el valor modificado del índice en PI6 después de que se haya procesado el trabajo.
77. PI13: (Para procesadores paralelos). Es un Índice basado en los requisitos de producción en un período determinado, tiempo de procesamiento y costo.
78. PI14: Es un índice basado en el costo de inventario en proceso, costo de procesamiento, costo de retraso, costo y tiempo de instalación, y prioridad externa.

6.4. Reglas de programación heurística

Estas reglas implican una consideración más compleja como la carga de la máquina anticipada, el efecto de la alternativa enrutamiento, programación de operación alternativa, etc. Estas reglas generalmente se usan junto con las reglas en la Categoría I. En algunos casos, una regla heurística puede involucrar aspectos no matemáticos de la inteligencia humana.

79. Operación anterna (*Alternate Operation*): Si la selección de un trabajo de acuerdo con alguna regla simple hace que otro trabajo sea “crítico” (holgura negativa), vea el efecto del trabajo ya seleccionado. Repita, si algún otro trabajo se ve afectado.
80. Ruta alterna (*Alternate Routing*): Estudie el efecto del enrutamiento de un trabajo a través de otro conjunto (predeterminado) de instalaciones alternativas de enrutamiento.

- a) Enrutamiento alternativo con SI (1)
 - b) Enrutamiento alternativo con LI (3)
 - c) Enrutamiento alternativo con Aleatorio (24).
81. “Mirar hacia adelante” (*Look Ahead*): Estudia el efecto de programar un trabajo (determinado por una regla simple) en otro trabajo que pueda llegar a la cola antes de que se complete el trabajo programado. (Todas las reglas simples consideran solo trabajos que ya están en cola).
 82. P*S-1: Aplicar la regla SI (1), excepto cuando el trabajo está en su operación final y se puede evitar el retraso tomando este trabajo fuera de servicio.
 83. SHOPNH: Seleccionar el trabajo con la regla SI (1), pero espere si hay algunos trabajos en la cola y pronto se espera otro trabajo con menor tiempo de procesamiento. (Mantenga la máquina inactiva hasta que este trabajo llegue).
 84. DSIJ: Usar S-1 (27) para los trabajos en una cola y los que se unirán a esta cola específica de sus operaciones actuales.
 85. Insertar (*Insert*): Usar la regla “Look Ahead”,(81); si se observa un intervalo de tiempo inactivo en la siguiente máquina, inserte otro trabajo de la cola correspondiente que pueda completarse antes de que el trabajo crítico llegue a la máquina.
 86. Subconjunto (*Subset*): Determinar los trabajos críticos, programar estos primero (en su totalidad), y luego programar otros trabajos alrededor de los trabajos críticos.
 87. Rehacer (*Re-do*): Si se descubre que uno o más trabajos llegan tarde a algún horario, se vuelve a hacer la programación con las fechas de vencimiento ajustadas para un horario mejorado.
 88. Flexibilidad (*Flexibility*): Similar al “*Insert*” (85), excepto que se hace después de que el diagrama de Gantt esté dispuesto para así realizar mejoras.
 89. Manipulación (*Manipulation*): Se intenta mejorar el cronograma manipulando diferentes operaciones mientras se traza el diagrama de Gantt.
 90. Programa que trasciende el tiempo (*Time Transcending Schedule*): Determinar la calificación de prioridad para cada trabajo. Programar la siguiente operación del trabajo con la máxima prioridad. Volver a evaluar la prioridad y repetir, siempre programando la siguiente operación del trabajo con la máxima prioridad.

6.5. Otras reglas

Estas pueden involucrar reglas diseñadas para un almacén específico, combinan índices de prioridad basados en funciones matemáticas de parámetros de trabajo, o aquellas reglas no categorizadas anteriormente.

91. S-3: Esta regla es igual que S-1 (27), excepto que el tiempo de espera anticipado para todas las operaciones se resta de la holgura de cada trabajo.
92. S-3/OP: Igual que la regla S-1/OP (29), excepto que se resta el tiempo de espera anticipado del valor de holgura (numerador).
93. MJSR: Esta regla es Similar a la JSR (31), excepto que el tiempo de retardo esperado se agrega a cada tiempo de operación.
94. RSPT2: Igual que la regla RSPT1 (32), excepto que se debe restar el tiempo de espera anticipado para todas las operaciones de un trabajo al valor de holgura.
95. RSPWTI: Igual que la regla RSPT1 (32), excepto que se debe agregar el tiempo de espera anticipado al denominador.
96. RSPWT2: Igual que la regla RSPT1 (32), excepto que se debe agregar el tiempo de espera anticipado para todas las operaciones de un trabajo al denominador y restar la misma cantidad del numerador.
97. RSWT1: Seleccionar el trabajo que tiene la relación más pequeña entre el tiempo de holgura del trabajo y el tiempo de espera futuro “estimado”.
98. RSWT2: Igual que la regla RSWT1 (97), excepto que se debe restar el tiempo de espera futuro al tiempo de holgura.
99. RSMWT: Seleccionar el trabajo con la menor proporción entre el tiempo de holgura del trabajo y el tiempo de desplazamiento estándar, más el tiempo de espera “estimado”.
100. XWINQ: Seleccionar el trabajo que continuará su próxima operación en la cola con el menor trabajo, tanto presente como esperado.
101. P + XWQ: Seleccionar el trabajo que tiene la suma ponderada más pequeña del tiempo de procesamiento de la operación y el trabajo esperado en la siguiente cola.
102. PI9: Es un índice basado en la fecha de vencimiento, suma ponderada del tiempo de procesamiento para la operación restante, suma ponderada de la espera “estimadoz el valor ponderado del tiempo de procesamiento de la siguiente operación.

103. **PI10:** Es un índice que selecciona el trabajo con la relación más pequeña entre el tiempo disponible hasta la fecha de vencimiento y el tiempo restante (que incluye los tiempos de procesamiento más el tiempo de espera “estimado”).

Las siguientes 4 reglas se usaron para un taller especial donde se puede iniciar un trabajo simultáneamente en todas las máquinas. Por conveniencia, las reglas se establecen como las recíprocas de la real con el valor mínimo cambiado al valor máximo.

104. **PI14:** Seleccionar el trabajo con el valor máximo de costo de penalización por unidad del producto y la diferencia entre el tiempo de finalización de todos los trabajos y la fecha de vencimiento del trabajo.
105. **PI15:** Igual que la regla PI14 (104), sólo que el producto está dividido por la carga de trabajo total de la máquina.
106. **PI16:** Igual que la regla PI14 (104), sólo que el producto está dividido por el tiempo máximo de procesamiento entre todas las máquinas.
107. **PI17:** Igual que la regla PI14 (104), sólo que el producto está dividido por el tiempo mínimo de procesamiento entre todas las máquinas.
108. **OSF** (Factor de holgura de operación): Es un índice basado en la fecha de vencimiento de la operación, el tiempo de procesamiento de la operación y el tiempo de llegada del trabajo.
109. **PTF** (Factor de tiempo de procesamiento): Es un índice basado en el producto ponderado del tiempo de procesamiento del i -ésimo trabajo y la suma de los tiempos de procesamiento de trabajos en ciertos grupos de máquinas.
110. **OUF** (Factor de urgencia operativa): Es un índice basado en una función de fecha de vencimiento, número de operaciones y número de grados de todas las operaciones en un cierto nivel.
111. **PCF** (Factor de restricción precedente): Es un índice basado en las restricciones de precedencia de un trabajo.
112. **SI/SI(F)**: Modificar el valor de holgura restando un parámetro de control para permitir las demoras en el sistema. Programar los trabajos con un valor de holgura modificado menor que cero primero, con la regla SI. Luego programar otros trabajos con la regla SI (1).
113. **F/SI(F)**: Calcular la holgura modificada como en la regla anterior (112). Luego programar el trabajo con el valor de holgura menos modificado, si es menor que cero; de lo contrario, programar según la regla SI (1).

Las reglas de despacho fueron extraídas del documento (*A survey of scheduling rules, Panwalkar, 1977*) [27]

7. Marco teórico

A lo largo de las décadas la secuenciación de tareas ha llevado a diversos estudios y métodos para la solución de esta problemática. Surgen en los años 50 los primeros métodos para organizar y planear la guerra [5] que posteriormente nacen como una rama de la administración llamada investigación de operaciones. La investigación de operaciones (IO) hace uso de los modelos matemáticos, estadística y algoritmos para llegar a la toma de una decisión.

El problema de *scheduling* ha sido resuelto usando técnicas exactas y técnicas aproximadas.[28] En la **figura 8** se muestran las diferentes alternativas de solución [29]

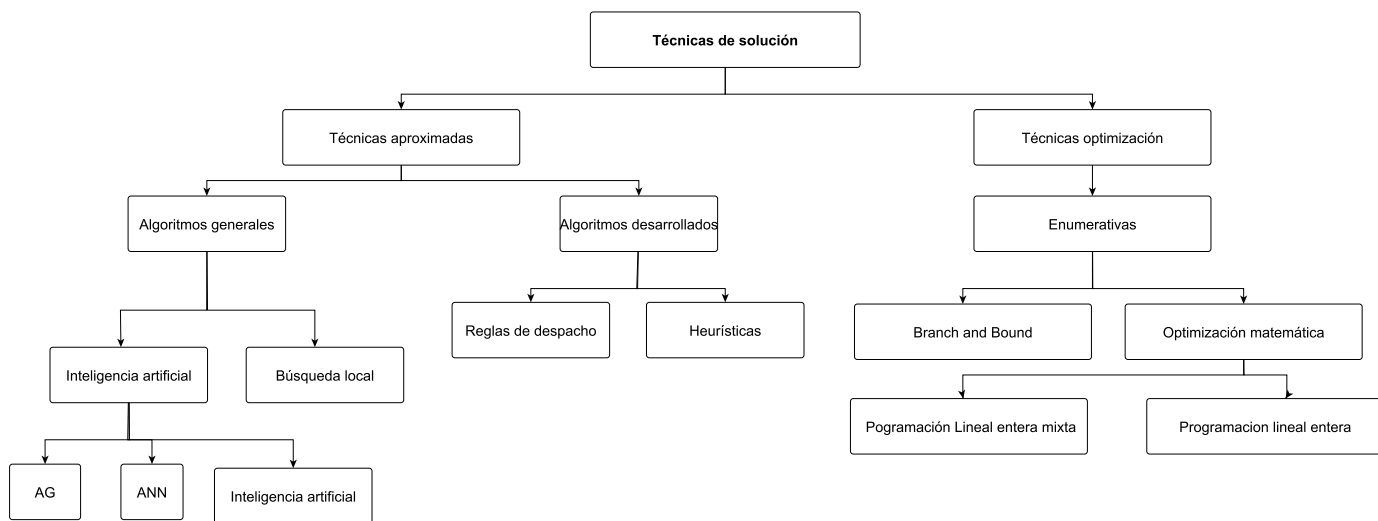


Figura 8: Técnicas de solución para secuenciación de tareas

La problemática de secuenciación de tareas ha sido investigada a lo largo de los años y se han planteado diversas alternativas para buscar una manera de solucionarla. [30]

Henry Gantt fue uno de los pioneros en el área pues contribuyó con ideas que brotaron en su famoso diagrama de Gantt creado en 1910, siendo una herramienta importante de gestión que proporciona un calendario gráfico para la planificación y control del trabajo. “El más pronto y mejor organizado tipo de diagrama de control para mostrar las relaciones entre lo planificado y lo realmente acontecido.” [4].

7.1. Tipos de Diagrama de Gantt

Se tienen dos tipos de este diagrama:

1. **Orientado a máquinas:** Este Diagrama de Gantt de la **figura 9** está orientado a máquinas, se muestra un grupo de 3 máquinas y 4 trabajos. En este relaciona cada máquina con los trabajos que se realizarán en ella en un tiempo determinado.

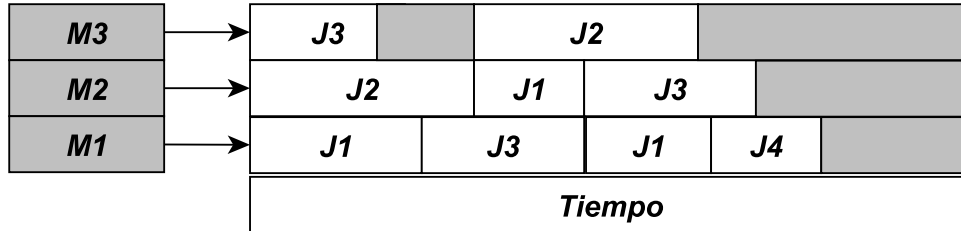


Figura 9: Gantt orientado a máquinas

2. **Orientado a trabajos:** Este Diagrama de Gantt de la **figura 10** está orientado a trabajos, se muestra el mismo grupo de 3 máquinas y 4 trabajos, pero en este caso se relaciona cada trabajo con las máquinas que están ligadas a él en un tiempo determinado.

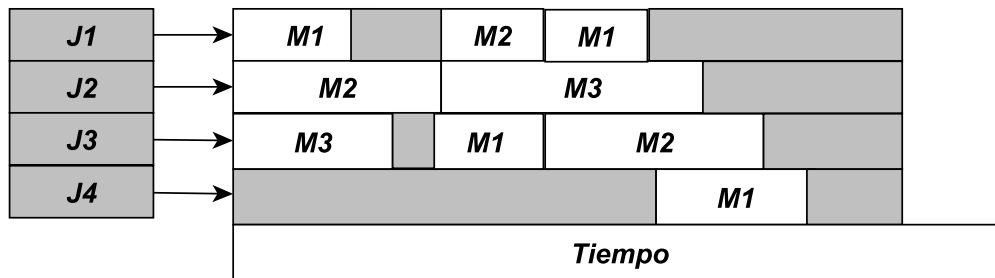


Figura 10: Gantt orientado a trabajos

Posteriormente en el momento de la creación de la teoría de complejidad computacional se cataloga el problema de secuenciación de tareas como un problema NP-Difícil (NP: tiempo polinomial no determinista, con mayor grado de complejidad) Se dice que un problema es NP-difícil cuando se demuestra que cualquier algoritmo de solución tiene un tiempo de ejecución que aumenta, en el peor de los casos, exponencialmente con el tamaño del problema. [31]. Entonces si n son nuestras variables de decisión y la manera de calcular el tiempo de resolución del modelo es $25^2 + n$, al ser una expresión polinómica sería factible

su resolución en un tiempo polinómico [32]. Para estos problemas complejos se empezaron a desarrollar softwares a mediados de los años 80 con todas las alternativas de solución.

Los recientes desarrollos en la teoría de la programación se han centrado en extender los modelos para incluir más restricciones prácticas. Además, debido a los estudios de complejidad realizados durante las últimas dos décadas, ahora se entiende ampliamente que la mayoría de los problemas prácticos son NP debido a que el espacio de soluciones crece exponencialmente con el incremento del número de tareas (o trabajos) a ser consideradas.[33].

En este proyecto de investigación se busca realizar una revisión del estado del arte para identificar líneas promisorias de investigación en el campo teórico y práctico debido a la importancia y gran cantidad de aplicaciones de este tipo de problema.

Para los problemas de programación secuencial se debe tener en cuenta unos elementos esenciales [13]:

1. **Ambiente de producción:** (Taller: Espacio donde se encuentran los procesos productivos)
 - Recursos.
 - Layout (Distribución de planta).
2. **Restricciones:** Asociadas a procesos, operaciones, formas de almacenamiento y medios de transporte.
3. **Variables del problema:** Orden en el que ingresan los trabajos a las máquinas, restricciones de precedencia.
4. **Medidas o indicadores de desempeño:** Para nuestro caso existen 2:
 - Indicadores relacionados con los trabajos.
 - Indicadores relacionados con las máquinas.

Las reglas de despacho son el parámetro principal en el cual se centra el trabajo de investigación. En las últimas dos décadas, los investigadores en el campo de la secuenciación y la programación han analizado varias reglas de despacho de prioridad a través de técnicas de simulación. [27].

En el presente proyecto se han enunciado alrededor de 100 reglas de despacho de prioridad.

Para analizar las reglas de despacho se utilizó un software académico libre llamado LE-KIN el cual resuelve 8 reglas de despacho y también heurísticas, en este caso en el software creado por Pinedo [8] se encuentran las alternativas de solución usadas comúnmente.

La planificación de la capacidad transforma los requisitos de fabricación, tal como se establece en la etapa MRP (Planificación de los requerimientos de material), en un plan detallado de carga de la máquina, para cada una o para cada grupo de ellas en la planta. Es una tarea de programación y secuenciación [34].

En la **figura 11** se aprecia que la secuenciación en LEKIN puede ser seleccionada por reglas y en este caso se tiene ATCS, EDD, MS, FCFS, LPT, SPT, WSPT, CR.

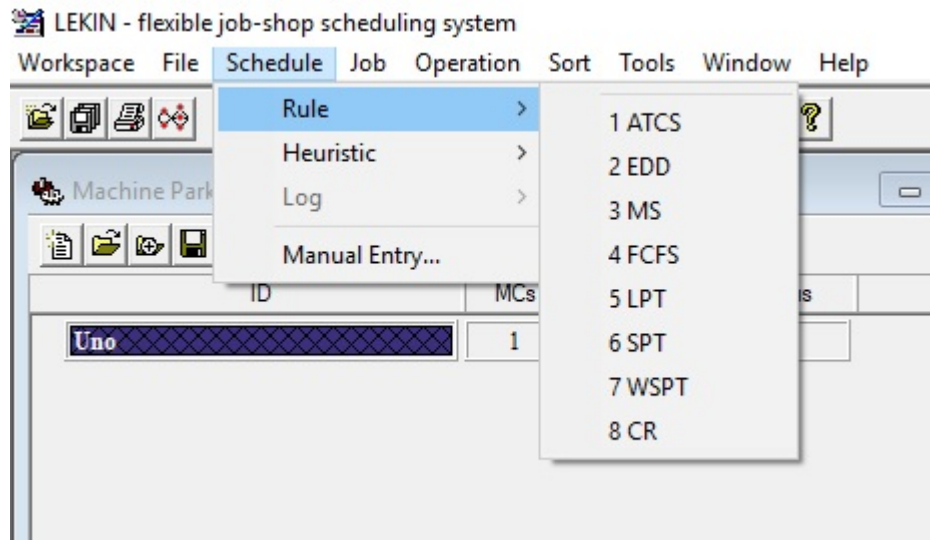


Figura 11: Reglas de despacho LEKIN

En la **figura 12** se tienen las heurísticas que trabaja LEKIN estando disponibles dependiendo del modelo inicial planteado al arrancar el programa, en algunos casos habrá heurísticas que no estarán disponibles para nuestro modelo por la información cargada inicialmente.

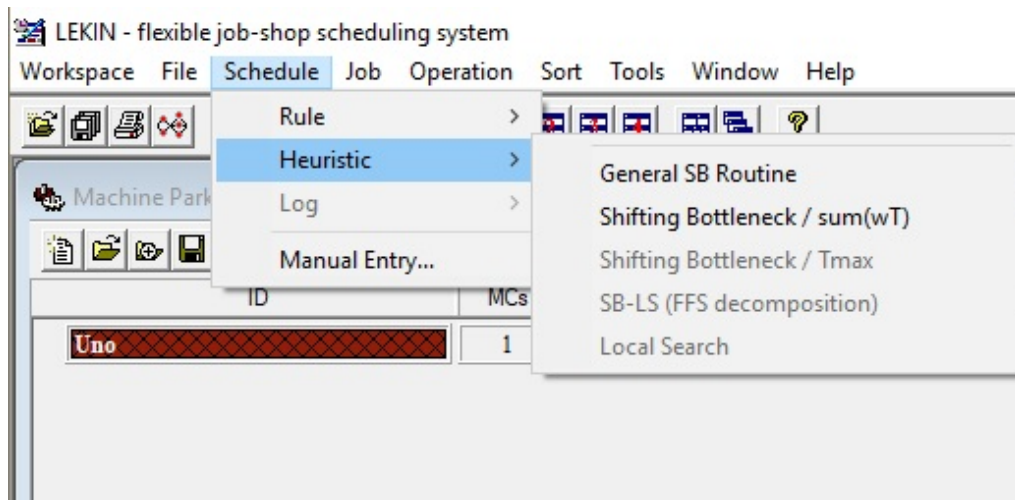


Figura 12: Heurísticas LEKIN

7.2. Modelo matemático del problema

Desde hace décadas la problemática de secuenciación de tareas ha generado conflictos entre los investigadores, los modelos matemáticos para describir los problemas son complejos y largos. A continuación se explicará el modelo para 2 y 3 trabajos. Usando programación lineal entera mixta.

En un problema lineal entero mixto (PLEM) se tienen variables de naturaleza entera y binaria en la **figura 13** se muestra un espacio de soluciones que se plantearía así

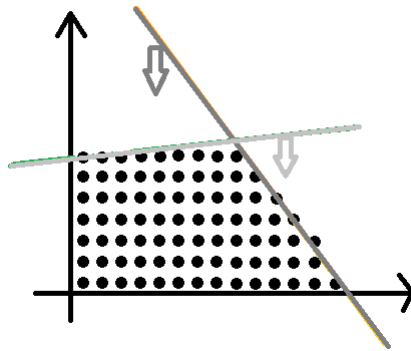


Figura 13: Gráfica del problema PLEM

Para poder darle naturaleza continua a las variables se deben relajar las restricciones para poder plantear el modelo como se muestra en la **figura 14**

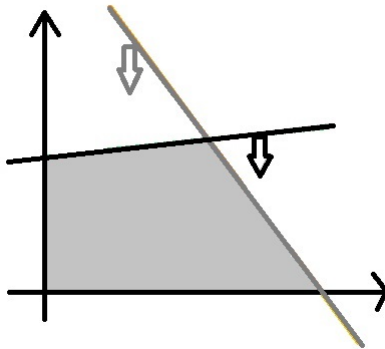


Figura 14: Gráfica del problema PL

Para poder transformar el modelo y volverlo lineal se deben relajar las restricciones teniendo en cuenta el concepto de ***“Restricciones disyuntivas”***. Donde se busca mediante un proceso deshabilitar una de las restricciones para acotar el espacio de soluciones como se muestra en la **figura 15 y 16**

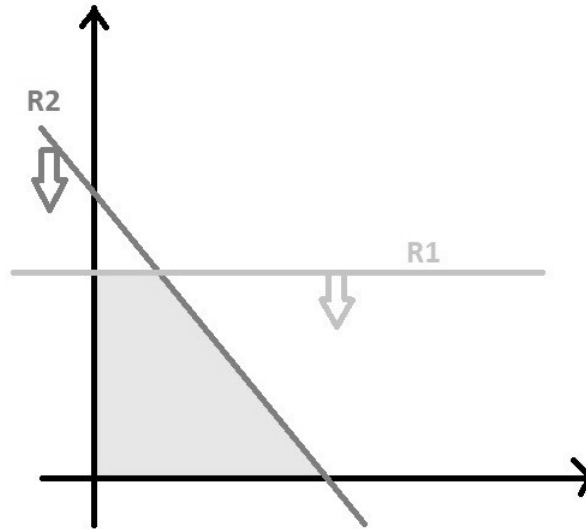


Figura 15: Restricción disyuntiva 1

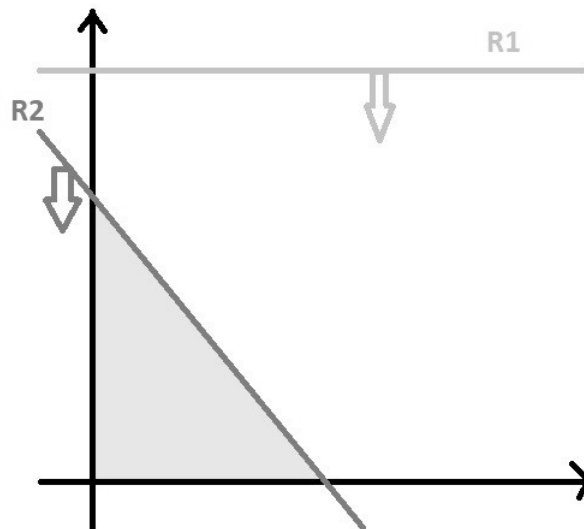


Figura 16: Restricción disyuntiva 2

Por lo tanto el modelo original será de la siguiente manera:

$$\begin{aligned}(R_1) \quad & a_i x \leq b_i \\ (R_2) \quad & a_j x \leq b_j\end{aligned}$$

En este tipo de restricciones para realizar el cambio necesario se debe hacer uso de una variable binaria llamada y_i Por lo tanto nuestro nuevo paquete de restricciones será:

$$\begin{aligned}(R_1) \quad & a_i x \leq b_i + M y_i \\ (R_2) \quad & a_j x \leq b_j - M(1 - y_i)\end{aligned}$$

El problema de secuenciación lleva a un problema de planeación abarcando los campos de investigación de operaciones y producción. En este proyecto se busca revisar este problema teniendo en cuenta las reglas de despacho y las medidas de desempeño tanto de trabajos como de máquinas, buscando así cumplir con una función multiobjetivo que pretende minimizar una serie de tiempos lo cual directamente minimiza los costos. Se evidencia que la mayoría de los problemas en la naturaleza tienen varios (posiblemente en conflicto) objetivos a cumplir. [35]

7.3. Heurísticas y algoritmos

Teniendo en cuenta las técnicas de solución se tienen los algoritmos que se dividen en heurísticas y reglas de despacho. Algunos referentes en la literatura son:

7.3.1. Algoritmo de Moore

Genera secuencias óptimas para problemas de n trabajos en una sola máquina [36] Busca minimizar el número de trabajos con retraso. Consiste en los siguientes pasos:

1. Ordenar los trabajos de acuerdo a la regla EDD, obteniendo la solución inicial
2. Seleccionar el primer trabajo i atrasado. (De no haber se prosigue con el paso 4)
3. Considerar los trabajos 1 al i . Rechazando el trabajo con mayor tiempo de proceso, volver al Paso 2
4. Formar la secuencia que resulta de tomar la secuencia actual y colocar todos los trabajos rechazados al final

7.3.2. Algoritmo de Lawler

Considera las restricciones de precedencia de los trabajos. Busca minimizar el retraso máximo. [36] Se realiza de la siguiente manera:

1. Se programa el último trabajo. Mirando los trabajos que no tienen precedencia. Y se comparan entre ellos hasta encontrar el que minimice el retraso. $L_i = F_i - d_i$
2. Se programa el penúltimo trabajo. Mirando los trabajos que no tienen precedencia. Y se comparan entre ellos hasta encontrar el que minimice el retraso. $L_i = F_i - d_i$
3. Se programa el antepenúltimo trabajo con los mismos criterios anteriores.
4. Se programa el siguiente trabajo con los criterios anteriores y así sucesivamente hasta programar todos los trabajos.

7.3.3. Giffler y Thomson

Los algoritmos de Giffler y Thompson se pueden considerar como la base de todas las heurísticas basadas en reglas de prioridad. El procedimiento de generación de GT explora el espacio de búsqueda por medio de una estructura de árbol. Los nodos en el árbol representan los *schedules* (secuencias) parciales, los arcos representan las posibles elecciones, y las hojas del árbol son el conjunto de *schedules*. [24]

7.3.4. Algoritmo de Johnson

Llamado también regla de Johnson, busca minimizar el tiempo en tránsito desde el comienzo del primer trabajo hasta la culminación del último. Los pasos son:

1. Anotar el tiempo de operación de cada trabajo en ambas máquinas
2. Elegir el tiempo más breve
3. si el tiempo más breve es de la primera máquina, se realiza el primer trabajo; si es para la segunda máquina se realiza el último trabajo. (Nota: De haber empate se realiza en la primera máquina)
4. Repetir los pasos anteriores hasta completar la programación

7.3.5. Heurística de cambios de cuello de botella

Es una heurística que busca minimizar la tardanza total ponderada en job shop [37]. Esta heurística se utiliza para un número finito de máquinas y trabajos.

El procedimiento usado para la solución del problema encontrando las secuencias fue:

1. Realizar el grafo disyuntivo
2. Hallar la ruta crítica del problema mediante el CPM (Critical Path Method)
3. Identificar los nodos con holgura 0
4. Seleccionar la máquina sin holgura y analizar los trabajos que entran en ella (El tiempo de proceso, el tiempo en que está disponible el trabajo y la fecha de entrega). La fecha de entrega se halla mediante los cálculos de la ruta crítica
5. Analizar diferentes secuencias de trabajos en la máquina de holgura 0 hasta encontrar: primero el TTC (*Time to completion*) el cual define el tiempo que demora un trabajo específico en ser completada, luego se encuentra la DD (*Due Date*) o sea la fecha de vencimiento o de entrega del trabajo
6. Hallar la diferencia entre el TTC y la DD para encontrar las tardanzas
7. Identificar las máximas tardanzas en cada una de las secuencias y escoger la mínima tardanza

7.3.6. Algoritmo de Page/Palmer

Page usó métodos de clasificación por computadora que implicaban el intercambio de trabajos individuales, el intercambio de grupos de trabajos, el emparejamiento y la fusión de trabajos. En el caso de intercambio, comenzando con una secuencia de trabajo específica, cada par sucesivo de trabajos adyacentes se prueba para ver si tal cambio reducirá la duración del calendario. Si se logra una mejora, entonces se repite el proceso. El mismo proceso se aplica al intercambio de cadenas de trabajos en lugar de trabajos individuales. El emparejamiento y la combinación de cadenas se basa en reagrupar cada par sucesivo de cadenas de trabajo en una nueva cadena ordenada de trabajos y probar una mejora en la vida útil. En esencia, el método implica una forma de enumeración restringida. [38]

7.3.7. Método de Gupta

Gupta diseñó un método el cual es similar al de Palmer, excepto que él define los índices en un manera diferente, tomando en cuenta unos interesantes hechos de la optimización de la regla de Johnson para el problema de tres máquinas.

El índice S_j para el trabajo j se calcula como se muestra en la figura 4:

$$S_j = \frac{e_j}{\min_{t \leq k \leq m-1} \{t_{k,j} + t_{k+1,j}\}} \quad (4)$$

$$e_j = \{ -1, t_{1,j} < t_{m,j} + 1, t_{1,j} \geq t_{m,j} \}$$

Después de esto, los trabajos son secuenciados de acuerdo al índice, de menor a mayor. Si hay un empate se programa primero el trabajo con menor suma total de tiempos de proceso. [39]

7.3.8. Heurística CDS

Hace uso de una múltiple aplicación del algoritmo de Johnson para dos máquinas para tratar de obtener una buena programación. Produce soluciones aproximadas de n trabajos en m máquinas. Se pueden encontrar hasta $m-1$ secuencias. [40]

7.3.9. Heurística NEH

Es un procedimiento constructivo de secuenciación muy eficiente que proporciona buenas secuencias para el problema del flow shop con el objetivo de minimizar el instante máximo de finalización de los trabajos (C_{max}). Como es bien conocido, este procedimiento se puede resumir en dos fases: en la primera los trabajos se ordenan según el criterio LPT y en la segunda, conocida como fase de inserción, se construye progresivamente la secuencia, incorporando cada trabajo, de uno en uno, en el orden previamente establecido, colocándolo en la mejor posición posible de la secuencia parcial generada, con el fin de minimizar el (C_{max}) parcial. Una vez colocado un trabajo, la posición relativa del mismo respecto a los demás trabajos ya considerados no se altera. [41]

7.3.10. Método de ruta crítica CPM (*Critical Path Method*)

Es un algoritmo basado en la teoría de redes diseñado para facilitar la programación de un conjunto de actividades. Es una herramienta importante para la gestión efectiva de proyectos. [42]

7.3.11. Algoritmo de Johnson

Tiene como objetivo minimizar el makespan, tradicionalmente en la literatura se considera este algoritmo para problemas de 2 máquinas pero puede ser adaptado para m máquinas. Se realiza de la siguiente manera:

1. Se anota el tiempo de operación de cada trabajo en las máquinas
2. Se selecciona el tiempo más corto.
3. Si el tiempo más corto es para la primera máquina, se hace el primer trabajo, de lo contrario se realizará el último. En caso de que ambos tengan el mismo tiempo se realizará el primero.
4. Repetir los dos pasos anteriores hasta completar toda la programación.

[43]

7.3.12. Branch and Bound (Secuenciación)

El algoritmo Branch and Bound (B&B) es un método exacto para encontrar una solución óptima para un problema NP-difícil. Es una técnica enumerativa que se puede aplicar a una amplia clase de problemas de optimización combinatorial. [3]

En el **cuadro 1** se presenta un ejemplo aplicado a un caso Flow Shop

Trabajo	Tiempo de procesamiento P_j	Fecha de entrega d_j
A	6	8
B	1	4
C	3	3
D	7	9

Cuadro 1: Branch & Bound aplicado a flow shop

Una secuencia corresponde a la asignación de un orden para cada uno de los trabajos. Para poder realizar el B&B se deber llevar a cabo los siguientes pasos:

- **Inicializar:** En este punto no se tiene ninguna secuencia definida por lo tanto se ponen (*, *, *, *) lo cual indica que no hay trabajos asignados a alguna posición.

- **Paso 1:** Se debe construir la secuencia con todas las posibles combinaciones, iniciando con la última posición. (*, *, *, A) (*, *, *, B) (*, *, *, C) (*, *, *, D)
- **Paso 2:** Se asignar el segundo trabajo en la secuencia, se deriva de cada uno de estos cuatro nodos a tres posibilidades: bifurcar desde (*, *, *, A) a (*, *, B, A), (*, *, C, A) y (*, *, D, A); La ramificación de (*, *, *, B) da tres posibilidades (*, *, A, B), (*, *, C, B), y (*, *, D, B), así para las demás posibilidades.
- **Paso 3:** Asignar el trabajo de la segunda posición genera automáticamente toda la secuencia.

Este proceso está representado por un árbol ramificado. Cada nodo corresponde a una secuencia parcial con varios trabajos asignados a las últimas posiciones y los trabajos restantes sin asignar. En la **figura 17** se ve el planteamiento para 4 trabajos.

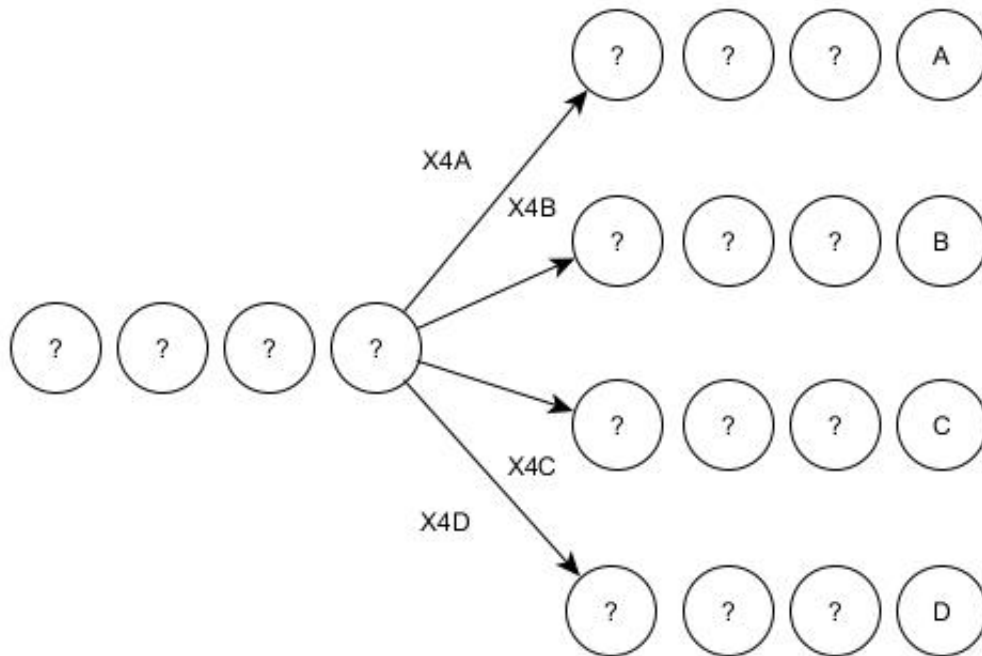


Figura 17: Branch & Bound para flow shop (1)

Se tienen los siguientes tiempos, que se obtienen hallando el valor para cada t como se muestra en la ecuación 5:

$$t = P - d_j \quad (5)$$

Donde P es la suma de los tiempos de cada trabajo y d_i es el la fecha de entrega del trabajo i . El valor de P será la suma de todos los tiempos de procesamiento en este caso $P = 6 + 1 + 3 + 7 = 17$.

Se comienza hallando el valor de $t_a = 17 - 8 = 9$ y así sucesivamente para cada uno de los trabajos.

- (?,?,?,9) $t_A \geq 9$
- (?,?,?,13) $t_b \geq 13$
- (?,?,?,14) $t_c \geq 14$
- (?,?,?,14)
- (?,?,?,8) $t_d \geq 8$

Por lo cual después de hallar todos los valores de t se toma el valor mínimo de estos tiempos, siendo D el trabajo que va al final de la secuencia.

Al quedar D como el trabajo final de la secuencia se ramifican las otras 3 posiciones como se muestra en la **figura 18**

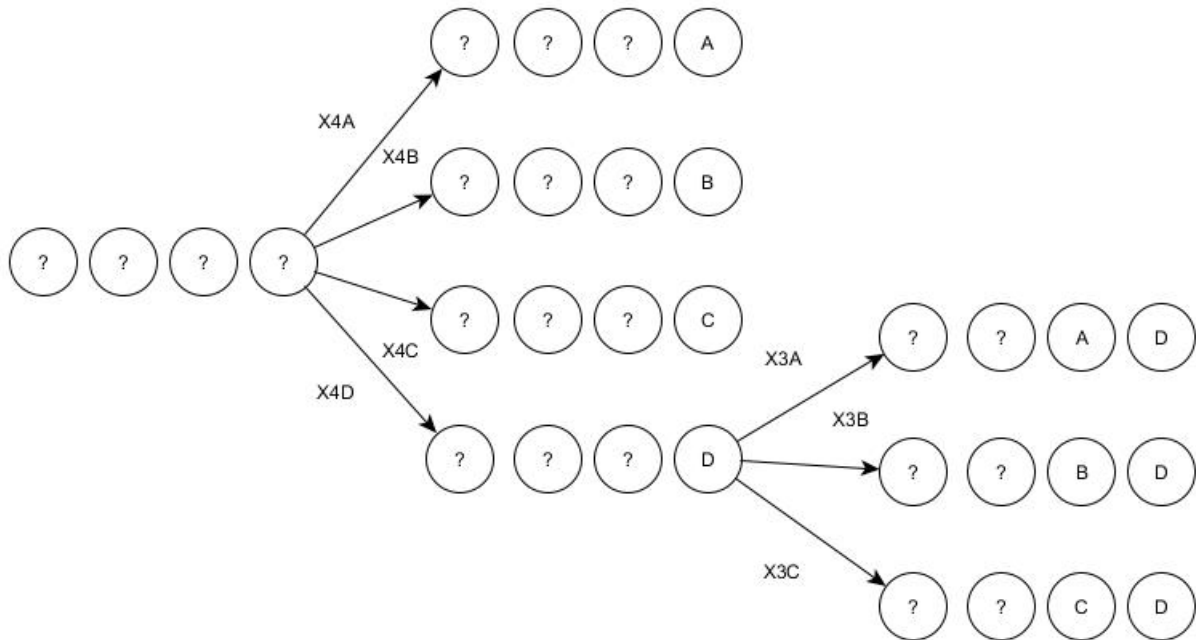


Figura 18: Branch & Bound para flow shop (2)

Ahora se hallan los nuevos tiempos, se excluye el tiempo del trabajo D que ya se asignó y se halla el valor de $P = P_a + P_b + P_c$ para poder reemplazar en la ecuación de t .

$P=6 + 1 + 3 =10$, de esta manera hallo el valor de t para cada trabajo y le sumo el valor de 8 hallado en la ramificación anterior. Por lo tanto los nuevos tiempos serán:

$t_a = 10 - 8 = 2$ por lo tanto sumando el 8 de la ramificación se obtiene un tiempo para $t_a=10$. Así se procede con los demás trabajos restando la fecha de entrega de cada uno al valor de P .

$$t_a \geq 10$$

(* , * , 2 , 8)

$$t_b \geq 14$$

(* , * , 6 , 8)

$$t_c \geq 15$$

(* , * , 7 , 8)

Se retoma la secuencia desde AD al ser A el menor tiempo y se consideran las siguientes posibilidades como se muestra en la **figura 19** .

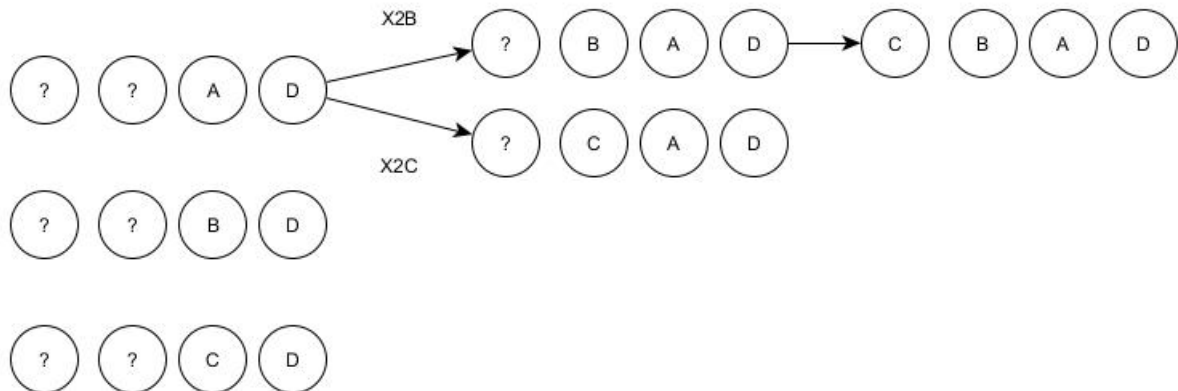


Figura 19: Branch & Bound para flow shop (3)

Para los siguientes tiempos se tiene en cuenta que el nuevo valor de P en este caso 4 y además de eso se suman los valores de los trabajos ya asignados 8 y 2. Realizando lo siguiente se obtiene:

$$t_b \geq 10$$

(* , 0 , 2 , 8)

$$t_c \geq 11$$

(* , 1 , 2 , 8)

Se programa el segundo trabajo y esto genera la secuencia completa, que para este caso será **CBAD**, pues el menor tiempo para la segunda posición es el de B, dejando inmediatamente a C en la primera posición.

A continuación se presenta un cuadro resumen de la literatura sobre el problema de Flow Shop no permutacional. La notación se define de la siguiente manera:

Para el campo β :

- rc: Restricción de recursos
- Saltar: Saltar operaciones
- Disponibilidad: Condiciones de disponibilidad de la máquina
- fmls: Grupo de familias de producto
- Aprender: Efecto de aprendizaje
- hr: Recursos heterogéneos
- rp: Relocalización

Una entrada del campo β en mayúsculas significa un parámetro estocástico.

En la columna de comentarios:

- B&B: Branch and Bound.
- SA: Algoritmo de recocido simulado
- MPF: Formulación de programación matemática
- SS: Búsqueda dispersa
- PR: Vinculación de ruta
- TS: Búsqueda tabú
- OM: Otras meta-heurísticas
- GA: Algoritmo genético
- ACO: Optimización de colonia de hormigas
- IG: Iteración *Greedy* (Codiciosa)
- CLP: Programación lógica de restricciones
- CCP: Programación restringida por azar
- FGP: Programación de objetivos difusos
- PFS: Flow Shop permutacional
- NPFS: Flow Shop no permutacional

A continuación en el **cuadro 2** se presenta un resumen de la literatura teniendo en cuenta el problema NPFS y sus posibles soluciones dependiendo de su planteamiento.

Cuadro 2: Resumen de la literatura NPFS

Referencia	Problema	Comentarios
[44]	F rc Cmax	Algoritmo B&B
[45]	F Cmax	Bound entre NPFS Cmax y PFS Cmax para estancias especiales
[46]	F Cmax	Enumerativo para pequeñas instancias y grandes instancias SA
[47]	F2 S_{ijk} , tiempos de eliminación — Cmax	PFS no es óptimo y el problema es NP-Difícil
	F2 Bloqueo Cmax	PFS no es óptimo y el problema es NP-Difícil
[48]	F b_i Cmax	Heurística para equilibrar el uso de recursos
[49]	F Lote Costos	Algoritmo B&B
[50]	F Lote, espera finita Cmax	Procedimiento recurrido a medida
[51]	F Cmax	Heurística HFC
[52]	F Tiempos de retraso Cmax	MPF
[53]	F Saltar $\sum C_j$	Reglas de despacho y heurísticas
[54]	F Cmax	Meta-heurísticas, basadas en SS y PR, y TS
	F Cmax	OM
[55]	F Saltar Cmax	Heurística
[56]	F Bloqueo Cmax	TS
[57]	F b_i Cmax	MPF
[58]	F Disponibilidad Cmax	GA y TS
[59]	F Saltar γ	$\gamma \in \{\sum w_j F_j, \sum F_j\}$ Heurística: Conjunto NPS
[60]	F Saltar, S_{ijk} γ	$\gamma \in \{\sum w_j F_j, \sum C_{max}\}$ Heurística a medida y Conjunto NPS
[61]	F Estocástico Costos	GA y ATC heurtística
[62]	F b_i Ingresos	MPF
[63]	F Retrasos de tiempo Cmax	NP-Difícil, para 2 máquinas PFS no óptimo
[64]	F γ	$\gamma \in \{C_{max} \sum C_j, \sum w_j C_j, T_{max}, \sum T_j, \sum w_j T_j\}$ TS y GA, compara las 6 funciones objetivo
[65]	F Bloqueo $\sum w_j C_j$	GA
[66]	F Cmax	SS
[67]	F Cmax	ACO

Cuadro 2 continuación de la página previa

Referencia	Problema	Comentarios
[68]	F Bloqueo $\sum w_j C_j$	GA y CLP
[69]	F $\sum w_j T_j$	Heurística ATC y GA
[70]	F Cmax	IG
[71]	F2 $p_{ij} = p$, tiempo de retraso — Cmax	Heurística - uet (unit execution time): tiempo de ejecución por unidad
	F $\sum w_j T_j$	
[72]	F Tiempo de retraso Cmax	MPF
	F S_{ijk} Cmax	
	F γ	$\gamma \in \{\sum F_j, Cmax\}$ ACO y búsqueda local
[73]	F fmls, S_{ijk} γ	$\gamma \in \{Cmax \sum C_j, \sum w_j C_j, Tmax, \sum T_j, \sum w_j T_j\}$ SA, TS y GA
[74]	F Cmax	SA y TS
[75]	F Cmax	Comparación de PFS y NPFS makespan, por caso general
[76]	F fmls, configuración γ	$\gamma \in \{Cmax \sum C_j, \sum w_j C_j, Tmax, \sum T_j, \sum w_j T_j\}$ configuración SA depende de la familia de secuencia
[77]	F $\sum T_j$	TS
[78]	F2 Bloqueo $\sum w_j C_j$	GA
[79]	F b_i Costos	MPF
[80]	F S_{ijk} Costos	Heurística basada en MPF
[81]	F Cmax	Algoritmo evolutivo diferencial cuántico (Quantum differential evolutionary algorithm- QDEA)
[82]	F Cmax	Híbrido CLP y GA
[83]	Programación inversa - Cmax	Condiciones suficientes para una secuencia óptima
[84]	F Saltar Cmax	GA y TS
[85]	F2 Aprender Cmax	Basado en la heurística NEH
[86]	F $\sum w_j C_j$ & $\sum w_j T_j$	TS con perturbación progresiva
[87]	F2 rp Cmax	Análisis complejo, es NP-Difícil
[88]	F Aprender, disponibilidad $\sum F_j$	Heurística VFR
[89]	F Aprender, disponibilidad Costos	Híbrido Firefly-SA
[90]	F S_{ijk} $\sum w_j T_j$	Heurística búsqueda local
[66]	F Lote, fmls, r_i γ	$\gamma \in \{\sum F_j, \sum C_j, Cmax\}$ MPF
[91]	F Saltar, dr, S_{ijk} , disponibilidad, r_j $\sum w_j C_j$ & $\sum w_j T_j$	OM
[78]	F Cmax	ACO
[51]	F $b_i = n - 2$ Cmax	ACO

Cuadro 2 continuación de la página previa

Referencia	Problema	Comentarios
[92]	F S_{ijk}, P_{ij} Costos	Heurística MPF y OM, demanda incierta
[93]	F Lote, configuración Cmax	TS
[94]	Fm Cmax	Procedimientos de delimitación
[95]	F2 uet, Tiempo de retraso Cmax	Algoritmo B&B
[96]	F Cmax	ACO
[97]	F hr Cmax	Heurística: SS y PR
[98]	Fm S_{ijk}, r_j Cmax	GA y TS
[99]	F Aprender, disponibilidad, r_j $\sum F_j$	Heurística y SA
[100]	F Reserva Costos	GA
	F Configuración, disponibilidad Cmax	ACO
[101]	F R_j, P_{ij} Cmax & $\sum F_j$ & $\sum T_j$	CCP y FGP
[102]	F OA= Aceptación de la orden $\sum w_j T_j$	TS-GA
[103]	F Aprender, S_{ijk} Cmax & $\sum F_j$ & $\sum T_j$	Heurística de restricción aumentada
[104]	F $\sum C_j$	IG
[105]	F Cmax	OM
[106]	F Disponibilidad Cmax	OM
[107]	F Saltar $\sum F_j$	SA
[108]	F Lot-streaming Cmax	MPF

Esta tabla fue tomada del artículo *The Non-Permutation Flow-Shop scheduling problem: A literature review*[15]

8. Casos de implementación

Se realizará un problema de implementación **flow shop** de una sola máquina con 6 trabajos, cuyos tiempos de entrega y llegada son distintos. La **figura 20** presenta la información:

Fecha de llegada r_j	Fecha de entrega d_j	Trabajo	Tiempo req (días)
10-may	22-may	1	6
9-may	13-may	2	1
22-may	26-may	3	3
13-may	25-may	4	7
14-may	21-may	5	6
15-may	22-may	6	2

Figura 20: Información caso flow shop

Se implementaron las siguientes reglas de despacho: FCFS, LCFS, SPT, LPT, EDD, STR y RANDOM.

En la **figura 21** se ven dos tablas que contienen las reglas de despacho FCFS y LCFS.

FCFS									
Fecha de llegada rj	Fecha de entrega dj	Trabajo	Tiempo req (días)	Plazo (días)	Tiempo de flujo	c(j)	Tardanza	NT	E(j)
9-may	13-may	2	1	5	1	1	0	0	4
10-may	22-may	1	6	13	7	7	0	0	6
13-may	25-may	4	7	13	14	14	1	1	0
14-may	21-may	5	6	8	20	20	12	1	0
15-may	22-may	6	2	8	22	22	14	1	0
22-may	26-may	3	3	5	25	25	20	1	0
$\Sigma=$						89	47	4	10

LCFS									
Fecha de llegada rj	Fecha de entrega dj	Trabajo	Tiempo req (días)	Plazo (días)	Tiempo de flujo	c(j)	Tardanza	NT	E(j)
22-may	26-may	3	3	5	3	3	0	0	2
15-may	22-may	6	2	8	5	5	0	0	3
14-may	21-may	5	6	8	11	11	3	1	0
13-may	25-may	4	7	13	18	18	5	1	0
10-may	22-may	1	6	13	24	24	11	1	0
9-may	13-may	2	1	5	25	25	20	1	0
$\Sigma=$						86	39	4	5

Figura 21: Implementación regla de despacho FCFS y LCFS

La **figura 22** referencia las reglas de despacho SPT y LPT respectivamente.

SPT									
Fecha de llegada rj	Fecha de entrega dj	Trabajo	Tiempo req (días)	Plazo (días)	Tiempo de flujo	c(j)	Tardanza	NT	E(j)
9-may	13-may	2	1	5	1	1	0	0	4
15-may	22-may	6	2	8	3	3	0	0	5
22-may	26-may	3	3	5	6	6	1	1	0
10-may	23-may	1	6	14	12	12	0	0	2
14-may	21-may	5	6	8	18	18	10	1	0
13-may	25-may	4	7	13	25	25	12	1	0
$\Sigma=$						65	23	3	11

LPT									
Fecha de llegada rj	Fecha de entrega dj	Trabajo	Tiempo req (días)	Plazo (días)	Tiempo de flujo	c(j)	Tardanza	NT	E(j)
13-may	25-may	4	7	13	7	7	0	0	6
14-may	21-may	5	6	8	13	13	5	1	0
10-may	23-may	1	6	14	19	19	5	1	0
22-may	26-may	3	3	5	22	22	17	1	0
15-may	22-may	6	2	8	24	24	16	1	0
9-may	13-may	2	1	5	25	25	20	1	0
$\Sigma=$						110	63	5	6

Figura 22: Implementación regla de despacho SPT y LPT

Por último la **figura 23** muestra la regla de despacho Random.

RANDOM									
Fecha de llegada rj	Fecha de entrega dj	Trabajo	Tiempo req (días)	Plazo (días)	Tiempo de flujo	c(j)	Tardanza	NT	E(j)
13-may	25-may	4	7	13	14	14	1	1	0
10-may	22-may	1	6	13	7	7	0	0	6
22-may	26-may	3	3	5	25	25	20	1	0
14-may	21-may	5	6	8	20	20	12	1	0
15-may	22-may	6	2	8	22	22	14	1	0
9-may	13-may	2	1	5	1	1	0	0	4
$\Sigma=$						89	47	4	10

Figura 23: Implementación regla de despacho RANDOM

En la **figura 24** se presenta un cuadro resumen de las medidas de desempeño.

MEDIDAS DE DESEMPEÑO					
REGLA	TIEMPO PROMEDIO DE FLUJO	TARDANZA MAXIMA	TARDANZA PROMEDIO	NUMERO DE TAREAS TARDIAS	ADELANTO TOTAL
FCFS	14,8333333	20	7,83333333	4	10
LCFS	14,3333333	20	6,5	4	5
SPT	10,8333333	12	3,83333333	3	11
LPT	18,3333333	20	10,5	5	6
EDD	13	11	4,5	5	2
STR	13	11	4,5	5	2
RANDOM	14,8333333	20	7,83333333	4	10

Figura 24: Resumen medidas de desempeño

Y posteriormente con el diagrama de Gantt se puede evidenciar que el tiempo total de flujo es de 25 días como se observa en la **figura 25**

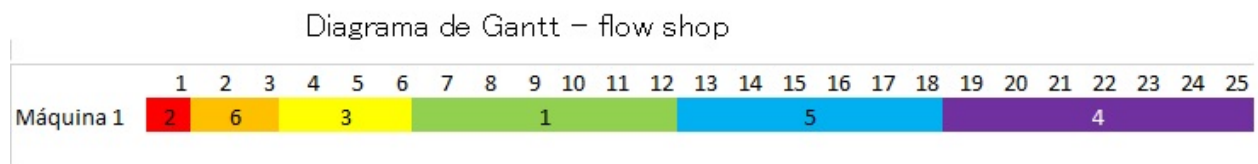


Figura 25: Diagrama de Gantt para flow shop

Un diagrama de Gantt para flow shop arrojará siempre el mismo horizonte de fabricación (*makespan*), sin importar la secuencia de los trabajos.

Para el ejemplo de implementación **flow shop** se concluye que la mejor regla de despacho para nuestro caso será la SPT, pues logra optimizar la mayoría de medidas de desempeño.

En la **figura 26** se muestra un gráfico de estrella que puede determinar cuál de las reglas se ajusta mejor a lo evaluado.

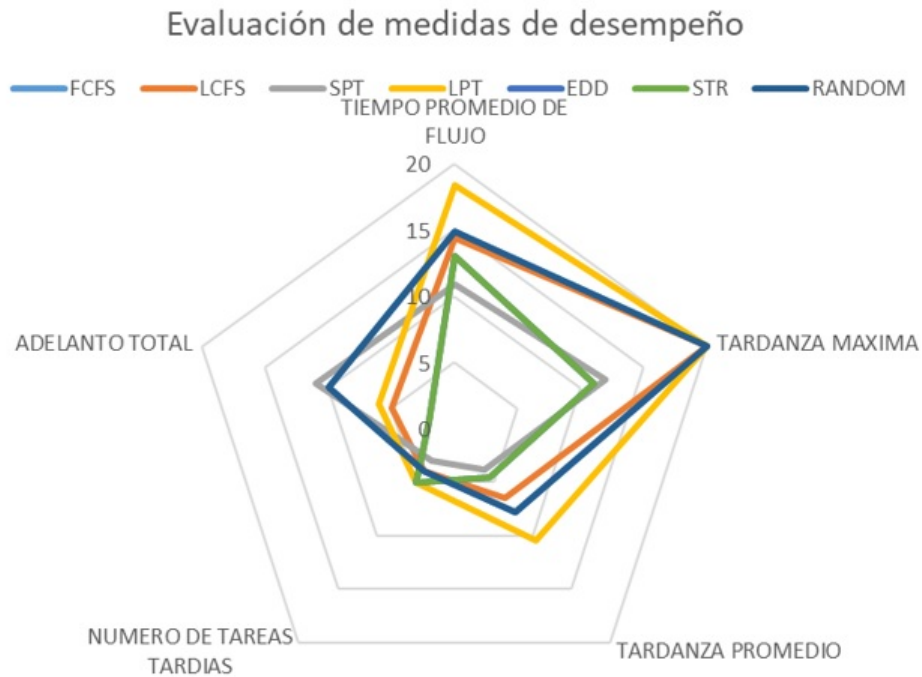


Figura 26: Evaluación de medidas de desempeño

En el siguiente ejemplo se implementa un caso de **job shop** de m máquinas con n trabajos utilizando el algoritmo de Giffler y Thomson, primeramente se planteará el modelo matemático. Se tiene la siguiente información:

$$\begin{aligned}m &= 1, 2, 3, 4 \\ n &= 1, 2, 3\end{aligned}$$

Se tiene una secuencia de producción dada así:

$$\begin{aligned}j=1 & 1, 3, 2 \\ j=2 & 2, 3, 4, 1 \\ j=3 & 1, 2, 4\end{aligned}$$

En la **figura 27** se aprecia la secuencia del trabajo número 1.

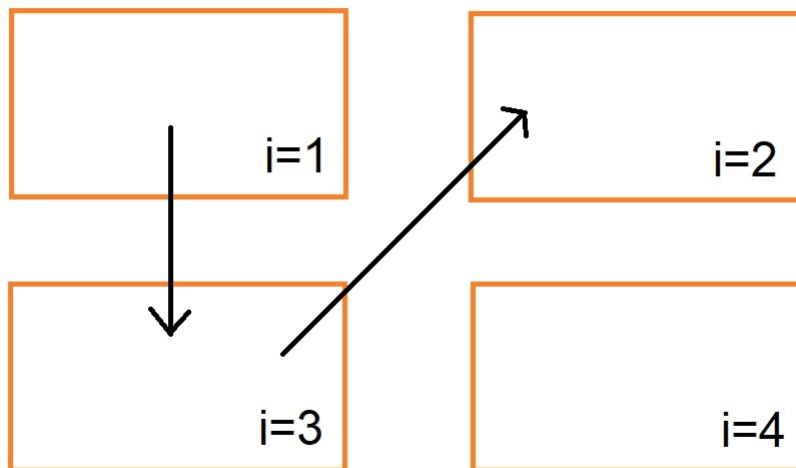


Figura 27: Secuencia 1

En la **figura 28** se aprecia la secuencia del trabajo número 2.

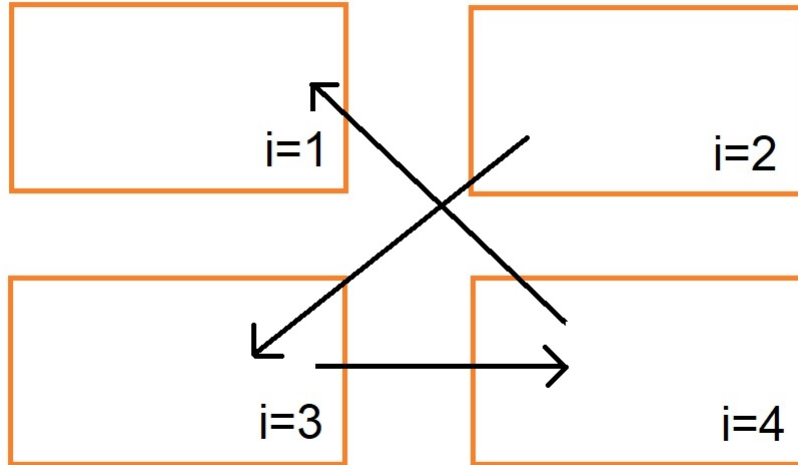


Figura 28: Secuencia 2

En la **figura 29** se aprecia la secuencia del trabajo número 3.

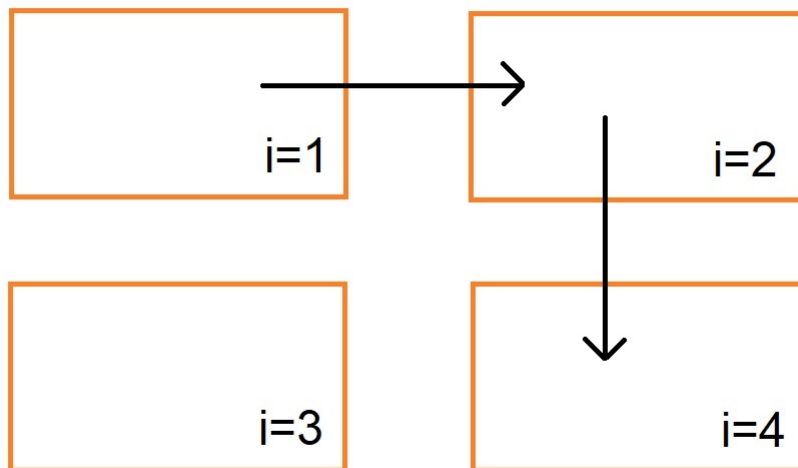


Figura 29: Secuencia 3

Los tiempos asociados a cada uno de los trabajos en las máquinas son:

Para el trabajo 1

$$P_{11} = 10$$

$$P_{31} = 7$$

$$P_{21} = 3$$

Para el trabajo 2

$$P_{22} = 5$$

$$P_{32} = 6$$

$$P_{42} = 8$$

$$P_{12} = 2$$

Para el trabajo 3

$$P_{13} = 2$$

$$P_{23} = 6$$

$$P_{43} = 9$$

Planteamiento del modelo lineal

En el proyecto de investigación "El sistema Productivo del Job Shop en el salón de clase" se describe el modelo como se presenta a continuación [109].

Definición de variables y parámetros

C_{max} = Tiempo total de proceso (*makespan*)

A = Conjunto de las restricciones de ruta $(i,j) \rightarrow (h,j)$

N = Conjunto de todas las operaciones a realizar

$i=1, \dots, m$: Número de máquinas a utilizar

$j=1, \dots, n$: Número de trabajos a realizar en las máquinas

Y_{ij} = Tiempo de iniciación de la operación (i,j)

P_{ij} = Duración de la operación (i,j)

La ecuación (6) representa la función objetivo que busca minimizar el tiempo total de proceso de los trabajos en el sistema:

$$\text{Min } C_{max} \tag{6}$$

Sujeto a:

$$Y_{hj} - Y_{ij} \geq P_{ij} \rightarrow \text{Para todo } (i,j) \rightarrow (h,j) \in A \tag{7}$$

$$C_{max} - Y_{ij} \geq P_{ij} \rightarrow \text{Para todo } (i,j) \in N \tag{8}$$

$$Y_{ij} - Y_{ik} \geq P_{ik} \rightarrow \text{Para todo } (i,k)y(i,j) \rightarrow i = 1, \dots, m. \tag{9}$$

$$Y_{ij} \geq 0 \rightarrow \text{Para todo } (i,j) \in N \tag{10}$$

El conjunto de ecuaciones (7)-(10) representan las restricciones del modelo.

Donde la ecuación número (7) representa la restricción de orden o ruta de los trabajos en las máquinas.

La ecuación número (8) corresponde a las restricciones de capacidad asociadas a los tiempos de los trabajos.

La ecuación número (9) representa las restricciones disyuntivas o de prioridad, donde se establecen las secuencias.

La ecuación número (10) corresponde a las restricciones de no negatividad, asociadas a los tiempos mayores a 0.

A continuación, se plantean las ecuaciones y se reemplazan los valores del problema.

$$Y_{hj} - Y_{ij} \geq P_{ij} \text{ restricciones de ruta}$$

$$Y_{31} - Y_{11} \geq P_{11}$$

$$Y_{21} - Y_{31} \geq P_{31}$$

$$Y_{32} - Y_{22} \geq P_{22}$$

$$Y_{42} - Y_{32} \geq P_{32}$$

$$Y_{12} - Y_{42} \geq P_{42}$$

$$Y_{23} - Y_{13} \geq P_{13}$$

$$Y_{43} - Y_{23} \geq P_{23}$$

$$Y_{31} - Y_{11} \geq 10$$

$$Y_{21} - Y_{31} \geq 7$$

$$Y_{32} - Y_{22} \geq 5$$

$$Y_{42} - Y_{32} \geq 6$$

$$Y_{12} - Y_{42} \geq 8$$

$$Y_{23} - Y_{13} \geq 2$$

$$C_{max} - Y_{ij} \geq P_{ij} \text{ restricciones de capacidad}$$

$$C_{max} - Y_{11} \geq P_{11}$$

$$C_{max} - Y_{31} \geq P_{31}$$

$$C_{max} - Y_{21} \geq P_{21}$$

$$C_{max} - Y_{22} \geq P_{22}$$

$$C_{max} - Y_{32} \geq P_{32}$$

$$C_{max} - Y_{42} \geq P_{42}$$

$$C_{max} - Y_{12} \geq P_{12}$$

$$C_{max} - Y_{13} \geq P_{13}$$

$$C_{max} - Y_{23} \geq P_{23}$$

$$\begin{aligned}
C_{max}-Y_{43} &\geq P_{43} \\
C_{max}-Y_{11} &\geq 10 \\
C_{max}-Y_{31} &\geq 7 \\
C_{max}-Y_{21} &\geq 3 \\
C_{max}-Y_{22} &\geq 5 \\
C_{max}-Y_{32} &\geq 6 \\
C_{max}-Y_{42} &\geq 8 \\
C_{max}-Y_{12} &\geq 2 \\
C_{max}-Y_{13} &\geq 2 \\
C_{max}-Y_{23} &\geq 6 \\
C_{max}-Y_{43} &\geq 9
\end{aligned}$$

$Y_{ik}-Y_{ij} \geq P_{ij}$ restricciones disyuntivas para la secuencia 1-2-3

i=1

$$\begin{aligned}
Y_{12}-Y_{11} &\geq P_{11} \\
Y_{13}-Y_{12} &\geq P_{12}
\end{aligned}$$

$$\begin{aligned}
Y_{12}-Y_{11} &\geq 10 \\
Y_{13}-Y_{12} &\geq 2
\end{aligned}$$

i=2

$$\begin{aligned}
Y_{22}-Y_{21} &\geq P_{22} \\
Y_{23}-Y_{22} &\geq P_{23}
\end{aligned}$$

$$\begin{aligned}
Y_{22}-Y_{21} &\geq 5 \\
Y_{23}-Y_{22} &\geq 6
\end{aligned}$$

i=3

$$Y_{32}-Y_{31} \geq P_{31}$$

$$Y_{32}-Y_{31} \geq 7$$

i=4

$$Y_{43}-Y_{42} \geq P_{42}$$

$$Y_{43}-Y_{42} \geq 8$$

$Y_{ij} \geq 0$ restricciones de positividad

Para **Job shop** se utilizó el algoritmo de Giffler y Thomson con reglas de despacho como método de solución.

La primera regla utilizada fue **SPT** se muestra en las siguientes figuras de la 30 a la 39 las iteraciones realizadas para un total de 10, así se obtuvo la secuencia de los trabajos.

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5		2,3 6	
Taladro m3	3,1 7		3,2 6			
Prensa m4			4,2 8		4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8

Figura 30: Primera iteración

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	
Taladro m3	3,1 7		3,2 6			
Prensa m4			4,2 8		4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5

Figura 31: Segunda iteración

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7		3,2 6			
Prensa m4			4,2 8		4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6

Figura 32: Tercera iteración

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6			
Prensa m4			4,2 8		4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6

Figura 33: Cuarta iteración

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6	10/16		
Prensa m4			4,2 8		4,3 9	

P 1,1 10
P 3,1 7
P 2,1 3

P 2,2 5
P 3,2 6
P 4,2 8
P 1,2 2

P 1,3 2
P 2,3 6
P 4,3 9

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6
3,2	6

Figura 34: Quinta iteración

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6	10/16		
Prensa m4			4,2 8	16/24	4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6
3,2	6

Figura 35: Sexta iteración

	J1		J2		J3	
Torno m1	1,1 10		1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6	10/16		
Prensa m4			4,2 8	16/24	4,3 9	24/33

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6
3,2	6
4,3	9

Figura 36: Séptima iteración

	J1		J2		J3	
Torno m1	1,1 10	10/20	1,2 2		1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6	10/16		
Prensa m4			4,2 8	16/24	4,3 9	24/33

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6
3,2	6
4,3	9

Figura 37: Octava iteración

	J1		J2		J3	
Torno m1	1,1 10	10/20	1,2 2	24/26	1,3 2	
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6	10/16		
Prensa m4			4,2 8	16/24	4,3 9	24/33

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6
3,2	6
4,3	9
1,2	2

Figura 38: Novena iteración

	J1		J2		J3	
Torno m1	1,1 10	10/20	1,2 2	24/26	1,3 2	33/35
Fresa m2	2,1 3	0/3	2,2 5	3/8	2,3 6	8/14
Taladro m3	3,1 7	3/10	3,2 6	10/16		
Prensa m4			4,2 8	16/24	4,3 9	24/33

P 1,1 10
P 3,1 7
P 2,1 3

P 2,2 5
P 3,2 6
P 4,2 8
P 1,2 2

P 1,3 2
P 2,3 6
P 4,3 9

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
2,2	5
2,3	6
3,2	6
4,3	9
1,2	2
1,3	2

Figura 39: Décima iteración

Luego se procede a utilizar la regla de despacho **LPT** en las figuras 40 a la 49 se evidencian las 10 iteraciones realizadas para esta regla de despacho.

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3		2,2 5		2,3 6	
Taladro m3	3,1 7		3,2 6			
Prensa m4			4,2 8		4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8

Figura 40: Primera iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3		2,2 5		2,3 6	
Taladro m3	3,1 7		3,2 6			
Prensa m4			4,2 8	0/8	4,3 9	

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2

Figura 41: Segunda iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3		2,2 5		2,3 6	
Taladro m3	3,1 7		3,2 6			
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1 10
P 3,1 7
P 2,1 3

P 2,2 5
P 3,2 6
P 4,2 8
P 1,2 2

P 1,3 2
P 2,3 6
P 4,3 9

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9

Figura 42: Tercera iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3		2,2 5		2,3 6	
Taladro m3	3,1 7	10/17	3,2 6			
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9

Figura 43: Cuarta iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3		2,2 5		2,3 6	
Taladro m3	3,1 7	10/17	3,2 6	17/23		
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9
3,2	6

Figura 44: Quinta iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3	17/20	2,2 5		2,3 6	
Taladro m3	3,1 7	10/17	3,2 6	17/23		
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9
3,2	6

Figura 45: Sexta iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3	17/20	2,2 5	23/28	2,3 6	
Taladro m3	3,1 7	10/17	3,2 6	17/23		
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9
3,2	6
2,2	5

Figura 46: Séptima iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2		1,3 2	
Fresa m2	2,1 3	17/20	2,2 5	23/28	2,3 6	28/34
Taladro m3	3,1 7	10/17	3,2 6	17/23		
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1	10	P 2,2	5	P 1,3	2
P 3,1	7	P 3,2	6	P 2,3	6
P 2,1	3	P 4,2	8	P 4,3	9
		P 1,2	2		

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9
3,2	6
2,2	5
2,3	6

Figura 47: Octava iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2	28/30	1,3 2	
Fresa m2	2,1 3	17/20	2,2 5	23/28	2,3 6	28/34
Taladro m3	3,1 7	10/17	3,2 6	17/23		
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1 10
P 3,1 7
P 2,1 3

P 2,2 5
P 3,2 6
P 4,2 8
P 1,2 2

P 1,3 2
P 2,3 6
P 4,3 9

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9
3,2	6
2,2	5
2,3	6

Figura 48: Novena iteración

	J1		J2		J3	
Torno m1	1,1 10	0/10	1,2 2	28/30	1,3 2	34/36
Fresa m2	2,1 3	17/20	2,2 5	23/28	2,3 6	28/34
Taladro m3	3,1 7	10/17	3,2 6	17/23		
Prensa m4			4,2 8	0/8	4,3 9	8/17

P 1,1 10
P 3,1 7
P 2,1 3

P 2,2 5
P 3,2 6
P 4,2 8
P 1,2 2

P 1,3 2
P 2,3 6
P 4,3 9

Trabajos	Precedencia	Tiempo de proceso
1,1	-	10
1,2	1,1	2
1,3	1,2	2
2,1	-	3
2,2	2,1	5
2,3	2,2	6
3,1	-	7
3,2	3,1	6
4,2	-	8
4,3	4,2	9

Tareas disponibles	Tiempo
1,1	10
2,1	3
3,1	7
4,2	8
1,2	2
4,3	9
3,2	6
2,2	5
2,3	6
1,3	2

Figura 49: Décima iteración

Se utiliza el software LEKIN cargando el modelo de Job shop para el entorno de trabajo, como se plantea en la **figura 50**.

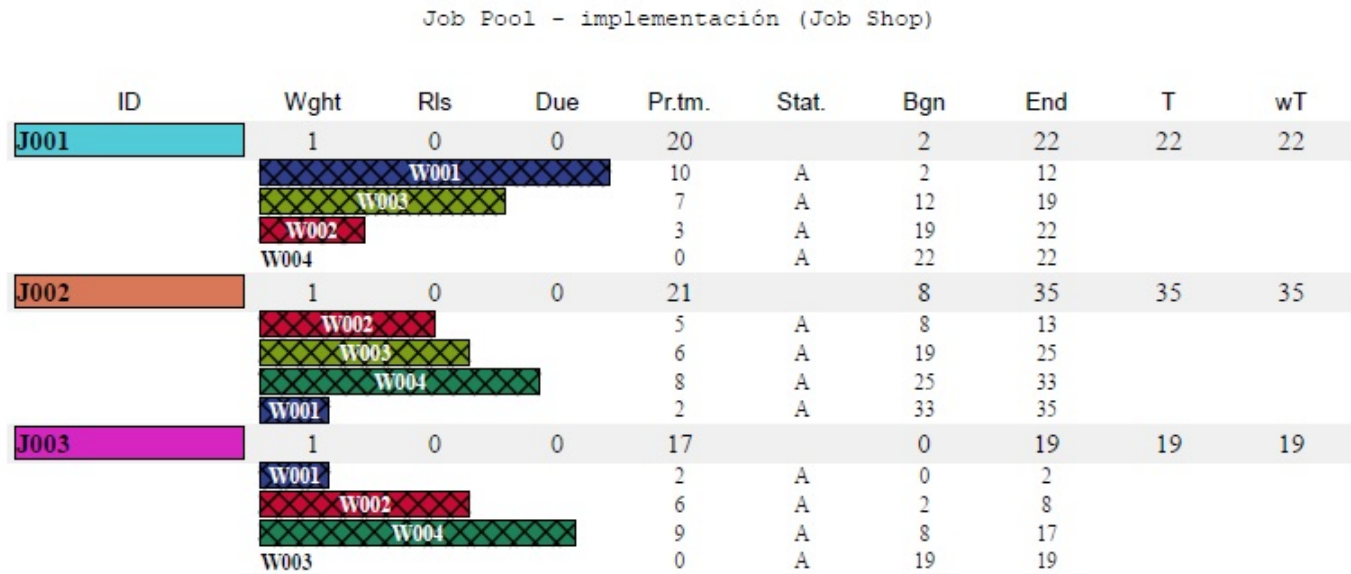


Figura 50: Interfaz LEKIN con implementación

En la **figura 51** se relacionan también las medidas de desempeño.

Sequence - implementación (SPT)

Mch/Job	Setup	Start	Stop	Pr.tm
W001	0			14
J003	0	0	2	2
J001	0	2	12	10
J002	0	33	35	2
W002	0			14
J003	0	2	8	6
J002	0	8	13	5
J001	0	19	22	3
W003	0			13
J001	0	12	19	7
J003	0	19	19	0
J002	0	19	25	6
W004	0			17
J003	0	8	17	9
J001	0	22	22	0
J002	0	25	33	8
Summary				
<i>Time</i>		1		
C_{max}			35	
T_{max}			35	
ΣU_j			3	
ΣC_j			76	
ΣT_j			76	
$\Sigma w_j C_j$			76	
$\Sigma w_j T_j$			76	

Figura 51: Interfaz LEKIN con implementación y medidas de desempeño

Para la regla de despacho se obtiene el mismo resultado del algoritmo Giffler y Thomson.

En la **figura 52** se muestra el diagrama de Gantt para la regla de despacho **SPT**.

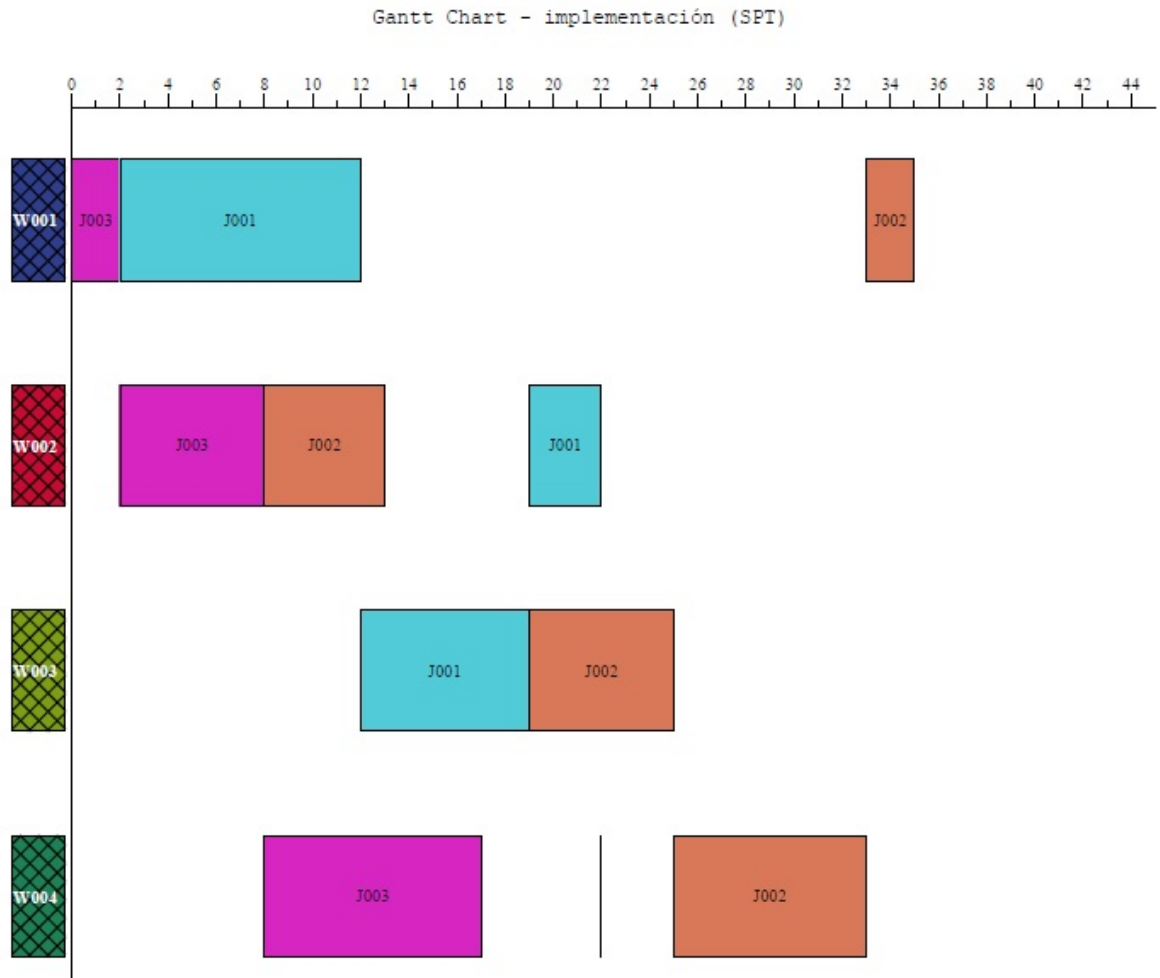


Figura 52: Gantt con la regla de despacho SPT

Y este resultado arroja el programa LEKIN para la secuencia con la regla **LPT** como se muestra en la **figura 53**

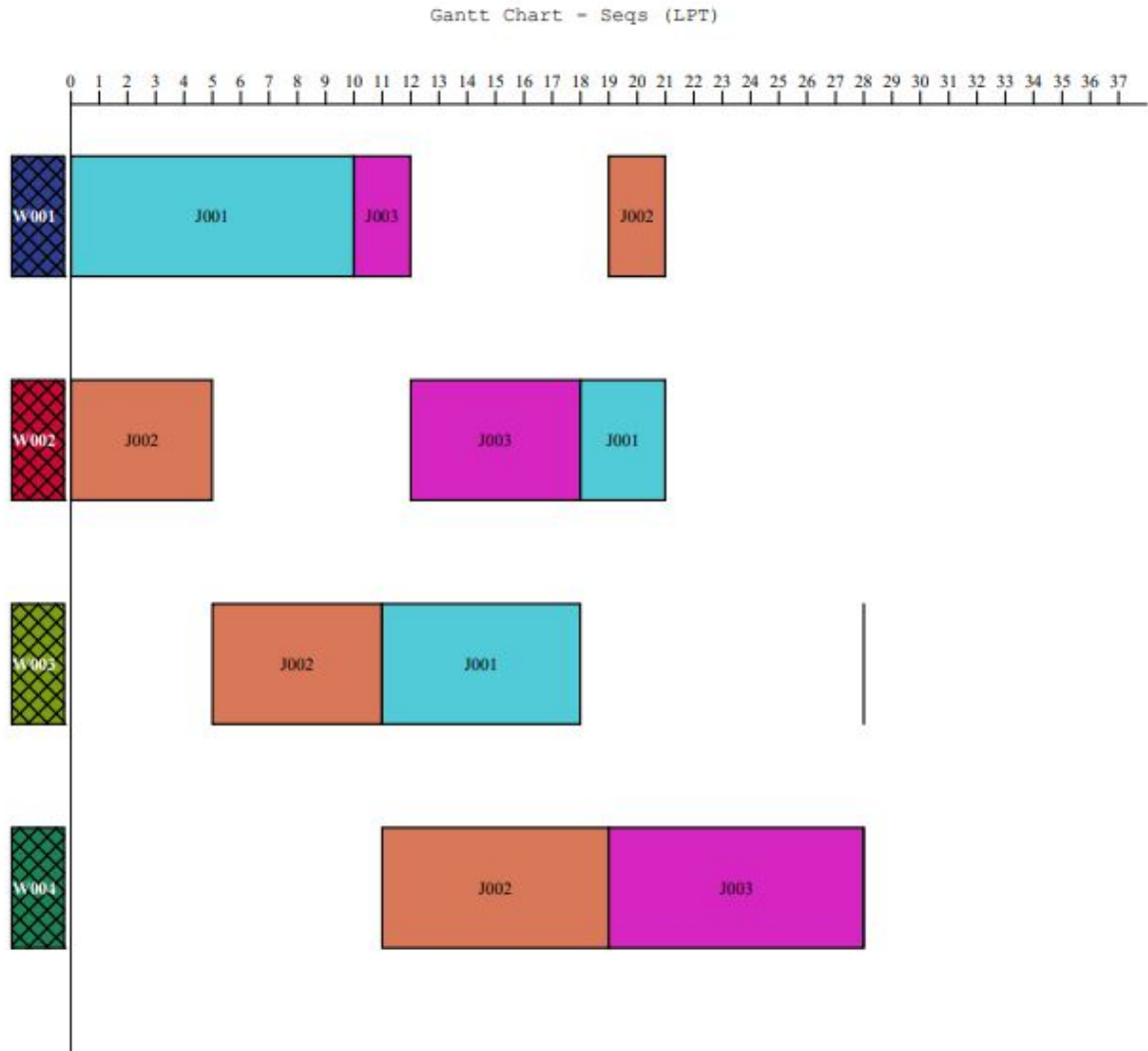
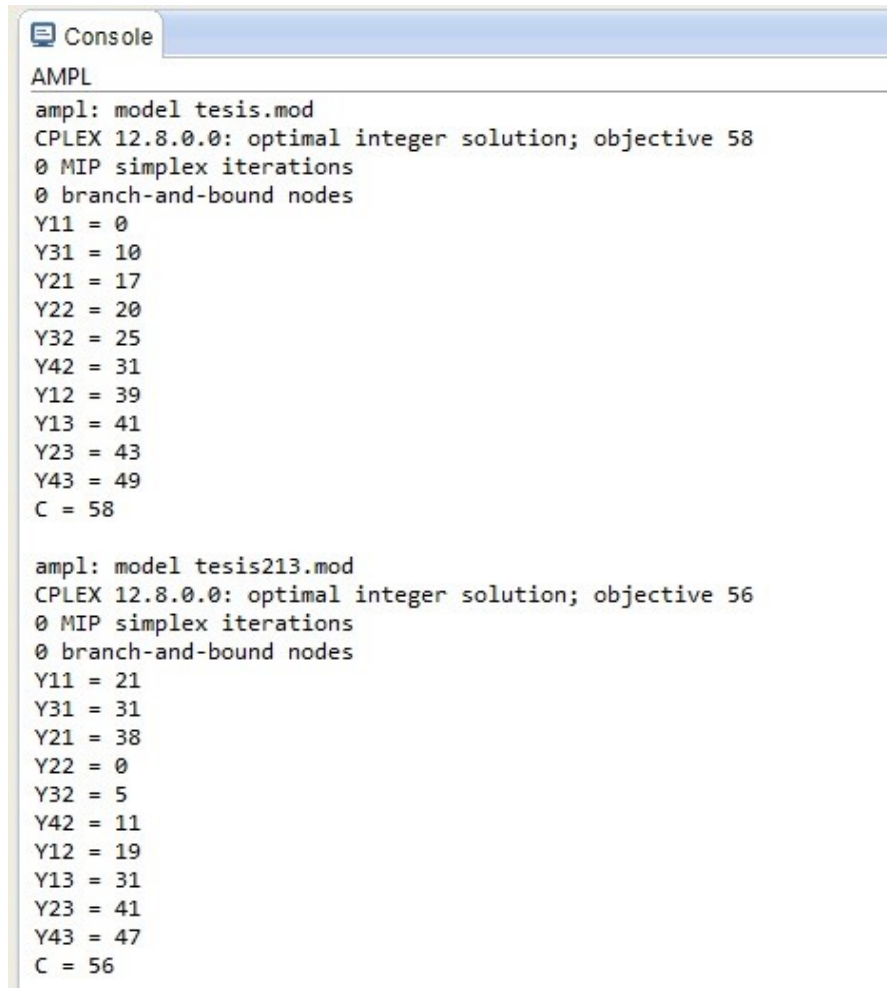


Figura 53: Gantt con la regla de despacho LPT

También se utilizó el software **AMPL** para implementar el modelo matemático planteado del **job shop**, pero se obtuvieron distintos resultados pues se evidencia que el software prioriza las restricciones disyuntivas las cuales son las encargadas de la secuencia de los trabajos en las máquinas. Los resultados obtenidos fueron:

La **figura 54** relaciona las secuencias 1-2-3 y 2-1-3.

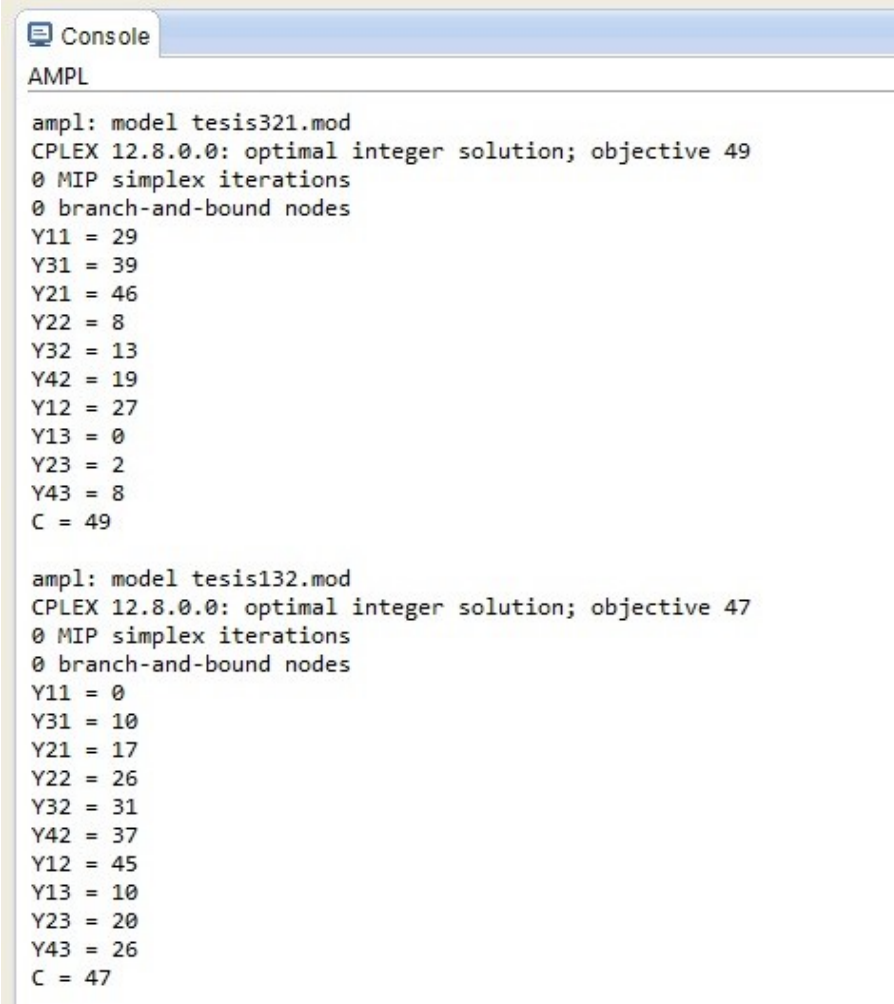


```
Console
AMPL
ampl: model tesis.mod
CPLEX 12.8.0.0: optimal integer solution; objective 58
0 MIP simplex iterations
0 branch-and-bound nodes
Y11 = 0
Y31 = 10
Y21 = 17
Y22 = 20
Y32 = 25
Y42 = 31
Y12 = 39
Y13 = 41
Y23 = 43
Y43 = 49
C = 58

ampl: model tesis213.mod
CPLEX 12.8.0.0: optimal integer solution; objective 56
0 MIP simplex iterations
0 branch-and-bound nodes
Y11 = 21
Y31 = 31
Y21 = 38
Y22 = 0
Y32 = 5
Y42 = 11
Y12 = 19
Y13 = 31
Y23 = 41
Y43 = 47
C = 56
```

Figura 54: Secuencia Job shop 1-2-3 y 2-1-3

Las secuencias 3-2-1 y 1-3-2 se muestran en la **figura 55**.



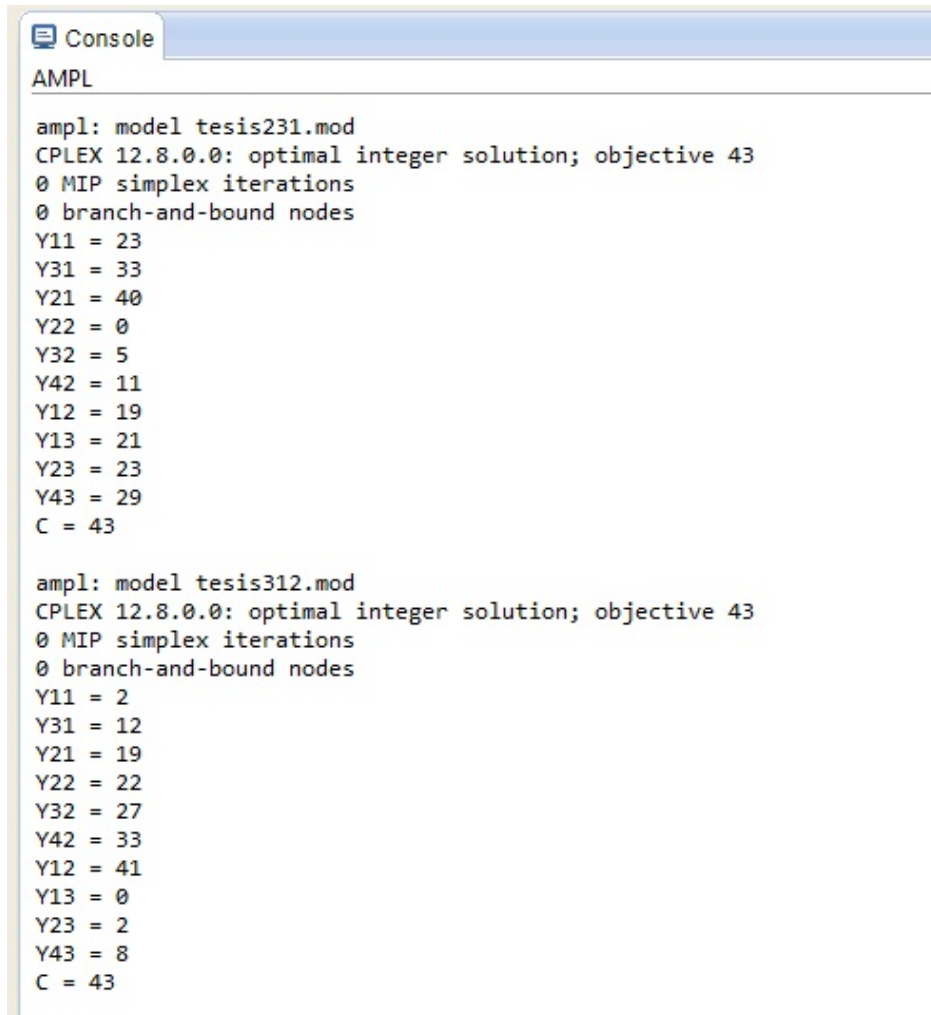
```
Console
AMPL

ampl: model tesis321.mod
CPLEX 12.8.0.0: optimal integer solution; objective 49
0 MIP simplex iterations
0 branch-and-bound nodes
Y11 = 29
Y31 = 39
Y21 = 46
Y22 = 8
Y32 = 13
Y42 = 19
Y12 = 27
Y13 = 0
Y23 = 2
Y43 = 8
C = 49

ampl: model tesis132.mod
CPLEX 12.8.0.0: optimal integer solution; objective 47
0 MIP simplex iterations
0 branch-and-bound nodes
Y11 = 0
Y31 = 10
Y21 = 17
Y22 = 26
Y32 = 31
Y42 = 37
Y12 = 45
Y13 = 10
Y23 = 20
Y43 = 26
C = 47
```

Figura 55: Secuencia Job shop 3-2-1 y 1-3-2

Posteriormente las secuencias 2-3-1 y 3-1-2 se muestran en la **figura 56**.



```
Console
AMPL

ampl: model tesis231.mod
CPLEX 12.8.0.0: optimal integer solution; objective 43
0 MIP simplex iterations
0 branch-and-bound nodes
Y11 = 23
Y31 = 33
Y21 = 40
Y22 = 0
Y32 = 5
Y42 = 11
Y12 = 19
Y13 = 21
Y23 = 23
Y43 = 29
C = 43

ampl: model tesis312.mod
CPLEX 12.8.0.0: optimal integer solution; objective 43
0 MIP simplex iterations
0 branch-and-bound nodes
Y11 = 2
Y31 = 12
Y21 = 19
Y22 = 22
Y32 = 27
Y42 = 33
Y12 = 41
Y13 = 0
Y23 = 2
Y43 = 8
C = 43
```

Figura 56: Secuencia Job shop 2-3-1 y 3-1-2

El mínimo C (makespan) para el problema utilizando el software AMPL fue de 43, mientras que usando el algoritmo y el programa LEKIN se lograba encontrar un C de 35. El modelo matemático planteado no es suficientemente robusto debido a que toma una única secuencia de producción para todas las maquinas.

El Branch and Bound permite llegar a una secuencia óptima para cada uno de los trabajos, se aplicó para el caso de implementación Job Shop en la **heurística de cambios de cuello de botella**.

En la **figura 57** se presenta el grafo disyuntivo con las precedencias y las posibles secuencias de cada trabajo. Las líneas punteadas son las posibles rutas que puede tomar la secuencia.

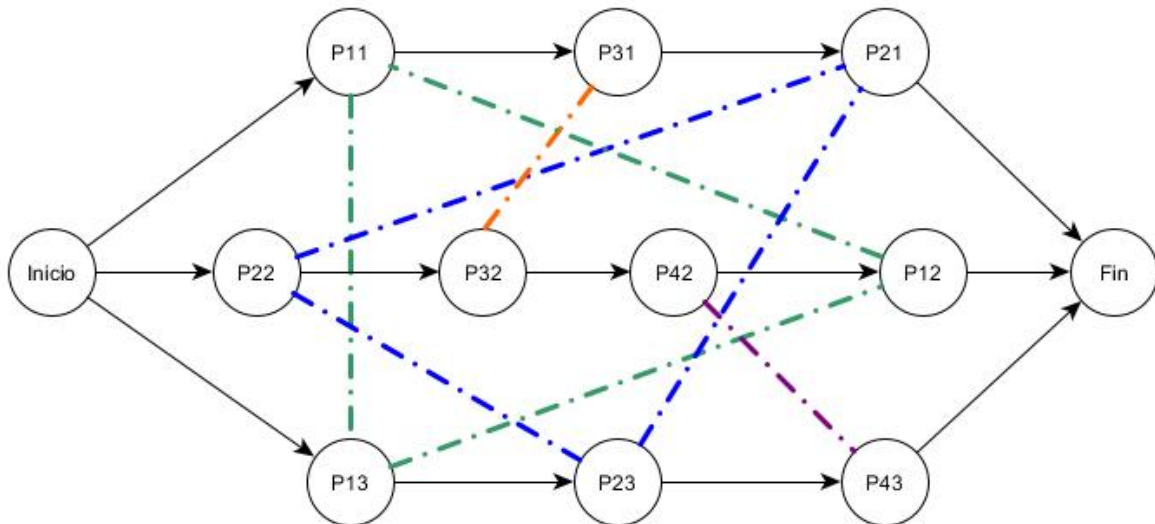


Figura 57: Descripción del grafo disyuntivo

Para el trabajo 1

$$P_{11} = 10$$

$$P_{31} = 7$$

$$P_{21} = 3$$

Para el trabajo 2

$$P_{22} = 5$$

$$P_{32} = 6$$

$$P_{42} = 8$$

$$P_{12} = 2$$

Para el trabajo 3

$$P_{13} = 2$$

$$P_{23} = 6$$

$$P_{43} = 9$$

Para poder determinar la secuencia es necesario utilizar el B&B y determinar la orientación de la ruta en cada una de las máquinas. Primero se realiza la ruta crítica y se determina la máquina con la menor holgura o con holgura cero, para empezar con las posibles secuencias de trabajo, como se muestra en la **figura 58**.

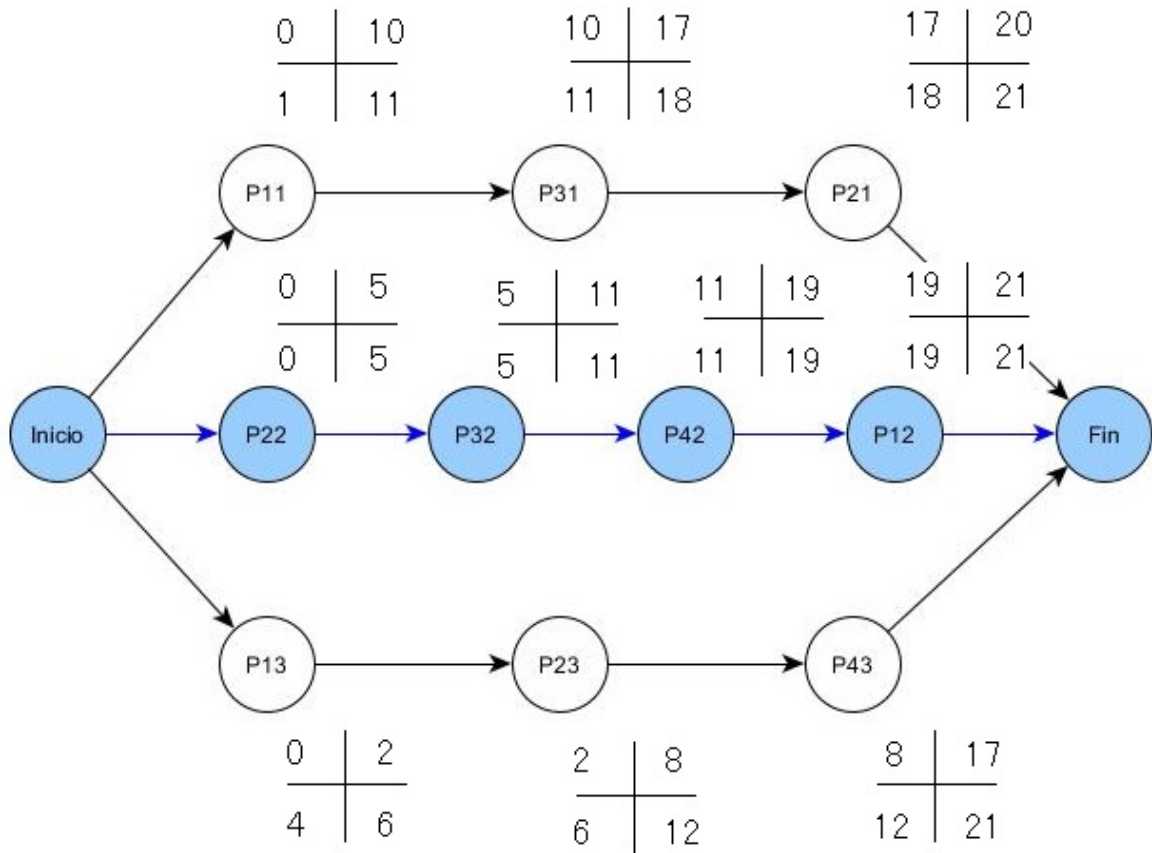


Figura 58: Ruta crítica inicial

Con esta primer ruta crítica se obtiene la holgura cero en la máquina 2. Se procede a realizar el B&B para esta máquina.

En la **figura 59** se realiza la ramificación por la máquina con holgura 0 y sus posibles resultados. Se realiza el siguiente procedimiento:

1. Se analizan cuántos trabajos se procesan en la máquina sin holgura. En este caso la máquina 2 procesa todos los trabajos 1,2 y 3.
2. Se realiza la suma de cada uno de los P_i que intervienen en este caso, para obtener como resultado P; como se dijo anteriormente los trabajos 1, 2 y 3 tienen los siguientes tiempos $P_1= 3$, $P_2=5$ y $P_3=6$. Por lo tanto la suma dará el siguiente resultado $P_1 + P_2 + P_3 = P$. Reemplazando se tiene $P= 14$.
3. Se debe identificar el d_i (*Due Date*) o fecha de entrega, para cada uno de los trabajos que intervienen. $d_1= 21$, $d_2=5$ y $d_3=8$.
4. Se deben tener en cuenta los r_j de cada trabajo o sea los tiempos de disponibilidad o el momento en que el trabajo ya se puede realizar. Esto puede ser un limitante a la hora de hallar nuestros valores de P o t.
5. Se hallan los valores t que se referencian en la **figura 59**. De la forma que se mostró en la ecuación número 5:

$$t = P - d_i \tag{11}$$

La ecuación (11) es la misma ecuación definida anteriormente como la ecuación (5) Para: $t_1=14-21=-7 = 0$, al tener un valor negativo t_1 se convierte en 0 automáticamente pues no existen tiempos negativos. $t_2=14-5 = 9$ y $t_3=14-8 = 6$, en resumen se tiene $t_1=0$, $t_2=9$ y $t_3=6$

6. Se escoge el tiempo menor o igual a 0, hallado en el numeral anterior en este caso nos habla de finalizar la secuencia con el trabajo 1, pues ese trabajo arrojó como resultado el menor tiempo.
7. Se ramifica con el B&B las dos posibles secuencias que son penúltimo trabajo el número 2 y penúltimo trabajo el número 3 para así obtener la secuencia de la máquina.
8. Se procede a hallar los nuevos tiempos t para el penúltimo trabajo en este caso se ve que el último trabajo tenía un tiempo de 0 (trabajo 1). Se realiza lo mismo del paso 5. Se suman los $P_2+P_3=11$. $P=11$. Y así se obtienen los valores del penúltimo trabajo en este caso $t_2= 11 - 5 = 6$ y $t_3=11 - 6 = 5$. Se toma el menor valor para t.

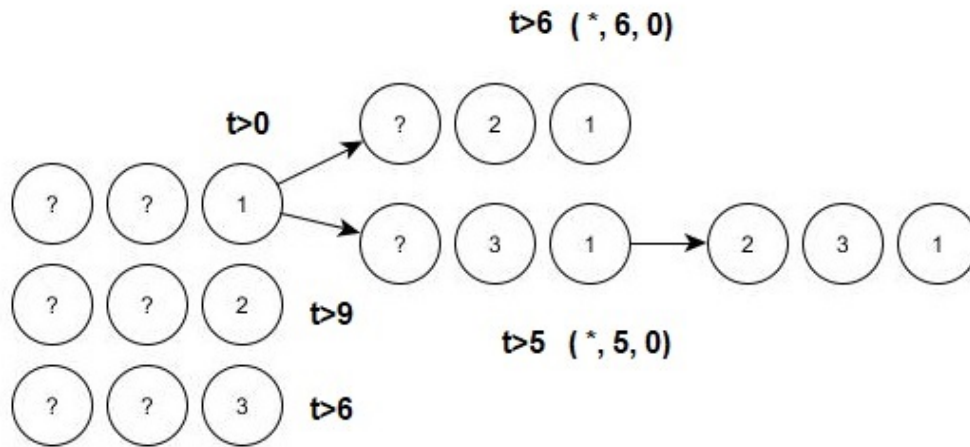


Figura 59: B&B máquina 2

Se llega a una secuencia para la máquina 2 de la siguiente manera: 2-3-1. Se debe recalculer la ruta crítica con la secuencia obtenida como se muestra en la **figura 60**.

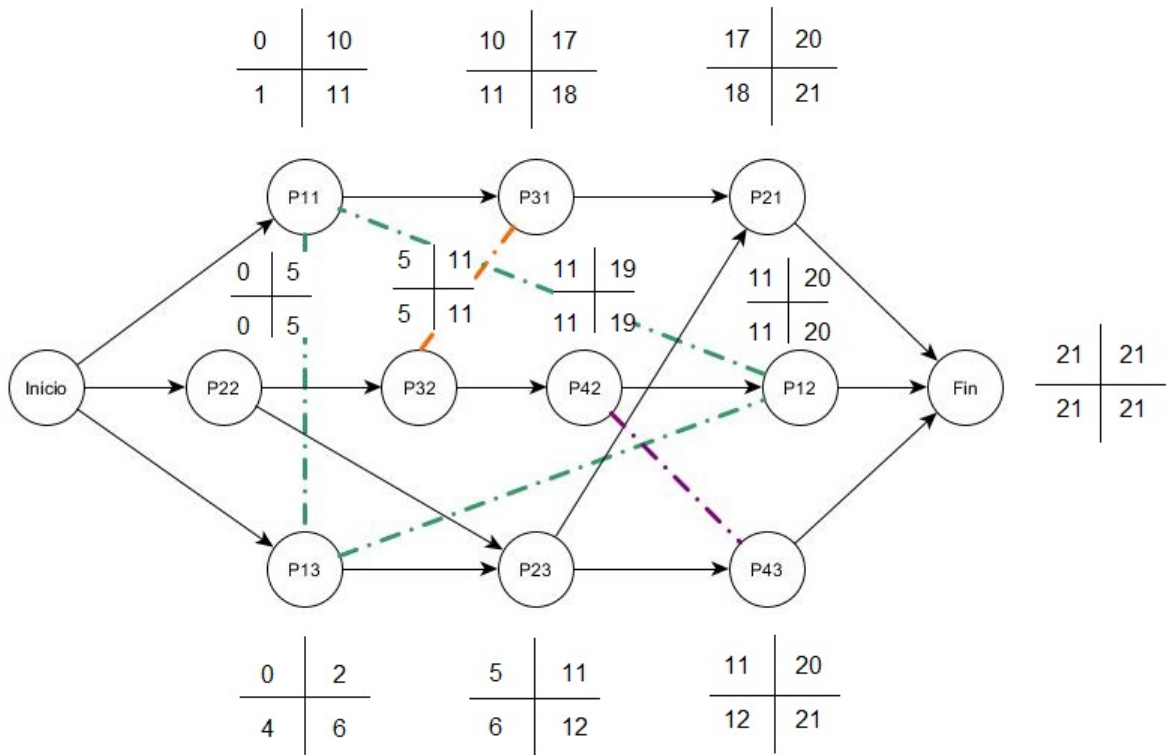


Figura 60: Recalculo de ruta crítica con máquina 2

Ahora en la **figura 61** se elije la máquina 1 porque es donde se presenta la menor holgura y se realiza el B&B para generar la secuencia óptima.

Se realizan los pasos anteriores obteniendo los valores de P y t : $P_1 + P_2 + P_3 = P$.
 $P=10 + 2 + 2 = 14$

Las fechas de entrega para cada los trabajos 1, 2 y 3 son 11, 21 y 6 respectivamente. Con esta información ya se pueden calcular los valores de t .

Teniendo: $t_1=14 - 11= 3$, $t_2=14 - 21 = -7 = 0$ y $t_3=14 - 6 = 8$, en resumen se tiene $t_2 = 0$ ha de finalizar la secuencia por prioridad. Se ramifica y se busca el nuevo tiempo menor para así completar toda la secuencia, teniendo como penúltimos trabajos el 3 y el 1. Los valores de t para el penúltimo trabajo deben ser analizados de la siguiente manera inicialmente en la **figura 60** se ve que el trabajo 2 puede arrancar en el momento 19 porque tiene un $r_j=19$, por lo tanto nuestro $P= 10+2= 12$ que se obtiene de la suma de los tiempos de los trabajos 3 y 1 no tiene validez porque aún la máquina no está disponible hasta el momento 19. Por lo tanto se tiene que nuestra $P=19$. Y allí se calcula nuestros tiempos $t_3=19 - 6 = 13$ y $t_1=19 - 11 = 8$, por lo tanto ya se obtiene la nueva secuencia tomando el menor valor de t .

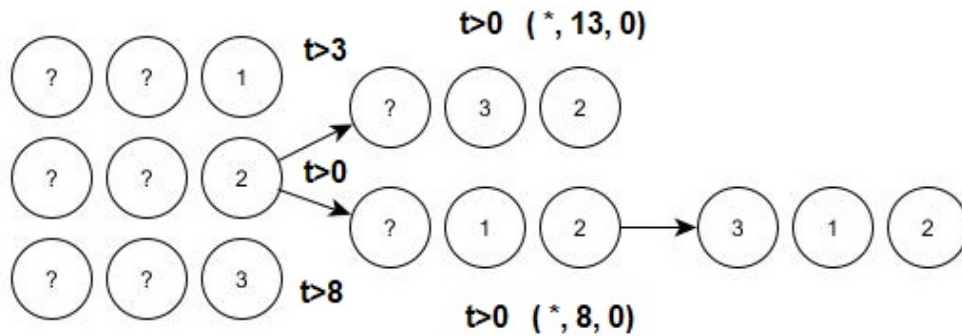


Figura 61: B&B máquina 1

La secuencia 3-1-2 es la óptima para esta máquina, entonces se procede a recalculer la ruta crítica y seleccionar la máquina con menos holgura o con holgura cero como se muestra en la **figura 62**.

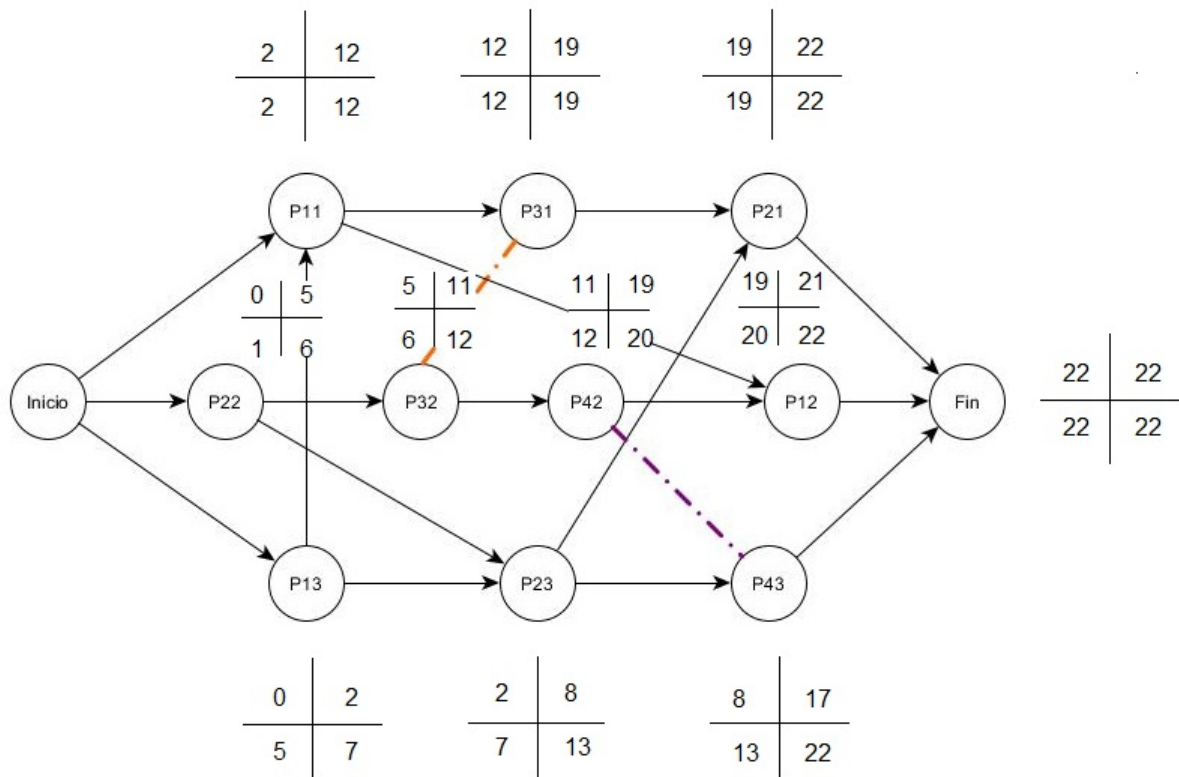


Figura 62: Recálculo de ruta crítica con máquina 1

Al realizar la ruta crítica, la próxima máquina será la número 3 puesto que presenta la menor holgura, en la **figura 63** se visualiza.

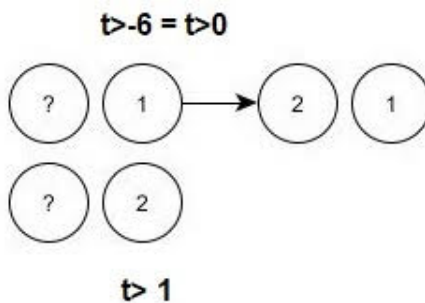


Figura 63: B&B máquina 3

Ahora se deben hallar los tiempos para la máquina número 3 que procesa tan solo los trabajos 1 y 2, en este caso si se encuentra el último trabajo ya se tendrá la secuencia completa de la máquina. para ambos se tiene un $t=0$ pues se tienen los siguientes valores $P_1=7$ y $P_2=6$. Por lo tanto $P=13$. Se tienen unas fechas de entrega para el trabajo 1 y 2 de 19 y 12 respectivamente. Por lo tanto se escoge el t menor y esa será la secuencia. $t_1=13-19=-6=0$ y $t_2=13-12=1$.

Obteniendo la secuencia 2-1. La nueva ruta crítica será la mostrada en la **figura 64**.

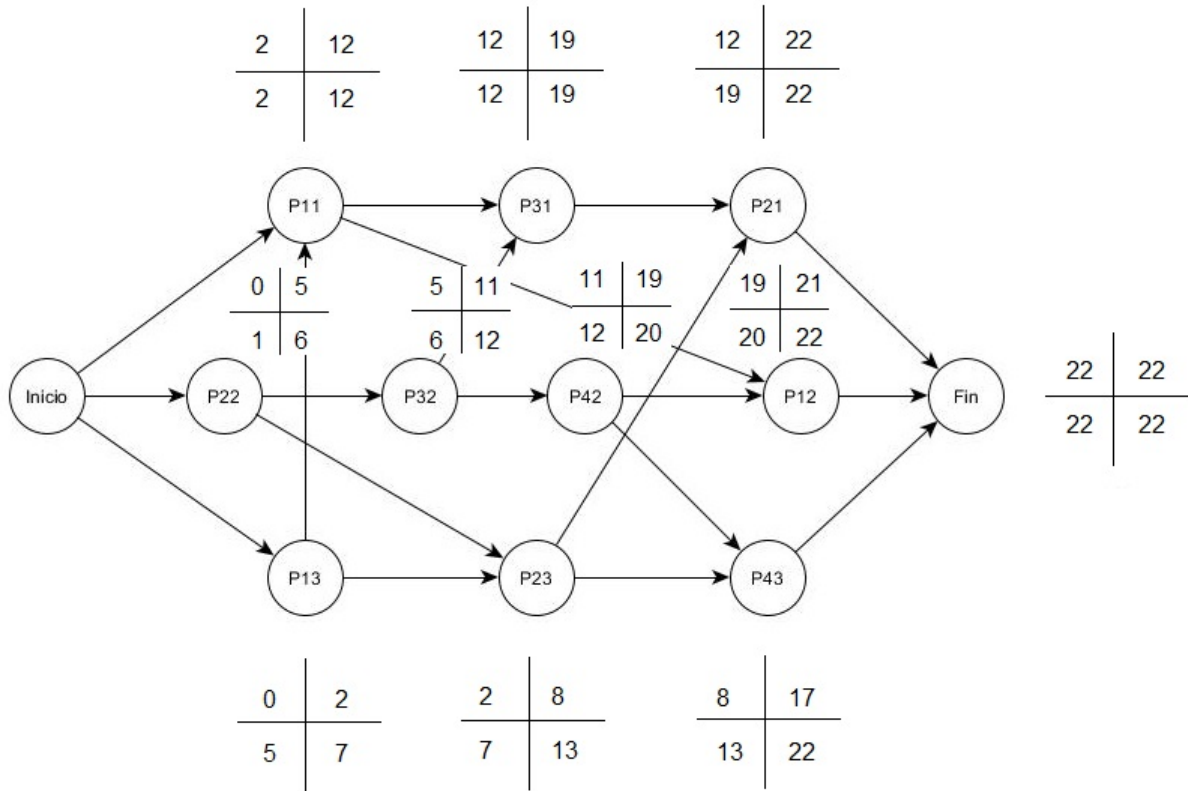


Figura 64: Recalculo de ruta crítica con máquina 3

Se procede a realizar el B&B para encontrar la secuencia de la última máquina la número 4, en la **figura 65** se visualiza la última ramificación para llegar a la solución del problema.

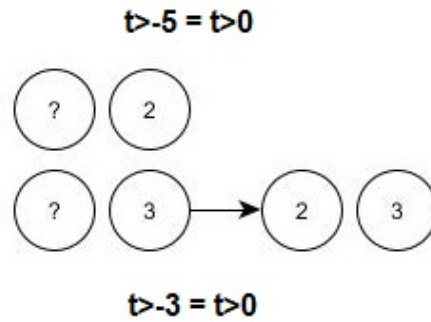


Figura 65: B&B máquina 4

Se hallan los valores para t , la máquina 4 solo procesa los trabajos 2 y 3, lo que nos indica que obteniendo el último trabajo se tiene la secuencia completa. Para este caso ambos valores de t son negativos se escoge la secuencia 2-3 ya que el trabajo 2 tiene una fecha de entrega más próxima.

Al obtener la secuencia óptima para la máquina 4 se obtiene entonces la ruta programada para todas las máquinas minimizando la tardanza total del sistema.

Resumen de resultados

El **cuadro 3** nos presente el resumen de resultados obtenidos para cada una de las máquinas y la secuencia óptima.

Máquina	Secuencia
1	3-1-2
2	2-3-1
3	2-1
4	2-3

Cuadro 3: Resumen de resultados heurística cuello de botella

Se comprueba con el programa LEKIN la secuencia para cada una de las máquinas y arroja los datos de la **figura 66**.

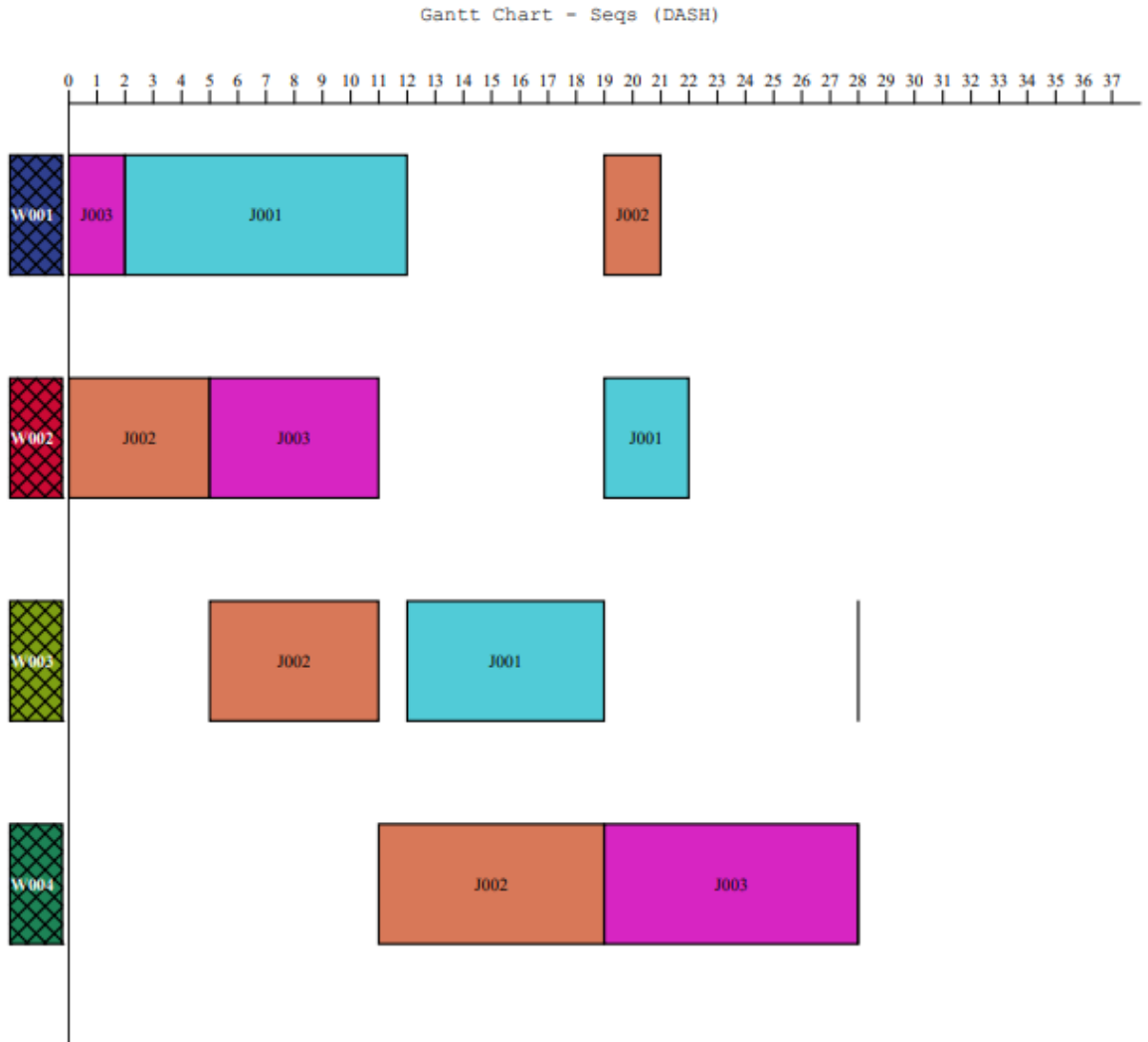


Figura 66: Diagrama de Gantt job shop

En el **cuadro 4** se presenta una tabla comparativa para los métodos usados en la implementación del caso Job shop

Método	Secuencia (Máquina)			
	1	2	3	4
SPT	3-1-2	3-2-1	1-2	3-2
LPT	1-3-2	2-3-1	2-1	2-3
H. Cuello de botella	3-1-2	2-3-1	2-1	2-3

Cuadro 4: Tabla comparativa de métodos

9. Análisis de resultados

Al realizar diversos métodos de implementación se tiene claro, que el problema de secuenciación de tareas es un problema robusto, que debe ser modelado e implementado dependiendo de las necesidades de cada organización, la planeación es muy importante para la toma de decisiones en una empresa, lograr planear y optimizar sus procesos productivos puede cumplir muchos de sus objetivos. Para esto se deben identificar muy bien cada uno de los elementos de un problema de secuenciación y así modelarlo de la manera más precisa para posteriormente utilizar uno de los métodos planteados anteriormente para buscar su solución óptima.

Los algoritmos y las heurísticas usadas pueden ser implementadas en distintos campos, los softwares utilizados arrojaron resultados que comprobaron los modelos y las implementaciones que se realizaron. Se evidencia que la literatura ha explorado el campo de la secuenciación buscando cada vez soluciones más eficaces al problema.

Los softwares usados fueron implementados para corroborar la información, en ocasiones difirieron resultados ya que el software puede trabajar y priorizar ciertas restricciones como en el caso de AMPL que no escanea todas las rutas posibles por sus restricciones de precedencia iniciales y se debe hacer una corrida para cada una de las secuencias como se mostró en páginas anteriores.

En este trabajo de investigación se exploraron las técnicas aproximadas con algoritmos desarrollados como lo son heurísticas y reglas de despacho, también las técnicas de optimización enumerativas como lo son el *branch and bound* y la optimización matemática, donde en optimización matemática se planteo el modelo de programación lineal entera y entera mixta.

10. Conclusiones

- Se lograron identificar las medidas de desempeño asociadas a los problemas de secuenciación de tareas siendo estas muy importantes para lograr plantear el modelo basándose en los resultados deseados y así plantear la función objetivo idónea de nuestro problema.
- Se describieron las reglas de despacho de la literatura actual y también los distintos métodos de solución que han surgido a lo largo de las décadas, este problema de secuenciación se aborda desde distintas técnicas de solución como se mostraba en la **figura 8**. Se implementaron heurísticas y algoritmos para dar solución a problemas planteados.
- Se lograron identificar los elementos esenciales de un problema de secuenciación de tareas para así tener claro los pasos a seguir a la hora de plantear el modelo. Estos elementos son:
 1. El ambiente de producción de cada organización asociado a los recursos y layout.
 2. Las restricciones asociadas a el proceso, las operaciones y lo concerniente a la producción y su cadena.
 3. Variables y parámetros a relacionar.
 4. Medidas de desempeño que se quieren optimizar.

Referencias

- [1] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.
- [2] J. Jablonsky and V. Skocdopolova, “Análisis y optimización del proceso de producción en una empresa procesadora de leche,” *Información tecnológica*, vol. 28, no. 4, pp. 39–46, 2017.
- [3] M. L. Pinedo, *Scheduling: theory, algorithms, and systems*. Springer, 2016.
- [4] J. F. Cox III and M. S. Spencer, *The constraints management handbook*. CRC Press, 1997.
- [5] H. A. Taha, *Investigación de operaciones*. Pearson Educación, 2004.
- [6] M. R. Garey, D. S. Johnson, and R. Sethi, “The complexity of flowshop and jobshop scheduling,” *Mathematics of operations research*, vol. 1, no. 2, pp. 117–129, 1976.
- [7] C. E. Maldonado, N. A. Gómez-Cruz *et al.*, “Modelamiento y simulación de sistemas complejos,” *Borradores de Investigación: Serie documentos Administración, ISSN 0124-8219, No. 66 (Febrero de 2010)*, 2010.
- [8] M. Pinedo, X. Chao, and J. Leung, “Lekin®—flexible job-shop scheduling system, stern school of business,” *URL (<http://www.ieor.columbia.edu/~andrew/scheduling/Lekin.html>)*. 9th August, 1999.
- [9] P. Brucker and P. Brucker, *Scheduling algorithms*. Springer, 2007, vol. 3.
- [10] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Scheduling computer and manufacturing processes*. springer science & Business media, 2013.
- [11] J. Lever, M. Wallace, and B. Richards, “Constraint logic programming for scheduling and planning,” *BT technology journal*, vol. 13, no. 1, pp. 73–80, 1995.
- [12] D. A. Rossit, “Diseño y desarrollo de estructuras de planificación eficientes a través de técnicas de simulación y optimización aplicables a entornos productivos complejos,” 2018.
- [13] M. Hoyos M, “Programación secuencial: Elementos esenciales y modelos de problemas de programación,” 2017.

- [14] A. S. Jain and S. Meeran, “Deterministic job-shop scheduling: Past, present and future,” *European journal of operational research*, vol. 113, no. 2, pp. 390–434, 1999.
- [15] D. A. Rossit, F. Tohmé, and M. Frutos, “The non-permutation flow-shop scheduling problem: a literature review,” *Omega*, vol. 77, pp. 143–153, 2018.
- [16] A. Allahverdi, J. N. Gupta, and T. Aldowaisan, “A review of scheduling research involving setup considerations,” *Omega*, vol. 27, no. 2, pp. 219–239, 1999.
- [17] A. Allahverdi, C. Ng, T. E. Cheng, and M. Y. Kovalyov, “A survey of scheduling problems with setup times or costs,” *European journal of operational research*, vol. 187, no. 3, pp. 985–1032, 2008.
- [18] A. Allahverdi, “The third comprehensive survey on scheduling problems with setup times/costs,” *European Journal of Operational Research*, vol. 246, no. 2, pp. 345–378, 2015.
- [19] J. Błażewicz, W. Domschke, and E. Pesch, “The job shop scheduling problem: Conventional and new solution techniques,” *European journal of operational research*, vol. 93, no. 1, pp. 1–33, 1996.
- [20] T. Kis, “Job-shop scheduling with processing alternatives,” *European Journal of Operational Research*, vol. 151, no. 2, pp. 307–332, 2003.
- [21] T. Kis and A. Kovács, “On bilevel machine scheduling problems,” *OR spectrum*, vol. 34, no. 1, pp. 43–68, 2012.
- [22] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Handbook on scheduling: from theory to applications*. Springer Science & Business Media, 2007.
- [23] S.-W. Lin and K.-C. Ying, “Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics,” *Omega*, vol. 64, pp. 115–125, 2016.
- [24] C. Salto, “Algoritmos evolutivos avanzados como soporte del proceso productivo,” Ph.D. dissertation, Facultad de Informática, 2000.
- [25] E. L. Lawler, J. K. Lenstra, and A. R. Kan, “Recent developments in deterministic sequencing and scheduling: a survey,” in *Deterministic and stochastic scheduling*. Springer, 1982, pp. 35–73.
- [26] A. Ramos, “Investigación operativa y optimización,” *Universidad Pontificia Comillas*, 1993.

- [27] S. S. Panwalkar and W. Iskander, "A survey of scheduling rules," *Operations research*, vol. 25, no. 1, pp. 45–61, 1977.
- [28] Jiménez, E. Toro, and C. Muñoz, "Solución del problema de flow shop flexible aplicando el algoritmo genético de chu-beasley," *Entre Ciencia e Ingeniería*, 2013.
- [29] R. Gallego, E. Toro, and A. Escobar, *Técnicas Heurísticas y Metaheurísticas*. Colección trabajos de investigación. Universidad Tecnológica de Pereira. Vicerrectoría de Investigaciones Innovación y Extensión. Ingenierías Eléctrica Electrónica Física y Ciencias de la Computación, 2015.
- [30] E. Toro, Y. Restrepo, and M. Granada, "Algoritmo genético modificado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal-flow shop," *Scientia et technica*, 2006.
- [31] R. Gallego, A. Escobar, and E. Toro, "Técnicas metaheurísticas de optimización," *Pereira: Textos Universitarios Universidad Tecnológica de Pereira*, 2008.
- [32] F. Carrasco Díaz, "Modelado y resolución de problemas de secuenciación de tareas: aplicación a una empresa de fabricación de electrodos," 2016.
- [33] C.-Y. Lee, L. Lei, and M. Pinedo, "Current trends in deterministic scheduling," *Annals of operations Research*, vol. 70, pp. 1–41, 1997.
- [34] G. Halevi, *Handbook of production management methods*. Elsevier, 2001.
- [35] G. Narzisi, "Multi-objective optimization, a quick introduction; courant institute of mathematical sciences, new york university: New york, ny, usa, 2008."
- [36] U. S. Bolivar, "Programación de sistemas productivos."
- [37] M. Pinedo and M. Singer, "A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop," *Naval Research Logistics (NRL)*, vol. 46, no. 1, pp. 1–17, 1999.
- [38] R. Aalla, "A heuristic algorithm for flow shop sequencing problems," Ph.D. dissertation, 1992.
- [39] J. H. Restrepo, "Aplicación de la heurística de gupta en la secuenciación de n tareas en m máquinas: un caso de estudio," *Scientia et technica*, vol. 3, no. 49, pp. 69–71, 2011.
- [40] H. G. Campbell, R. A. Dudek, and M. L. Smith, "A heuristic algorithm for the n job, m machine sequencing problem," *Management science*, vol. 16, no. 10, pp. B–630, 1970.

- [41] C. Pascual, R. Vila *et al.*, “El curioso comportamiento del método de inserción de la heurística neh en el problema,” in *6th International Conference on Industrial Engineering and Industrial Management*, 2012, pp. 855–862.
- [42] J. Santiago and D. Magallon, “Critical path method,” *CEE320, Winter 2013*, 2009.
- [43] M. Nawaz, E. E. Enscore Jr, and I. Ham, “A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem,” *Omega*, vol. 11, no. 1, pp. 91–95, 1983.
- [44] A. Janiak, “General flow-shop scheduling with resource constraints,” *International Journal of Production Research*, vol. 26, no. 6, pp. 1089–1103, 1988.
- [45] C. N. Potts, D. B. Shmoys, and D. P. Williamson, “Permutation vs. non-permutation flow shop schedules,” *Operations Research Letters*, vol. 10, no. 5, pp. 281–284, 1991.
- [46] M. Tandon, P. Cummings, and M. LeVan, “Flowshop sequencing with non-permutation schedules,” *Computers & chemical engineering*, vol. 15, no. 8, pp. 601–607, 1991.
- [47] V. A. Strusevich and C. M. Zwaneveld, “On non-permutation solutions to some two machine flow shop scheduling problems,” *Zeitschrift für Operations Research*, vol. 39, no. 3, pp. 305–319, 1994.
- [48] D. Deal, T. Yang, and S. Hallquist, “Job scheduling in petrochemical production: two-stage processing with finite intermediate storage,” *Computers & chemical engineering*, vol. 18, no. 4, pp. 333–344, 1994.
- [49] R. Grau, A. Espuña, and L. Puigjaner, “Environmental considerations in batch production scheduling,” *Computers & chemical engineering*, vol. 19, pp. 651–656, 1995.
- [50] —, “Completion times in multipurpose batch plants with set-up, transfer and clean-up times,” *Computers & chemical engineering*, vol. 20, pp. S1143–S1148, 1996.
- [51] C. Koulamas, “A new constructive heuristic for the flowshop scheduling problem,” *European Journal of Operational Research*, vol. 105, no. 1, pp. 66–71, 1998.
- [52] C. Schwindt and N. Trautmann, “Batch scheduling in process industries: an application of resource-constrained project scheduling,” *OR-Spektrum*, vol. 22, no. 4, pp. 501–524, 2000.

- [53] C. Rajendran and H. Ziegler, “A performance analysis of dispatching rules and a heuristic in static flowshops with missing operations of jobs,” *European Journal of Operational Research*, vol. 131, no. 3, pp. 622–634, 2001.
- [54] A. S. Jain and S. Meeran, “A multi-level hybrid framework applied to the general flow-shop scheduling problem,” *Computers & Operations Research*, vol. 29, no. 13, pp. 1873–1901, 2002.
- [55] S. Pugazhendhi, S. Thiagarajan, C. Rajendran, and N. Anantharaman, “Performance enhancement by using non-permutation schedules in flowline-based manufacturing systems,” *Computers & Industrial Engineering*, vol. 44, no. 1, pp. 133–157, 2003.
- [56] P. Brucker, S. Heitmann, and J. Hurink, “Flow-shop problems with intermediate buffers,” *OR Spectrum*, vol. 25, no. 4, pp. 549–574, 2003.
- [57] C. A. Méndez and J. Cerdá, “An milp continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies,” *Optimization and Engineering*, vol. 4, no. 1-2, pp. 7–22, 2003.
- [58] R. Aggoune, “Minimizing the makespan for the flow shop scheduling problem with availability constraints,” *European Journal of Operational Research*, vol. 153, no. 3, pp. 534–543, 2004.
- [59] S. Pugazhendhi, S. Thiagarajan, C. Rajendran, and N. Anantharaman, “Generating non-permutation schedules in flowline-based manufacturing systems with sequence-dependent setup times of jobs: a heuristic approach,” *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 1-2, pp. 64–78, 2004.
- [60] —, “Relative performance evaluation of permutation and non-permutation schedules in flowline-based manufacturing systems with flowtime objective,” *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 11-12, pp. 820–830, 2004.
- [61] R. Swaminathan, J. W. Fowler, M. E. Pfund, and S. J. Mason, “Minimizing total weighted tardiness in a dynamic flowshop with variable processing times,” in *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 2004, p. 1.
- [62] P. Doganis, H. Sarimveis, G. Bafas, and D. Koufos, “An milp model for optimal scheduling of the lubricant production plant,” *Chemical Engineering Communications*, vol. 192, no. 8, pp. 1067–1084, 2005.
- [63] A. Rossi and M. Lanzetta, “Scheduling flow lines with buffers by ant colony digraph,” *Expert Systems with Applications*, vol. 40, no. 9, pp. 3328–3340, 2013.

- [64] C. Liao, L. Liao, and C. Tseng, “A performance evaluation of permutation vs. non-permutation schedules in a flowshop,” *International Journal of Production Research*, vol. 44, no. 20, pp. 4297–4309, 2006.
- [65] G. Färber and A. M. C. Moreno, “Performance study of a genetic algorithm for sequencing in mixed model non-permutation flowshops using constrained buffers,” in *International Conference on Computational Science and Its Applications*. Springer, 2006, pp. 638–648.
- [66] M.-A. Isenberg and B. Scholz-Reiter, “The multiple batch processing machine problem with stage specific incompatible job families,” in *Dynamics in Logistics*. Springer, 2013, pp. 113–124.
- [67] K.-C. Ying and S.-W. Lin, “Multi-heuristic desirability ant colony system heuristic for non-permutation flowshop scheduling problems,” *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 7-8, pp. 793–802, 2007.
- [68] G. Färber, S. Salhi, and A. M. C. Moreno, “Semi-dynamic demand in a non-permutation flowshop with constrained resequencing buffers,” in *International Conference on Large-Scale Scientific Computing*. Springer, 2007, pp. 536–544.
- [69] R. Swaminathan, M. E. Pfund, J. W. Fowler, S. J. Mason, and A. Keha, “Impact of permutation enforcement when minimizing total weighted tardiness in dynamic flowshops with uncertain processing times,” *Computers & Operations Research*, vol. 34, no. 10, pp. 3055–3068, 2007.
- [70] K.-C. Ying, “Solving non-permutation flowshop scheduling problems by an effective iterated greedy heuristic,” *The International Journal of Advanced Manufacturing Technology*, vol. 38, no. 3-4, p. 348, 2008.
- [71] V. J. Rayward-Smith and D. Rebaïne, “Analysis of heuristics for the uet two-machine flow shop problem with time delays,” *Computers & Operations Research*, vol. 35, no. 10, pp. 3298–3310, 2008.
- [72] S. Sadjadi, M. Aryanezhad, and M. Ziaee, “The general flowshop scheduling problem: mathematical models,” *J. of app. sci*, vol. 8, no. 17, pp. 3032–3037, 2008.
- [73] S.-W. Lin, K.-C. Ying, and Z.-J. Lee, “Metaheuristics for scheduling a non-permutation flowline manufacturing cell with sequence dependent family setup times,” *Computers & operations research*, vol. 36, no. 4, pp. 1110–1121, 2009.
- [74] S.-W. Lin and K.-C. Ying, “Applying a hybrid simulated annealing and tabu search approach to non-permutation flowshop scheduling problems,” *International Journal of Production Research*, vol. 47, no. 5, pp. 1411–1424, 2009.

- [75] V. Nagarajan and M. Sviridenko, "Tight bounds for permutation flow shop scheduling," *Mathematics of Operations Research*, vol. 34, no. 2, pp. 417–427, 2009.
- [76] K.-C. Ying, J. N. Gupta, S.-W. Lin, and Z.-J. Lee, "Permutation and non-permutation schedules for the flowline manufacturing cell with sequence dependent family setups," *International Journal of Production Research*, vol. 48, no. 8, pp. 2169–2184, 2010.
- [77] L.-M. Liao and C.-J. Huang, "Tabu search for non-permutation flowshop scheduling problem with minimizing total tardiness," *Applied Mathematics and Computation*, vol. 217, no. 2, pp. 557–567, 2010.
- [78] J.-t. Li, L.-z. Zhang, C.-x. ShangGuan, and H. Kise, "A ga-based heuristic algorithm for non-permutation two-machine robotic flow-shop scheduling problem of minimizing total weighted completion time," in *2010 IEEE International Conference on Industrial Engineering and Engineering Management*. IEEE, 2010, pp. 1281–1285.
- [79] G. Liberopoulos, G. Kozanidis, and O. Hatzikonstantinou, "Production scheduling of a multi-grade pet resin plant," *Computers & chemical engineering*, vol. 34, no. 3, pp. 387–400, 2010.
- [80] M. Mohammadi, S. Fatemi Ghomi, B. Karimi, and S. Torabi, "Mip-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setups," *International Journal of Production Research*, vol. 48, no. 10, pp. 2957–2973, 2010.
- [81] T. Zheng and M. Yamashiro, "A novel quantum differential evolutionary algorithm for non-permutation flow shop scheduling problems," in *2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*. IEEE, 2010, pp. 357–362.
- [82] G. Farber, A. M. Coves Moreno, and S. Salhi, "Performance evaluation of hybrid-clp vs. ga: non-permutation flowshop with constrained resequencing buffers," *International Journal of Manufacturing Technology and Management*, vol. 20, no. 1-4, pp. 242–258, 2010.
- [83] P. Brucker and N. V. Shakhlevich, "Inverse scheduling: two-machine flow-shop problem," *Journal of Scheduling*, vol. 14, no. 3, pp. 239–256, 2011.
- [84] R. Ramezani, M. M. SAIDI, and D. Rahmani, "Flow shop scheduling problem with missing operations: Genetic algorithm and tabu search," 2011.
- [85] R. Rudek, "Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect," *Computers & Industrial Engineering*, vol. 61, no. 1, pp. 20–31, 2011.

- [86] Y. Mehravaran and R. Logendran, “Non-permutation flowshop scheduling in a supply chain with sequence-dependent setup times,” *International Journal of Production Economics*, vol. 135, no. 2, pp. 953–963, 2012.
- [87] T. E. Cheng, B. M. Lin, and H. Huang, “Resource-constrained flowshop scheduling with separate resource recycling operations,” *Computers & Operations Research*, vol. 39, no. 6, pp. 1206–1212, 2012.
- [88] B. Vahedi-Nouri, P. Fattahi, and R. Ramezani, “Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints,” *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 167–173, 2013.
- [89] B. Vahedi Nouri, P. Fattahi, and R. Ramezani, “Hybrid firefly-simulated annealing algorithm for the flow shop problem with learning effects and flexible maintenance activities,” *International Journal of Production Research*, vol. 51, no. 12, pp. 3501–3515, 2013.
- [90] M. Ziaee, “General flowshop scheduling problem with the sequence dependent setup times: A heuristic approach,” *Information Sciences*, vol. 251, pp. 126–135, 2013.
- [91] Y. Mehravaran and R. Logendran, “Non-permutation flowshop scheduling with dual resources,” *Expert Systems with Applications*, vol. 40, no. 13, pp. 5061–5076, 2013.
- [92] R. Ramezani and M. Saidi-Mehrabad, “Hybrid simulated annealing and mip-based heuristics for stochastic lot-sizing and scheduling problem in capacitated multi-stage production system,” *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5134–5147, 2013.
- [93] L. Shen, J. N. Gupta, and U. Buscher, “Flow shop batching and scheduling with sequence-dependent setup times,” *Journal of scheduling*, vol. 17, no. 4, pp. 353–370, 2014.
- [94] A. Gharbi, M. Labidi, and M. A. Louly, “The nonpermutation flowshop scheduling problem: adjustment and bounding procedures,” *Journal of Applied Mathematics*, vol. 2014, 2014.
- [95] A. Moukrim, D. Rebaine, and M. Serairi, “A branch and bound algorithm for the two-machine flowshop problem with unit-time operations and time delays,” *RAIRO-Operations Research*, vol. 48, no. 2, pp. 235–254, 2014.
- [96] A. Rossi and M. Lanzetta, “Native metaheuristics for non-permutation flowshop scheduling,” *Journal of Intelligent Manufacturing*, vol. 25, no. 6, pp. 1221–1233, 2014.

- [97] A. J. Benavides, M. Ritt, and C. Miralles, “Flow shop scheduling with heterogeneous workers,” *European Journal of Operational Research*, vol. 237, no. 2, pp. 713–720, 2014.
- [98] B. Nikjo and Y. Zarook, “A non-permutation flow shop manufacturing cell scheduling problem with part’s sequence dependent family setup times,” *International Journal of Applied Metaheuristic Computing (IJAMC)*, vol. 5, no. 4, pp. 70–86, 2014.
- [99] B. Vahedi-Nouri, P. Fattahi, R. Tavakkoli-Moghaddam, and R. Ramezani, “A general flow shop scheduling problem with consideration of position-based learning effect and multiple availability constraints,” *The International Journal of Advanced Manufacturing Technology*, vol. 73, no. 5-8, pp. 601–611, 2014.
- [100] M. Babaei, M. Mohammadi, and S. F. Ghomi, “A genetic algorithm for the simultaneous lot sizing and scheduling problem in capacitated flow shop with complex setups and backlogging,” *The International Journal of Advanced Manufacturing Technology*, vol. 70, no. 1-4, pp. 125–134, 2014.
- [101] D. Rahmani, R. Ramezani, and M. Saidi-Mehrabad, “Multi-objective flow shop scheduling problem with stochastic parameters: fuzzy goal programming approach,” *International Journal of Operational Research*, vol. 21, no. 3, pp. 322–340, 2014.
- [102] Y. Xiao, Y. Yuan, R.-Q. Zhang, and A. Konak, “Non-permutation flow shop scheduling with order acceptance and weighted tardiness,” *Applied Mathematics and Computation*, vol. 270, pp. 312–333, 2015.
- [103] H. Amirian and R. Sahraeian, “Augmented ε -constraint method in multi-objective flowshop problem with past sequence set-up times and a modified learning effect,” *International Journal of Production Research*, vol. 53, no. 19, pp. 5962–5976, 2015.
- [104] A. J. Benavides and M. Ritt, “Iterated local search heuristics for minimizing total completion time in permutation and non-permutation flow shops,” in *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.
- [105] —, “Two simple and effective heuristics for minimizing the makespan in non-permutation flow shops,” *Computers & Operations Research*, vol. 66, pp. 160–169, 2016.
- [106] X. Cui, “Calisthenics with words: The effect of readability and investor sophistication on investors’ performance judgment,” *International Journal of Financial Studies*, vol. 4, no. 1, p. 1, 2016.

- [107] M. Henneberg and J. Neufeld, “A constructive algorithm and a simulated annealing approach for solving flowshop problems with missing operations,” *International Journal of Production Research*, vol. 54, no. 12, pp. 3534–3550, 2016.
- [108] D. Rossit, F. Tohmé, M. Frutos, J. Bard, and D. Broz, “A non-permutation flowshop scheduling problem with lot streaming: A mathematical model,” *International Journal of Industrial Engineering Computations*, vol. 7, no. 3, pp. 507–516, 2016.
- [109] B. RAMIREZ, “El sistema productivo del job shop en el salón de clase,” *Trabajo de grado para obtención de título de Ingeniera Industrial. Pereira: Universidad Tecnológica de Pereira. Facultad de Ingeniería Industrial*, pp. 1–39, 2007.
- [110] H. Ford, *My life and work*. Cosimo, Inc., 2007.