

Universidad Autónoma de Coahuila

Facultad de Sistemas



## TESIS

Minimizando el tiempo total de terminación en un problema de secuenciación de tareas en máquinas paralelas

Presentada por

Antonio de Jesús Estrada Vásquez

Para obtener el grado de

Ingeniero Industrial y de Sistemas

Universidad Autónoma de Coahuila

Facultad de Sistemas



## TESIS

Minimizando el tiempo total de terminación en un problema de secuenciación de tareas en máquinas paralelas

Presentada por

Antonio de Jesús Estrada Vásquez

Para obtener el grado académico de

Ingeniero Industrial y de Sistemas

Director de tesis:

Dr. Oliver Ávalos Rosales

Codirector de tesis:

Dra. Yajaira Cardona Valdés

# Índice general

<b>1. Introducción</b>	<b>8</b>
1.1. Descripción del problema . . . . .	8
1.2. Relevancia del problema . . . . .	8
1.3. Antecedentes . . . . .	8
1.4. Objetivos . . . . .	10
1.5. Metodología . . . . .	10
1.6. Contribuciones . . . . .	10
1.7. Organización del documento . . . . .	11
<b>2. Marco teórico</b>	<b>12</b>
2.1. Investigación de operaciones . . . . .	12
2.2. Áreas de IO determinista . . . . .	15
2.3. Secuenciación de tareas . . . . .	18
2.3.1. Ambientes de producción . . . . .	18
2.3.2. Características de las tareas . . . . .	19
2.3.3. Medidas de rendimiento y objetivos . . . . .	20
2.3.4. Métodos heurísticos en secuenciación . . . . .	22
<b>3. Metodología de solución</b>	<b>23</b>
3.1. Descripción del problema . . . . .	23
3.1.1. Cálculo del tiempo total de terminación . . . . .	25
3.2. Formulación propuesta . . . . .	26
3.3. Algoritmos heurísticos . . . . .	28
3.3.1. Formas de selección . . . . .	28
3.3.2. Formas de evaluación . . . . .	28
3.3.3. Constructivos . . . . .	29
<b>4. Experimentación</b>	<b>34</b>
4.1. Instancias . . . . .	34
4.2. Desempeño del modelo . . . . .	35
4.3. Desempeño de las heurísticas . . . . .	36
4.3.1. En instancias pequeñas y medianas . . . . .	36
4.3.2. En instancias grandes . . . . .	38
<b>5. Conclusiones y trabajo futuro</b>	<b>41</b>

# Índice de figuras

3.1. Tiempos de procesamiento de la instancia de ejemplo . . . . .	24
3.2. Tiempos de preparación de la instancia del ejemplo . . . . .	24
3.3. Representación gráfica de la solución del ejemplo con 2 máquinas y 6 tareas . . .	24
3.4. Red multinivel usada como auxiliar en la modelación matemática del problema .	27

# Índice de tablas

4.1. Desempeño del modelo y de su relajación lineal para resolver las instancias medianas minimizando el TCT . . . . .	35
4.2. Resultados de las heurísticas en $n$ instancias pequeñas con parámetro de $a = 2$ .	36
4.3. Resultados de las heurísticas en $n$ instancias pequeñas con parámetro de $a = 4$ .	37
4.4. Resultados de las heurísticas en $n$ instancias medianas con parámetro de $a = 2$ .	37
4.5. Resultados de las heurísticas en $n$ instancias medianas con parámetro de $a = 4$ .	38
4.6. Resultados de las heurísticas en $n$ instancias grandes con parámetro de $a = 2$ . .	39
4.7. Resultados de las heurísticas en $n$ instancias grandes con parámetro de $a = 4$ . .	40

# Agradecimientos

A la Facultad de Sistemas, así como los maestros que me dieron los conocimientos base para poder entender y llevar acabo la tesis.

A la maestra Vanesa Avalos por ser mi maestra en la materia base para este tema y por ser quien me recomendó para la elaboración de este proyecto.

Al CIMA por brindarme el equipo e instalaciones necesarias para poder trabajar en ello.

A PRODEP por proporcionarme la beca para poder continuar económicamente sustentando los gastos necesarios en la tesis.

Al Doctor Oliver por ser mi tutor y estar conmigo todo el tiempo ayudandome a desarrollar la programación y experimentación.

A la Doctora Yajaira por la supervisión y ayuda en la redacción de la tesis.

A profesores, Irma García, Miguel Ángel Zamora y Diana Salas por aceptar ser parte del jurado de tesis y ayudar a mejorar y enriquecer el contenido de la misma.

A mis padres por haberme brindado todo el apoyo y recursos necesarios para terminar mis estudios y llegar hasta este punto de mi vida.

# Resumen

En esta tesis se plantea un problema de secuenciación de máquinas en paralelo, donde cada tarea tiene un tiempo de procesamiento y existe un tiempo de preparación de las máquinas entre cada par de tareas que son procesadas. La función objetivo a minimizar es el tiempo total de terminación, la cual es una medida de desempeño poco estudiada en este ambiente de producción y con importantes implicaciones económicas en ambientes de producción.

Modelamos matemáticamente el problema y con un optimizador comercial encontramos soluciones óptimas dentro de un tiempo límite de ejecución de una hora. También diseñamos e implementamos diferentes heurísticas para obtener soluciones para instancias del problema que el optimizador comercial no es capaz de resolver en el tiempo límite establecido. Realizamos experimentación computacional tanto para el modelo como para las diferentes heurísticas en las que determinamos su desempeño y alcance, y analizamos las características que los favorecen para ofrecer mejores resultados.

# Capítulo 1

## Introducción

### 1.1. Descripción del problema

Un área importante de la aplicación de la ingeniería industrial recae en la optimización en los sistemas de producción y uno de gran importancia es el de máquinas en paralelo. En sistemas de producción existen diferentes medidas para evaluar el desempeño del sistema, una de interés en términos de los beneficios económicos que se obtienen por la reducción de inventarios en piso, es el tiempo total de terminación.

En esta tesis se aborda un problema de secuenciación de tareas en máquinas en paralelo, donde cada tarea tiene un tiempo de procesamiento que depende de la máquina en la que se procese y hay un tiempo de preparación de las máquinas que depende tanto de la tarea que se termina de procesar como de la tarea que entra a procesamiento. Las máquinas están continuamente disponibles y pueden procesar solo un trabajo a la vez sin interrupción. No hay restricciones de precedencia entre los trabajos y la función objetivo a minimizar es el tiempo total de terminación.

### 1.2. Relevancia del problema

Las aplicaciones que involucran problemas de secuenciación con tiempos de preparación son relevantes, tal como lo muestra Allahverdi y Soroush (2008), quienes resumen diferentes tipos de aplicaciones donde se han considerado los tiempos de preparación.

En problemas de secuenciación de tareas, el tiempo total de terminación (TCT) es importante de acuerdo con los beneficios económicos que se obtienen en ambientes de producción cuando éste es minimizado, sin embargo, esta función objetivo ha sido poco estudiada, especialmente en máquinas individuales, máquinas paralelas y *Job Shop*, según se puede observar en la revisión de los problemas de secuenciación con tiempos de preparación que realizan (Allahverdi et al., 1999, 2008; Allahverdi, 2015) desde mediados de los años 60 hasta el año 2015.

El problema es NP-hard y no ha sido previamente estudiado en la literatura científica.

### 1.3. Antecedentes

En nuestra revisión de literatura nos enfocamos en trabajos que presentan formulaciones lineales enteras mixtas (MIP Mixed Integer Programming por sus siglas en inglés) y algoritmos



exactos para la minimización del tiempo total de terminación. Usualmente cuando los autores presentan un algoritmo exacto no informan el tamaño de las instancias resueltas por el modelo MIP puro. Al final presentamos los pocos trabajos que encontramos en el desarrollo de heurísticas para la minimización del tiempo de terminación ponderado en máquinas paralelas.

En el problema de una máquina con tiempos de preparación y minimización del tiempo total de terminación, conocido en el contexto de ruteo de vehículos como: el problema del reparador, problema del repartidor ó problema de mínima latencia, Lucena (1990) desarrolló un procedimiento enumerativo basado en una formulación entera no lineal utilizando una relajación Lagrangiana para obtener cotas inferiores. El autor resolvió instancias de hasta 30 nodos. Fischetti et al. (1993) propusieron un modelo MIP utilizando restricciones de orden para la eliminación de subtours y desarrollaron un algoritmo de ramificación y acotamiento utilizando una estructura de matroides del problema para obtener cotas inferiores, ellos resolvieron instancias de hasta 60 nodos. Méndez-Díaz et al. (2008) propusieron un modelo MIP aprovechando las conexiones con el problema de ordenamiento lineal, presentaron un algoritmo de planos cortantes que usa desigualdades válidas asociadas con la formulación propuesta.

Otros investigadores han desarrollado formulaciones basadas en las propuestas por Picard y Queyranne (1978) y Queyranne y Wang (1991). Gouveia y Voß (1995) propusieron dos formulaciones que usan variables de asignación y muestran mejores cotas inferiores que otros modelos desarrollados para el problema del reparador. Angel-Bello et al. (2013) proponen dos reformulaciones. Los autores se dieron cuenta de que el número de variables binarias podría reducirse al no eliminar un conjunto de restricciones como Gouveia y Voß (1995) habían considerado, lo cual le dió al modelo una ligera mejora. En el contexto de secuenciación de tareas, Bigras et al. (2008) consideran el modelo de Picard y Queyranne (1978) y desarrollan un algoritmo de planos cortantes para obtener cotas inferiores e implementar un algoritmo de ramificación y acotamiento. Usando el algoritmo, resuelven instancias de hasta 50 trabajos. Las mejores formulaciones son capaces de resolver instancias generales de hasta 30 trabajos e instancias que satisfacen la desigualdad triangular de hasta 40 trabajos.

En el problema de máquinas paralelas idénticas con tiempos de preparación y minimizando el tiempo total de terminación, conocido en el contexto de ruteo de vehículos como: problema de múltiples reparadores, problema de múltiples repartidores y problema de mínima latencia múltiple, Nucamendi-Guillén et al. (2016) proponen una formulación MIP y un enfoque de solución basado en el metaheurístico *iterated greedy*, mientras que Angel-Bello et al. (2017) analizan varias formulaciones diferentes. Para este problema las mejores formulaciones resuelven instancias de hasta 70 trabajos (al menos dos máquinas). Fowler et al. (2003) proponen un algoritmo genético híbrido para el problema con la misma función objetivo y considera además fechas de liberación. Nessah et al. (2005) consideran fechas de liberación y la función objetivo del tiempo total de terminación, ellos proponen cotas inferiores y una heurística basada en condiciones de optimalidad.

En el caso de máquinas paralelas no relacionadas y la función objetivo del tiempo de terminación ponderado, Weng et al. (2001) presentan siete heurísticas simples. La mejor heurística asigna un trabajo basado en la proporción más pequeña entre el tiempo de procesamiento más el tiempo de preparación y su peso.

Los algoritmos desarrollados para el caso de una máquina y máquinas idénticas, no son directamente aplicables cuando se consideran máquinas paralelas no relacionadas, sin embargo algunas de sus características pueden ser útiles en el diseño de un algoritmo que sí las considere. Adicionalmente un algoritmo que se desarrolle para máquinas no relacionadas puede compararse

con aquellos diseñados para máquinas idénticas. Por otro lado, los algoritmos desarrollados para funciones objetivo más complicadas pueden desaprovechar características que los hagan más eficientes en funciones particulares como el tiempo total de terminación.

Es claro entonces que no hay en la literatura, modelo matemático ni algoritmos heurísticos que aprovechen al máximo las particularidades de la función objetivo del tiempo total de terminación en el ambiente de producción de máquinas paralelas no relacionadas.

## 1.4. Objetivos

El objetivo general de este trabajo es modelar matemáticamente y proponer algoritmos de solución, que proporcionen soluciones de buena calidad en tiempos de cómputo razonables, para el problema de secuenciación de tareas en máquinas paralelas no relacionadas con tiempos de preparación dependientes de la secuencia y la minimización del tiempo total de terminación.

Los objetivos particulares son los siguientes:

- Revisión de la literatura científica para identificar características prometedoras en algoritmos para problemas de secuenciación de tareas.
- Obtener un modelo matemático que permita encontrar soluciones óptimas del problema.
- Desarrollar un generador de instancias de tamaño mediano que permitan determinar el alcance del modelo.
- Desarrollar algoritmos heurísticos de características variadas para proveer soluciones en instancias que el optimizador comercial no es capaz de resolver utilizando el modelo en un tiempo límite de una hora.
- Comparar el desempeño de los algoritmos heurísticos para determinar cuáles características conducen a un mejor desempeño en términos de la calidad de las soluciones obtenidas y tiempos de ejecución.

## 1.5. Metodología

Describiremos un modelo matemático entero mixto para el problema y dada la complejidad del problema estudiado será necesario desarrollar algoritmos heurísticos para instancias que que el optimizador comercial no sea capaz de resolver utilizando el modelo en un tiempo límite de una hora.

Proponemos algoritmos heurísticos basados en las conocidas reglas de prioridad de *tiempo procesamiento más largo*, o *LPT* y *tiempo de procesamiento más corto*, o *SPT*, altamente utilizados en problemas de producción y otros basados en estrategias recientes, surgidas como alternativa a los constructivos tipo GRASP.

## 1.6. Contribuciones

- Se estudia por primera vez el problema de máquinas paralelas, considerando de manera específica la minimización del tiempo total de terminación.

- Se plantea el modelo matemático para el problema basado en un tipo de variables poco utilizado en problemas relacionados.
- Se contribuye con experimentación computacional que muestra el desempeño y alcance del modelo matemático.
- Se contribuye con el diseño y análisis de características de diferentes tipos de heurísticas.
- Se contribuye con la experimentación computacional que determina cuál es la heurística más eficiente y se analiza cuáles pueden ser las razones.

## 1.7. Organización del documento

En el capítulo 2 se describen los métodos y problemas clásicos de la investigación de operaciones. También se describen los conceptos y notación necesarios para el estudio y entendimiento del problema, así como algoritmos frecuentes que son utilizados en problemas de secuenciación.

En el capítulo 3 se define formalmente el problema que se trabajará, se presenta la formulación matemática del mismo, se realiza un pequeño análisis de las diferentes formas en que se puede evaluar la función objetivo y se describen los algoritmos heurísticos propuestos.

En el capítulo 4 se describen las características de las instancias usadas en la experimentación y se presentan los resultados de los diferentes experimentos que ayudan a determinar el desempeño tanto del modelo como de las heurísticas propuestas.

Finalmente, en el capítulo 5 se describen las conclusiones generales y se proponen futuras líneas de investigación que se derivan de esta tesis.

# Capítulo 2

## Marco teórico

En este capítulo se dará una pequeña introducción a la investigación de operaciones (secciones 2.1 y 2.2), dicha información es tomada del libro “Investigación de operaciones” de Taha (2012). Posteriormente, en la sección 2.3 describiremos características y notación empleada en problemas de secuenciación de tareas, dicha información puede ser consultada en Pinedo (2016). También discutiremos brevemente qué son las heurísticas y cuáles son las reglas de prioridad usadas frecuentemente en problemas de secuenciación de tareas.

### 2.1. Investigación de operaciones

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.

Imagine que tiene un compromiso de negocios que requiere 5 semanas de traslado continuo entre Chihuahua (CH) y Veracruz (VC). Sale de Chihuahua los lunes y regresa los miércoles. Un boleto regular de viaje redondo cuesta \$400, pero se ofrece 20% de descuento si el viaje redondo comprende un fin de semana. Un boleto sencillo en cualquier dirección cuesta 75% del precio regular. ¿Cómo debe comprar los boletos para reducir el costo del traslado durante las 5 semanas?

Podemos considerar la situación como un problema de toma de decisiones, cuya solución requiere responder tres preguntas:

1. ¿Cuáles son las alternativas de decisión?
2. ¿Conforme a qué restricciones se toma la decisión?
3. ¿Cuál es el criterio objetivo apropiado para evaluar las alternativas?

Se consideran tres alternativas razonables:

1. Comprar cinco boletos normales CH-VC-CH para salir el lunes y regresar el miércoles de la misma semana.

2. Comprar un boleto CH-VC, cuatro VC-CH-VC que abarquen fines de semana, y uno VC-CH.
3. Comprar un boleto CH-VC-CH para el lunes de la primera semana y el miércoles de la última semana, y cuatro VC-CH-VC para los viajes restantes. Todos los boletos en esta alternativa cubren por lo menos un fin de semana.

La restricción en estas opciones es que pueda salir de CH el lunes y regresar el miércoles de la misma semana.

Un criterio objetivo obvio para evaluar la alternativa propuesta es el precio de los boletos. La alternativa que dé el costo mínimo será la mejor. Específicamente, tenemos:

Costo de la alternativa 1:  $5 \times 400 = \$2000$

Costo de la alternativa 2:  $.75 \times 400 + 4 \times (.8 \times 400) + .75 \times 400 = \$1880$

Costo de la alternativa 3:  $5 \times (.8 \times 400) = \$1600$

La alternativa 3 es la mejor porque es la más económica. Aunque el ejemplo anterior ilustra los tres componentes principales de un modelo de *IO*, los cuales son: alternativas, criterio objetivo y restricciones, las situaciones difieren por los detalles de la construcción de cada componente y la solución del modelo resultante. Para ilustrar este punto, considere la formación de un rectángulo de área máxima con un trozo de alambre de  $L$  pulgadas de longitud. ¿Cuál será el mejor y altura del rectángulo?

En contraste con el ejemplo de los boletos, el número de alternativas en este ejemplo no es finito; es decir, el ancho y la altura del rectángulo pueden asumir una cantidad infinita de valores porque son variables continuas. Para formalizar esta observación, las alternativas del problema se identifican definiendo el ancho y la altura como variables algebraicas

$w$  = ancho del rectángulo en pulgadas,

$h$  = altura del rectángulo en pulgadas.

Con base en estas definiciones, las restricciones de la situación pueden expresarse verbalmente como:

1. Ancho del rectángulo, la altura del rectángulo a la mitad de la longitud del alambre.
2. El ancho y la altura no pueden ser negativos.

Estas restricciones se traducen de manera algebraica como sigue:

1.  $2(w + h) = L$

2.  $w \geq 0, h \geq 0$

Ahora el único componente restante es el objetivo del problema, es decir, maximizar el área del rectángulo. Si  $z$  se define como el área del rectángulo, el modelo completo es:

$$\begin{aligned} &\text{Maximizar } z = wh \\ &\text{sujeto a} \\ &2(w + h) = L \\ &w, h \geq 0 \end{aligned}$$

Utilizando cálculo diferencial, la mejor solución de este modelo es  $w = h = L/4$ , la cual requiere la construcción de una forma cuadrática. Con los datos de los dos ejemplos anteriores, el modelo general de *IO* se organiza en el siguiente formato general:

$$\begin{aligned} &\text{Maximizar (o minimizar) la función objetivo} \\ &\text{Sujeto a: un conjunto de restricciones} \end{aligned}$$

## Solución del modelo IO

La técnica de *IO* más importante es la programación lineal. Está diseñada para modelos con funciones objetivo y restricciones lineales. Otras técnicas incluyen la programación entera (en la cual las variables asumen valores enteros), la programación dinámica (en la cual el modelo original puede descomponerse en subproblemas más pequeños y manejables), la programación de red (en la cual el problema puede modelarse como una red), y la programación no lineal (en la cual las funciones del modelo son no lineales). Éstas son sólo algunas de las muchas herramientas de *IO* con que se cuenta.

Una peculiaridad de la mayoría de las técnicas de *IO* es que por lo general las soluciones no se obtienen en formas cerradas (como si fueran fórmulas), sino que más bien se determinan mediante algoritmos. Un algoritmo proporciona reglas fijas de cálculo que se aplican en forma repetitiva al problema y cada repetición (llamada iteración) acerca la solución al óptimo. Como los cálculos asociados con cada iteración suelen ser tediosos y voluminosos, es recomendable que estos algoritmos se ejecuten con la computadora.

Algunos modelos matemáticos pueden ser tan complejos que es imposible resolverlos con cualquiera de los algoritmos de optimización disponibles. En esos casos quizá sea necesario abandonar la búsqueda de la solución óptima y buscar una buena solución aplicando reglas empíricas, o bien métodos heurísticos o metaheurísticos.

## Fases de un estudio de IO

Los estudios de investigación de operaciones se basan en la labor de equipo, donde los analistas de *IO* y el cliente trabajan codo con codo. Los conocimientos de modelado de los analistas de *IO* se deben complementar con la experiencia y cooperación del cliente para quien realizan el estudio.

Para implementar la *IO* en la práctica, las fases principales son:

### 1. Definición del problema:

Implica definir el alcance del problema investigado. Esta función debe ser realizada por todo el equipo de *IO*. El objetivo es identificar tres elementos principales del problema de decisión: (1) descripción de las alternativas de decisión; (2) determinación del objetivo

del estudio, y (3) especificación de las limitaciones bajo las cuales funciona el sistema modelado

#### 2. **Construcción del modelo:**

Implica un intento de transformar la definición del problema en relaciones matemáticas. Si el modelo resultante se ajusta a uno de los modelos matemáticos estándar, como la programación lineal, se suele obtener una solución utilizando los algoritmos disponibles. Por otra parte, si las relaciones matemáticas son demasiado complejas como para permitir la determinación de una solución analítica, el equipo de *IO* puede optar por simplificar el modelo y utilizar un método heurístico, o bien considerar la simulación, si es lo apropiado. En algunos casos, una simulación puede combinarse con modelos heurísticos para resolver el problema de decisión.

#### 3. **Solución del modelo:**

Es por mucho la más sencilla de todas las fases de *IO* porque implica el uso de algoritmos de optimización bien definidos. Un aspecto importante de la fase de solución del modelo es el análisis de sensibilidad. Tiene que ver con la obtención de información adicional sobre el comportamiento de la solución óptima cuando el modelo experimenta algunos cambios de parámetros. El análisis de sensibilidad es particularmente necesario cuando no se pueden estimar con precisión los parámetros del modelo. En estos casos es importante estudiar el comportamiento de la solución óptima en el entorno de los parámetros estimados

#### 4. **Validación del modelo:**

Comprueba si el modelo propuesto hace en realidad lo que dice que hace, es decir, ¿predice adecuadamente el comportamiento del sistema que se estudia? Al principio, el equipo de *IO* debe estar convencido de que el resultado del modelo no contenga “sorpresas”. En otras palabras, ¿tiene sentido la solución? ¿los resultados son intuitivamente aceptables? Del lado formal, un método común de comprobar la validez de un modelo es comparar su resultado con resultados históricos. El modelo es válido si, en condiciones de datos de entrada iguales, reproduce de forma razonable el desempeño pasado. Sin embargo, no suele haber seguridad de que el desempeño futuro continuará copiando el comportamiento pasado, además, como el modelo se basa en el examen cuidadoso de datos pasados, la comparación propuesta casi siempre es favorable. Si el modelo propuesto representara un sistema nuevo (inexistente), no habría datos históricos disponibles. En esos casos podemos utilizar la simulación como una herramienta independiente para comprobar el resultado del modelo matemático.

#### 5. **Implementación de la solución:**

Implica la transformación de los resultados en instrucciones de operación comprensibles que se emitirán a las personas que administrarán el sistema recomendado. La responsabilidad de esta tarea recae principalmente en el equipo de *IO*.

## 2.2. Áreas de IO determinista

### Programación lineal

El objetivo de un modelo matemático es reproducir la realidad de la forma más fiel posible a fin de entender cómo se comporta y poder obtener respuestas a determinadas acciones.

La programación lineal es un tipo de modelo matemático que se desarrolló a partir de la Segunda Guerra Mundial para resolver cierto tipo de problemas de asignación de recursos entre distintas actividades.

En el proceso de formulación de un modelo de programación lineal hay que dar los siguientes pasos:

**1. Determinación de las variables de decisión:**

Representan los elementos del sistema a modelar que son controlables por el decisor. En los modelos lineales continuos estas variables toman como valores números reales y se representan por letras con subíndices como se acostumbra a hacer con las variables matemáticas, o literales alusivos a su significado: peso, valor, etc. En el primer caso también se utiliza la representación como vector de un conjunto indexado de variable:

$$x = (x_1, x_2, \dots, x_n)$$

**2. Determinación de las restricciones:**

Representan las limitaciones prácticas de determinados recursos o imposiciones físicas de la realidad. Se expresan como ecuaciones e inecuaciones lineales de las variables de decisión. Matemáticamente adoptan una de las siguientes formas:

$$g_i(x) \geq b_i; \quad g_i(x) \leq b_i; \quad g_i(x) = b_i; \quad i = 1, \dots, m$$

con  $g_i$ , una función lineal en  $x$

**3. Formulación de la función objetivo:** Se trata de la función que mide la calidad de la solución y que hay que optimizar (maximizar un beneficio o minimizar un costo). También es una función lineal de todas o parte de las variables de decisión.

$$\text{Maximizar } z = f(x); \text{ Minimizar } z = f(x)$$

## Modelo de redes

Optimización de redes es un tipo especial de modelo en programación lineal. Los modelos de redes tienen tres ventajas importantes con respecto a la programación lineal.

1. Pueden resolverse muy rápidamente. Problemas que con programación lineal tendrían 1000 filas y 30,000 columnas pueden ser resueltos en segundos. Esto permite que los modelos de redes sean usados en muchas aplicaciones (tal como la toma de decisión en tiempo real) para lo cual, la programación lineal no es la ideal.
2. Requieren en forma natural de soluciones enteras. Al reconocer que un problema puede formularse como algún modelo de red nos permitirá resolver tipos especiales de problemas de programación entera aumentando la eficiencia y reduciendo el tiempo consumido por los algoritmos clásicos de programación lineal.
3. Son intuitivos. Los modelos de redes proveen un lenguaje para tratar los problemas, mucho más intuitivo que “variables, objetivo y restricciones”.



Obviamente los modelos de redes no son capaces de cubrir la amplia gama de problemas que puede resolver la programación lineal. Sin embargo, ellos ocurren con suficiente frecuencia como para ser considerados como una herramienta importante para la real toma de decisiones.

### Terminología:

1. Una red o grafo consiste de puntos, y líneas que conectan pares de puntos. Los puntos se llaman nodos o vértices. Las líneas se llaman arcos. Los arcos pueden tener una dirección asociada, en cuyo caso se denominan arcos dirigidos. Si un arco no tiene dirección normalmente se le denomina rama. Si todos los arcos en la red son dirigidos, la red se denomina una red dirigida. Si todos los arcos son no-dirigidos, la red es una red no-dirigida.
2. Dos nodos pueden estar conectados por un conjunto de arcos. Una trayectoria (*path* en inglés) es una secuencia de arcos distintos (con nodos no repetidos) conectando a los nodos. Una trayectoria dirigida desde nodo  $i$  al nodo  $j$  es una secuencia de arcos, cada uno de los cuales apunta al nodo  $j$  (si es que hay dirección). Una trayectoria no dirigida puede incluir arcos dirigidos apuntando en cualquier dirección.
3. Una trayectoria que comienza y que termina en el mismo nodo se denomina ciclo y puede ser dirigida o no-dirigida.
4. Una red está conectada si existe una trayectoria no-dirigida entre cualquier par de nodos. Una red conectada que no tiene ciclos se denomina árbol.

## Programación lineal entera

Por lo general, las aplicaciones de programación lineal entera caen dentro de dos categorías: directa y transformada. En la categoría directa, la naturaleza de la situación impide la asignación de valores fraccionarios a las variables del modelo. Por ejemplo, el problema puede implicar la determinación de si se emprende o no un proyecto (variable binaria), o la determinación del número óptimo de máquinas necesarias para realizar una tarea (variable general entera).

En la categoría transformada se utilizan variables enteras auxiliares para convertir analíticamente situaciones insolubles en modelos que pueden resolverse por medio de algoritmos de optimización disponibles. Por ejemplo, en la secuencia de dos trabajos,  $A$  y  $B$ , en una sola máquina, el trabajo  $A$  puede preceder al trabajo  $B$  o viceversa. La naturaleza “ $o$ ” de las restricciones es lo que hace al problema analíticamente insoluble, porque todos los algoritmos de programación matemáticos tratan con sólo restricciones “ $y$ ”.

Por comodidad, un problema se define como programa entero “puro” cuando todas las variables son enteras. En caso contrario, es un programa entero “combinado” que implica una combinación de variables enteras y continuas.

## Heurísticas

La heurística está diseñada para encontrar buenas soluciones aproximadas de problemas combinatorios difíciles que de lo contrario no pueden resolverse mediante los algoritmos de

optimización disponibles.

Una heurística es una técnica de búsqueda directa que utiliza reglas favorables prácticas para localizar buenas soluciones. La ventaja de la heurística es que en general determina (buenas) soluciones con rapidez, utilizando reglas de solución simples. La desventaja es que la calidad de la solución (con respecto a la óptima) suele desconocerse.

Una heurística es un método basado en la experiencia que puede utilizarse como ayuda para resolver problemas de diseño, desde calcular los recursos necesarios hasta en planear las condiciones de operación de los sistemas. En general, deben considerarse como ayudas o apoyos para hacer estimaciones rápidas y diseños preliminares.

Existen heurísticas de tipo Constructivas y heurísticas conocidas como De mejora. Las primeras buscan construir paso a paso una solución, mientras que las segundas, parten de una solución y se basan en la regla de búsqueda codiciosa que dicta que se mejore el valor de la función objetivo con cada movimiento de búsqueda, la búsqueda termina en un óptimo local donde ya no son posibles más mejoras.

## 2.3. Secuenciación de tareas

La secuenciación de tareas es un proceso de toma de decisiones que se utiliza de forma regular en muchas industrias de manufactura y servicios. Se ocupa de la asignación de recursos a tareas en períodos de tiempo determinados y se busca optimizar una o más funciones objetivo.

Los recursos (máquinas) pueden ser máquinas en un taller, pistas en un aeropuerto, cuadrillas en un sitio de construcción, unidades de procesamiento en un entorno informático, etc.

Las tareas pueden ser operaciones en un proceso de producción, despegues y aterrizajes en un aeropuerto, etapas en un proyecto de construcción, ejecuciones de programas de computación, y más.

Los problemas de secuenciación de tareas los podemos denotar y clasificar de acuerdo a 3 secciones diferentes, conocidas como  $\alpha|\beta|\gamma$ , donde  $\alpha$  son los ambientes de producción,  $\beta$  son las características de las tareas y  $\gamma$  es la función objetivo. La mayor parte del contenido de esta sección fue tomada del libro de secuenciación de Pinedo (2016).

### 2.3.1. Ambientes de producción

**Ambiente de una sola máquina:** Los ambientes de una sola máquina ( $\alpha = 1$ ) han sido analizados minuciosamente bajo todo tipo de condiciones y con muchas funciones objetivo diferentes. El resultado es una colección de reglas que, si bien son fáciles de identificar y aplicar, a menudo brindan soluciones óptimas en el entorno de una sola máquina. Por ejemplo, se ha demostrado que la regla de la fecha de vencimiento más temprana (EDD, Earliest Due Date por sus siglas en inglés), que ordena los trabajos en orden creciente de sus fechas de vencimiento, minimiza la demora máxima entre todos los trabajos. Se ha demostrado que la regla de tiempo de procesamiento más corto primero (SPT, Shortes Processing Times, por sus siglas en inglés) minimiza el número promedio de trabajos en espera de procesamiento.

**Ambiente de máquinas paralelas:** Los ambientes de máquina paralela son importantes por la misma razón que los modelos de una sola máquina son importantes: si un centro de trabajo en particular es un cuello de botella, entonces el horario en ese centro de trabajo determinará el rendimiento de todo el sistema. Ese cuello de botella puede modelarse como

un banco de máquinas paralelas y analizarse por separado. Las máquinas pueden ser todas idénticas ( $\alpha = P$ ), deferir su velocidad de procesamiento por un factor ( $\alpha = Q$ ), o en otro caso no estar relacionadas las velocidades de procesamiento ( $\alpha = R$ ). En ocasiones, algunas de las máquinas en paralelo pueden ser más antiguas y operar a menor velocidad; o, algunas máquinas pueden mantenerse mejor y ser capaces de realizar trabajos de mayor calidad. Si ese es el caso, entonces algunos trabajos pueden procesarse en cualquiera de las  $m$  máquinas, mientras que otros trabajos pueden procesarse solo en subconjuntos específicos de las  $m$  máquinas. Cuando las “máquinas” son personas, entonces el tiempo de procesamiento de una operación puede depender tanto del trabajo como de la persona u operador.

**Ambiente *flow shop*:** En muchas configuraciones de fabricación y ensamblaje, los trabajos tienen que someterse a operaciones múltiples en varias máquinas diferentes. Si las rutas de todos los trabajos son idénticas, es decir, todos los trabajos visitan las mismas máquinas en la misma secuencia, se hace referencia al entorno como un *flow shop*. Las máquinas se configuran en serie y cada vez que un trabajo completa su procesamiento en una máquina, se une a la cola en la siguiente máquina. La secuencia de trabajo puede variar de una máquina a otra, ya que los trabajos pueden volver a secuenciarse entre máquinas. Sin embargo, si hay un sistema de manejo de materiales que transporta los trabajos de una máquina a la siguiente, entonces la misma secuencia de trabajo se mantiene en todo el sistema. Una generalización del *flow shop*, consiste en una serie de etapas en serie, en la que cada etapa dispone de una cantidad de máquinas en paralelo.

**Ambiente *job shop*:** Los ambiente *job shop* asumen que un trabajo sigue su propia ruta predeterminada a través del sistema de máquinas.

### 2.3.2. Características de las tareas

**Tiempo de procesamiento ( $p_{ij}$ ):** El tiempo de procesamiento  $p_{ij}$  representa el tiempo que la tarea  $j$  debe pasar en la máquina  $i$ . El subíndice  $i$  se omite si el tiempo de procesamiento del trabajo  $j$  no depende de la máquina o si sólo necesita procesarse en una máquina. Si hay un número de trabajos idénticos que necesitan un tiempo de procesamiento  $p_j$  en una máquina, entonces nos referimos a este conjunto de trabajos como elementos de tipo  $j$ . La velocidad de producción de la máquina de elementos tipo  $j$  se denota por  $q_j = 1/p_j$  (número de elementos por unidad de tiempo).

**Fecha de liberación ( $r_j$ ):** La fecha de liberación  $r_j$  del trabajo  $j$  también se conoce como la fecha de disponibilidad. Es el momento en que el trabajo llega al sistema, es decir, el momento más temprano en el que el trabajo  $j$  puede comenzar su procesamiento.

**Fecha de vencimiento:** La fecha de vencimiento  $d_j$  del trabajo  $j$  representa la fecha de envío comprometida o de finalización (la fecha en que le prometió el trabajo al cliente). Se permite completar un trabajo después de su fecha de vencimiento, pero luego se incurre en una multa. Cuando la fecha de vencimiento debe cumplirse absolutamente, se la conoce como fecha límite.

**Peso:** El peso  $w_j$  del trabajo  $j$  es un factor de prioridad, que refleja la importancia del

trabajo  $j$  en relación con otros trabajos en el sistema. Puede representar el costo de mantener el trabajo  $j$  en el sistema por una unidad de tiempo. El peso puede ser una retención o un costo de inventario, o puede ser la cantidad de valor que ya se ha agregado al trabajo.

**Tiempo de inicio ( $S_{ij}$ ):** El tiempo de inicio  $S_{ij}$  es el tiempo cuando la tarea  $j$  comienza su procesamiento en la máquina  $i$ . Si se omite el subíndice  $i$ , entonces  $S_j$  se refiere al momento en que el trabajo  $j$  comienza con su primer procesamiento en el sistema.

**Tiempo de finalización ( $C_{ij}$ ):** El tiempo de finalización  $C_{ij}$  es el momento en que se completa el trabajo  $j$  en la máquina  $i$ . Si se omite el subíndice  $i$ , entonces  $C_j$  se refiere al momento en que el trabajo  $j$  abandona el sistema.

Una característica importante en la programación de la producción es su configuración de máquina. Hay varias configuraciones importantes, la parte restante de esta sección cubre los más básicos.

- **Restricciones de precedencia:** En los problemas de secuenciación un trabajo puede empezar sólo después de haber terminado otros trabajos que han sido completados.
- **Restricciones de elegibilidad de la máquina:** En un ambiente de máquinas paralelas, es el caso en el que el trabajo  $j$  no puede ser asignado a solo alguna de las máquinas disponibles, esto es, puede estar sólo en alguna máquina de un subconjunto específico de máquinas. Como se describe anteriormente, esto puede ocurrir cuando las máquinas en paralelo no son exactamente idénticas.
- **Ajustes de tiempos y costos dependientes de la secuencia:** Cuando las máquinas tienen que ser reconfiguradas o debe hacerse una limpieza entre trabajos, entonces se dice que se hace un cambio o ajuste. Si la longitud del ajuste depende del trabajo que acaba de completarse y del trabajo que se iniciará, entonces el tiempo de ajuste es dependiente de la secuencia. Si el trabajo  $j$  es seguido por el trabajo  $k$  en máquina  $i$ , entonces el ajuste es denotado por  $s_{ijk}$ .
- **Interrupciones:** A veces, durante la ejecución de un trabajo, ocurre un evento que hace forzar al planeador a interrumpir el proceso de un trabajo para que la máquina quede disponible para realizar otro trabajo. Esto pasa, por ejemplo, cuando una orden con alta prioridad entra al sistema. Hay varias formas de interrupciones, en una de ellas el procesamiento adelantado no se pierde, y cuando el trabajo es regresado a la máquina se retoma desde donde se dejó. De acuerdo con otra forma de interrupción, el avance realizado del proceso se pierde.

### 2.3.3. Medidas de rendimiento y objetivos

En muchas instalaciones, muchos tipos de objetivos son importantes en ambientes de manufactura. A continuación se describen los objetivos básicos más importantes.

**Makespan:** Un *makespan* mínimo generalmente implica una buena utilización de la(s) máquina(s) y es importante sólo si el número de trabajos es finito. Se denota por  $C_{max}$  y se define como el tiempo cuando el último trabajo deja el sistema,

$$C_{max} = \max(C_1, \dots, C_n),$$

donde  $C_j$  es el tiempo de terminación del trabajo  $j$ .

**Tiempo total de terminación:** El tiempo total de terminación (TCT), se calcula como la suma de los tiempos de terminación de las tareas y es un indicador del inventario originado por una programación de tareas dada. Se expresa como

$$\sum C_j.$$

**Número de tareas tardías:** Minimizar el número de tareas tardías es un objetivo relacionado con la fecha de vencimiento  $d_j$ . Al principio puede parecer un objetivo algo artificial y sin interés práctico, sin embargo, en el mundo real es una medida de desempeño que se monitorea y según la cual se miden los gerentes. Se denotado como

$$\sum U_j,$$

donde  $U_j$  vale uno si  $C_j \leq d_j$  y cero en otro caso.

En una secuencia óptima para  $\sum U_j$  toma la forma de un conjunto de trabajos que cumplirán con sus fechas de vencimiento y que están programados primero, seguidos por el conjunto de trabajos restantes que no cumplirán con sus fechas de vencimiento. El primer conjunto de trabajos debe programarse de acuerdo con la regla de prioridad fecha de entrega más temprana (EDD).

**Tardanza total:** La minimización de la tardanza total de las tareas es un objetivo que también es importante en la práctica. Un trabajo tardío es un trabajo cuya tardanza es positiva. Minimizar el número de trabajos tardíos en la práctica no debe ser el único objetivo considerado. Algunos trabajos pueden tener que esperar un tiempo inaceptablemente prolongado si se reduce al mínimo el número de trabajos retrasados. Si, en cambio, la suma de las tardanzas se reduce al mínimo, es menos probable que la espera de un determinado trabajo sea inaceptablemente larga.

La tardanza del trabajo  $j$  se calcula como  $T_j = \max\{0, C_j - d_j\}$ , y la función objetivo de la tardanza total se expresa como:

$$\sum_{j=1}^n T_j$$

El programador a menudo busca también minimizar la tardanza máxima.

**Tardanza más prontitud:** En la práctica, puede ocurrir que si el trabajo  $j$  se completa antes de su fecha de vencimiento  $d_j$ , se incurre en una penalización por la prontitud. La prontitud del trabajo  $j$  se define como  $E_j = \max\{0, d_j - C_j\}$  y la función objetivo de la tardanza más prontitud se expresa como:

$$\sum_{j=1}^n E_j + \sum_{j=1}^n T_j.$$

El problema que se estudia en esta tesis siguiendo la notación anteriormente descrita se denota como  $R|s_{ijk}|\sum C_j$ . Esto significa que el ambiente de producción es de máquinas paralelas no relacionadas, se tienen tiempos de preparación de la máquina que dependen tanto de la tarea que se termina de procesar como de la tarea que entra a procesamiento, y la función objetivo a minimizar es el tiempo total de terminación.

### 2.3.4. Métodos heurísticos en secuenciación

Como se mencionó anteriormente existen heurísticas constructivas y heurísticas de mejora. En el contexto de problemas de secuenciación de tareas las heurísticas más conocidas son del tipo constructivo y son mejor conocidas como “Reglas de Prioridad”. El interés en una u otra reglas depende fuertemente de la función objetivo que se desee minimizar. La mayoría están pensadas para casos simples y son usualmente generalizadas para obtener soluciones a problemas más complejos.

Las reglas de prioridad más conocidas en problemas de secuenciación se describen a continuación. Vea Apéndice C en Pinedo (2005) para más heurísticas y reglas de prioridad en problemas de secuenciación.

1. (FCFS–*First Come First Served*): Primero en entrar, primero en salir. El primer trabajo en llegar al centro de trabajo se procesa primero.
2. (SPT–*Shortest Processing Time*): Tiempo de procesamiento más corto. Los trabajos más breves se procesan y terminan primero.
3. (EDD–*Earliest Due Date*): Fecha de entrega más próxima. El trabajo que tiene fecha de entrega más próxima se selecciona primero.
4. (LPT–*Longest Processing Time*): Tiempo de procesamiento más largo. Los trabajos con los tiempos de procesamientos más grandes a menudo son muy importantes y se seleccionan primero.
5. Regla de Johnson: Es un enfoque que minimiza el tiempo de procesamiento para establecer la secuencia de un grupo de trabajos en dos máquinas, al mismo tiempo que minimiza el tiempo muerto total en las máquinas.

# Capítulo 3

## Metodología de solución

En este capítulo describimos la notación usada para caracterizar el problema, luego presentamos el modelo matemático propuesto y finalmente presentamos las diferentes heurísticas.

### 3.1. Descripción del problema

Las siguientes suposiciones y anotaciones se usan para describir el problema:

- Hay un conjunto  $M$  de  $m$  máquinas paralelas.
- Las máquinas están continuamente disponibles, y cada máquina puede manejar un trabajo a la vez sin preferencia, es decir, una vez que el procesamiento de un trabajo ha comenzado, no se puede interrumpir.
- Hay un conjunto  $N$  de  $n$  trabajos para programar.
- Todos los trabajos están disponibles en el tiempo cero. No se imponen restricciones de precedencia entre los trabajos.
- Cada trabajo  $j$  tiene asociado un tiempo de procesamiento  $p_{ij}$  en cada máquina  $i$ .
- Hay un tiempo de preparación dependiente de la máquina  $s_{ijk}$  para procesar el trabajo  $k$  justo después del trabajo  $j$ , en la máquina  $i$ .
- El objetivo es minimizar  $\sum_j C_j$ , el tiempo total de terminación o TCT.

A continuación se mostrará un ejemplo del problema con tiempos de preparación y tiempos de procesamiento.

Las figuras 3.1 y 3.2 muestran los tiempos de procesamiento y tiempos de preparación para una instancia de nombre `I_6_2_S_1-9_1.txt` que indica que es un instancia con 6 tareas, 2 máquinas, tiempos de preparación generados en el rango  $[1, 9]$ .

En la figura 3.3 se muestra el diagrama de Gantt correspondiente a la solución con los trabajos 6, 3 y 1 en la máquina 0 y los trabajos 2, 4 y 5 en la máquina 1, donde  $C_1$  corresponde al tiempo de terminación de la tarea 1,  $C_2$  al tiempo de terminación de la tarea 2 y así sucesivamente.

Maq 0	$P_{i0}$	Maq 1	$P_{i1}$
0	1	1	4
0	87	1	21
0	28	1	68
0	32	1	17
0	38	1	43
0	9	1	48

Figura 3.1: Tiempos de procesamiento de la instancia de ejemplo

	0	1	8	1	3	9
	4	0	7	3	7	8
$S_{0ij}$	7	3	0	2	3	5
	3	8	3	0	5	2
	8	3	7	9	0	5
	8	8	1	2	2	0

	0	5	1	6	1	7
	6	0	7	7	6	2
$S_{1ij}$	7	6	0	9	6	9
	3	7	3	0	1	7
	5	8	5	6	0	9
	7	4	1	7	9	0

Figura 3.2: Tiempos de preparación de la instancia del ejemplo

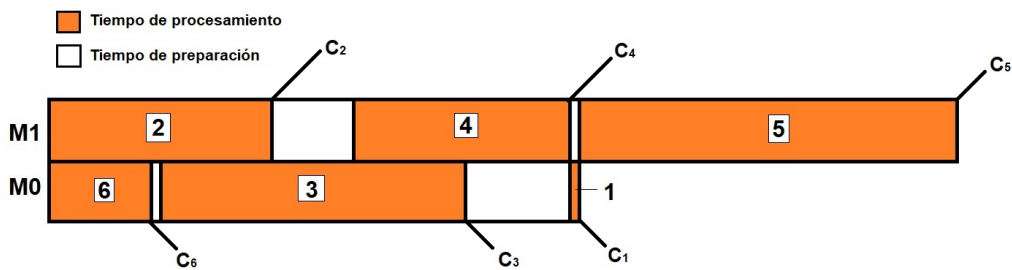


Figura 3.3: Representación gráfica de la solución del ejemplo con 2 máquinas y 6 tareas



### 3.1.1. Cálculo del tiempo total de terminación

Considerando que una secuencia de trabajos en una máquina se puede definir como un subconjunto de trabajos ordenados de  $N$  tareas (con el trabajo 0 en la primera posición) y se representa por  $S = \{0, [1], \dots, [r]\}$ , aquí  $r$  es la cantidad de trabajos en la secuencia y  $[j]$  significa el trabajo que ocupa la posición  $j$  en la secuencia  $S$ . Cuando no hay tiempo de ocio, utilizando los valores  $z_{ijk} = s_{ijk} + p_{ik}$ , el tiempo de terminación del trabajo que ocupa la posición  $q$  en la secuencia  $C_{i[q]}$  se puede calcular como la suma de los valores  $z_{ijk}$  para todos los trabajos que preceden a  $[q]$  en la máquina  $i$ , significa que

$$C_{i[q]} = \sum_{j=1}^q z_{i[j-1][j]}, \quad (3.1)$$

luego agregando los tiempos de terminación de los trabajos en la misma máquina obtenemos

$$\begin{aligned} \sum_{q=1}^r C_{i[q]} &= rz_{i0[1]} + (r-1)z_{i[1][2]} + \dots + 2z_{i[r-2][r-1]} + z_{i[r-1][r]} \\ &= \sum_{q=0}^{r-1} (r-q)z_{i[q][q+1]} \\ &= \sum_{q=0}^{r-1} (r-q)s_{i[q][q+1]} + \sum_{l=1}^r (r+1-q)p_{i[q]} \end{aligned} \quad (3.2)$$

eso significa que si el trabajo  $j$  procede inmediatamente al trabajo  $k$  en una máquina  $i$ , entonces contribuye a la función objetivo con  $(r-q)z_{i[q][q+1]}$ , donde  $(r-q)$  es la cantidad de trabajos que siguen al trabajo  $[q]$  en la secuencia.

A continuación ejemplificamos cómo se calcula la evaluación de la función objetivo para la solución de la figura 3.3 usando los datos que ahí aparecen. Para ello, haremos notar que hay dos formas de evaluarla: 1) Calcular todos los tiempos de terminación de las tareas y luego hacer la suma, y 2) usar la expresión (3.2) para calcular la suma de los tiempos de terminación de las tareas en una misma máquina.

Primero, vamos a calcular para la máquina 0 los tiempos de preparación más tiempo de procesamiento para cada par de tareas consecutivas, donde  $z_{i,j,k} = s_{i,j,k} + p_{i,k}$ :

$$\begin{aligned} z_{0,0,6} &= 0 + 9 = 9 \\ z_{0,6,3} &= 1 + 28 = 29 \\ z_{0,3,1} &= 7 + 1 = 8 \end{aligned}$$

Así en cada uno de los 3 casos presentados en la máquina 0, calculamos los tiempos de terminación de las tareas asignadas a la máquina 0 ( $C_6, C_3$  y  $C_1$ ), sumando todos los tiempos de preparación y de procesamiento correspondientes.

$$\begin{aligned} C_6 &= z_{0,0,6} = 9 \\ C_3 &= z_{0,0,6} + z_{0,6,3} = 9 + (1+28) = 38 \\ C_1 &= z_{0,0,6} + z_{0,6,3} + z_{0,3,1} = 9 + (1+28) + (7+1) = 46 \end{aligned}$$

Hacemos cálculos de manera análoga con la máquina 1, calculamos  $z_{i,j,k}$  y sus respectivos  $C_i$ :

$$\begin{aligned} z_{1,0,2} &= 0 + 21 = 21 \\ z_{1,2,4} &= 7 + 17 = 24 \\ z_{1,4,5} &= 1 + 43 = 44 \end{aligned}$$

$$\begin{aligned} C_2 &= z_{1,0,2} = 21 \\ C_4 &= z_{1,0,2} + z_{1,2,4} = 21 + (7+17) = 45 \\ C_5 &= z_{1,0,2} + z_{1,2,4} + z_{1,4,5} = 21 + (7+17) + (1+43) = 89 \end{aligned}$$

Para terminar calculamos el TCT de las dos formas antes mencionadas:

Forma 1: Hacer una suma de todos nuestros tiempos ( $C_j$ ) obtenidos en cada tarea.

$$\begin{aligned} TCT &= (46) + (21) + (38) + (45) + (89) + (9) \\ TCT &= 248 \end{aligned}$$

Forma 2: Sumando cada tiempo necesario para cambiar de una tarea a otra, multiplicando el numero de veces que aparece un tiempo de preparación o de procesamiento.

$$\begin{aligned} TCT &= 3z_{0,0,6} + 2z_{0,6,3} + z_{0,3,1} + 3z_{1,0,2} + 2z_{1,2,4} + z_{1,4,5} \\ TCT &= 3(9) + 2(29) + 8 + 3(21) + 2(24) + 44 \\ TCT &= 248 \end{aligned}$$

Para ambos métodos, podemos observar que el resultado al final es el mismo. Sin embargo, la segunda forma es mas conveniente para la implementación de la evaluación del TCT en un algoritmo, ya que considera datos concretos del problema.

## 3.2. Formulación propuesta

Para una mejor comprensión del modelo matemático dependiente del tiempo presentado a continuación, nos apoyaremos en la red multinivel que se muestra en la figura 3.4. La red es usada para modelar la asignación de tareas a una máquina dada y el orden en que las tareas son procesadas en dicha máquina. Cada nodo representa a una tarea y cada nivel representa una posición en la que las tareas son procesadas.

Entonces, la contribución de cada arco  $(j, k)$  depende del nivel en este arco conectado en la red y qué máquina lo usó. Si un arco  $(j, k)$  conecta el nivel  $l$  con el nivel  $l + 1$  en la máquina  $i$ , entonces la contribución es  $(n - l) \times (s_{ijk} + p_{ik})$ . Note que si un trabajo se asigna en el nivel  $l$  en la máquina  $i$  los siguientes niveles en la misma máquina también deben tener asignados los trabajos.

Sea  $Y_{ij}^l = 1$  si el trabajo  $j$  se procesa en el nivel  $l$  en la máquina  $i$ , y 0 en caso contrario, y  $X_{ijk}^l = 1$  si el trabajo  $j$  se procesa en el nivel  $l$  en la máquina  $i$  y el trabajo  $k$  en el nivel  $l + 1$ , 0 en caso contrario.

Nuestra formulación (denotada por M1) para el problema  $R|s_{ijk}|\sum C_j$  se basa en la reformulación realizada por Angel-Bello et al. (2013) y se establece de la siguiente manera:

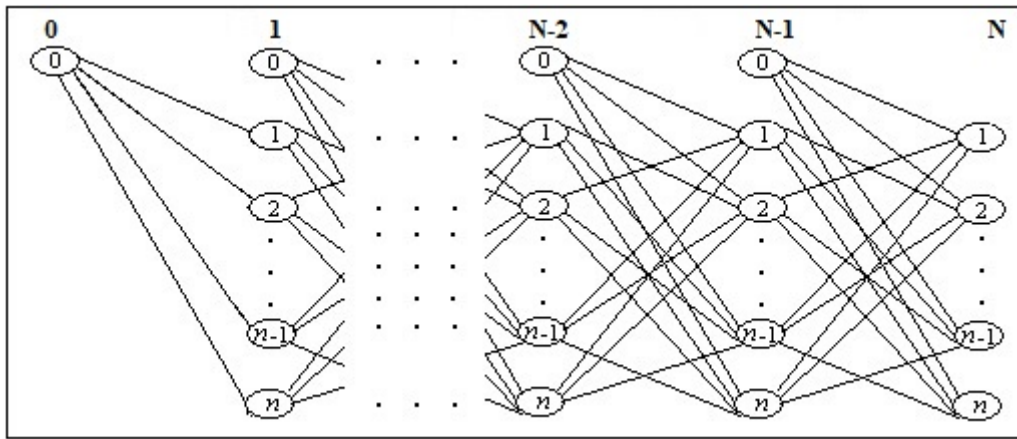


Figura 3.4: Red multinivel usada como auxiliar en la modelación matemática del problema

Minimizar:

$$\sum_{i=1}^m \left( \sum_{j=0}^n \sum_{\substack{k=1 \\ k \neq j}}^n \sum_{l=0}^{n-1} (n-l) s_{ijk} X_{ijk}^l + \sum_{k=1}^n \sum_{l=1}^n (n+1-l) p_{ik} Y_{ik}^l \right) \quad (3.3)$$

Sujeto a:

$$\sum_{i=1}^m \sum_{l=1}^n Y_{ij}^l = 1, \quad j \in N, \quad (3.4)$$

$$\sum_{j=1}^n Y_{ij}^l \leq 1, \quad i \in M, l \in N, \quad (3.5)$$

$$\sum_{\substack{k=1 \\ k \neq j}}^n X_{ijk}^l = Y_{ij}^l \quad j \in N, l \in N^-, i \in M, \quad (3.6)$$

$$\sum_{\substack{j=0, \\ j \neq k}}^n X_{ijk}^l = Y_{ik}^{l+1} \quad k \in N, l \in N_0^-, i \in M, \quad (3.7)$$

$$\begin{aligned} Y_{ij}^l &\in \{0, 1\}, & j &\in N_0, l \in N, i \in M, \\ X_{ijk}^l &\geq 0, & j, k &\in N, l \in N, i \in M. \end{aligned} \quad (3.8)$$

La función objetivo (3.3) minimiza el tiempo total de terminación, tenga en cuenta que es sólo la expresión (3.2) en términos de las variables  $Y$ 's y  $X$ 's. Las restricciones (3.4) asignan cada trabajo a un solo nivel en una sola máquina. Las restricciones (3.5) determina que en cada nivel de cada máquina se puede asignar como máximo un trabajo. Las restricciones (3.6) determinan que si un trabajo  $j \in N$  se asigna en un nivel (diferente del último) en una máquina, entonces tiene una tarea sucesora  $k \in N$  en la misma máquina en el siguiente nivel. De forma similar, las restricciones (3.7) establecen que cada trabajo sólo tiene un predecesor en la misma máquina justo en el nivel anterior. El predecesor puede ser el trabajo ficticio 0. Estas restricciones junto con las restricciones (3.6) también garantizan que los trabajos asignados a una máquina estén

ubicados juntos en los últimos niveles. La naturaleza de las variables  $Y$ 's y las restricciones (3.6) y (3.7) garantizan que las variables  $X$ 's tomen los valores 0, 1.

Se realizó experimentación computacional con el modelo anterior, utilizando el optimizador comercial CPLEX, y sólo fue posible resolver a optimalidad instancias de hasta 40 tareas y 5 máquinas. Algunas instancias de 50 tareas no fue posible resolverlas a optimalidad. Lo anterior, motiva el desarrollo de algoritmos heurísticos para resolver instancias de mayor tamaño.

Los resultados de esta experimentación se describen a detalle en el capítulo 4.

### 3.3. Algoritmos heurísticos

Los algoritmos heurísticos propuestos son métodos constructivos, esto es, construyen una solución factible que satisfaga todos los requerimientos y/o restricciones del problema. En general en un algoritmo constructivo la solución se construye paso a paso, en cada uno de ellos del se tiene una solución parcial  $S_p$  y una lista de los elementos  $L_a$  que aún no han sido agregados a la solución. Luego, un subconjunto de elementos de  $L_a$  es seleccionados y/o evaluados, se elige solo uno de ellos para incorporarse a la solución parcial  $S_a$  de la forma más favorable. Luego  $L_a$  se actualiza y el proceso se repite hasta que todos los elementos han sido agregados a la solución. Lo que varía de un constructivo a otro es la forma en que los elementos son seleccionados y/o evaluados.

En este trabajo de tesis se proponen cuatro procedimientos constructivos, variando ambas características: 1) cómo se selecciona el siguiente elemento para ser agregado a la solución y 2) cómo evaluar la contribución de los elementos si se agregarán a la solución. Se proponen dos formas de selección y dos formas de evaluación por lo que combinando una forma de selección con una de evaluación obtenemos en total cuatro constructivos.

#### 3.3.1. Formas de selección

1. La primer forma de selección es una adaptación entre la regla de prioridad de tiempo de procesamiento más corto (SPT) o la regla de prioridad del tiempo de procesamiento más largo (LPT), en dependencia del tipo de evaluación seleccionado. Se crea una lista de candidatos restringidos basada en el tiempo de procesamiento promedio más corto. De ahí se selecciona un elemento al azar y este será agregado a la solución como mejor convenga.
2. De los elementos disponibles, seleccionamos al azar  $a$  elementos y determinamos la contribución de los mismos si fueran agregados, el elemento con la mejor contribución es elegido.

Las dos formas de selección del elemento que se agregará a la solución se debe a que, por un lado, las reglas de prioridad basadas en SPT o LPT son muy frecuentes en secuenciación, mientras que la segunda forma es usada por los algoritmos tipo GRASP que también se usan con mucha frecuencia en problemas combinatorios.

#### 3.3.2. Formas de evaluación

1. Evaluación del incremento en el *makespan* para la inserción de una tarea. La contribución de los elementos considera el menor incremento del *makespan*, para ello todas las posibles

posiciones son consideradas. Al finalizar la construcción se determina cuál es el valor de la función objetivo que se obtuvo.

2. Evaluación del incremento en el TCT para la inserción de una tarea. La contribución de los elementos es evaluado como el menor incremento del tiempo total de terminación. Todas las posibles posiciones son consideradas y dada la complejidad de la función objetivo se utiliza la ecuación (3.2) para reducir el tiempo que se tarda en cada evaluación.

Como se mencionó en la sección anterior, una secuencia  $S_i = \{0, [1], [2], \dots, [n_i]\}$  es un subconjunto ordenado de  $N$  tareas en la máquina  $i$ , con la tarea 0 en la primera posición,  $n_i$  es el cantidad de tareas en la secuencia en la máquina  $i$  y  $[q]$  significa la tarea que ocupa la posición  $q$  en secuencia  $S_i$ . Luego, el TCT de la secuencia  $S$  se puede calcular como:

$$TCT = \sum_{q=0}^{n_i-q} (n_i - i) z_{[q][q+1]},$$

donde los elementos  $z_{ijk}$  corresponde a la matriz de costos calculados como  $z_{ijk} = s_{ijk} + p_{ik}$ .

Basados en la ecuación anterior, Nucamendi-Guillén et al. (2016) representar el incremento del mejor punto de inserción de una tarea  $j$  en una máquina para el caso de máquinas idénticas, la siguiente ecuación se establece para una máquina  $i$ , la cual determina el incremento en la función objetivo si una tarea  $j$  se insertara en cada posible posición  $q$ .

$$\Delta = \begin{cases} (n_i + 1)z_{i0j} + n_i(z_{ij[1]} - z_{i0[1]}) & \text{si } q = 1 \\ \sum_{l=1}^{q-1} z_{i[l-1][l]} + (n_i - q + 2)z_{i[q-1]j} + (n_i - q + 1)(z_{ij[q]} - z_{i[q-1][q]}) & \text{si } 2 \leq q \leq n_i \\ \sum_{l=1}^{n_i} z_{i[l-1][l]} + z_{i[n_i]j} & \text{si } q = n_i + 1 \end{cases}$$

La expresión anterior para  $2 \leq q$  involucra un término que se incrementa por el valor de un arco desde la posición  $q - 1$  hasta la posición  $q$ . Es decir, a través del término  $\sum_{l=1}^{q-1} z_{i[l-1][l]}$  estas expresiones permiten el uso de cálculos realizados en la posición  $q - 1$  para calcular el valor de inserción en la posición  $q$ .

Las dos formas de evaluación de los elementos se proponen debido a que el TCT es una función objetivo difícil de evaluar, mientras que el *makespan* es una función objetivo más fácil de evaluar, por lo que puede ser interesante construir una solución de acuerdo al *makespan* y observar qué tan bueno es el valor del TCT de dicha solución.

### 3.3.3. Constructivos

La forma mas común de usar la representación de solución para los problemas de secuenciación en máquinas paralelas es con un arreglo de trabajos  $S_i$  para cada máquina  $i$  que representa el orden de procesamiento de trabajos asignados para la máquina, esto es,  $S = \{S_1, S_2, \dots, S_m\}$ , donde,  $S_1$ , representa el orden de procesamiento de trabajos asignados para la máquina 1 y  $S_2$  lo mismo para la máquina  $S_2$  y así sucesivamente.

Realizando una combinación de características de selección de elementos con características de evaluación tenemos cuatro constructivos que comparten las siguientes características:

1. Se inicializa S.

2. Se crea una lista restringida de candidatos (RCL).
3. Se evalúa un elemento  $j$  de la RCL.
4. Se determina el mejor punto de inserción al evaluar la inserción de  $j$  en cada posición de la solución parcial.
5. Se elige el elemento y posición mejor evaluados.
6. Se actualiza la solución  $S$ , y demás información.

**Constructivo 1 (C1)** Los trabajos se ordenan de acuerdo con el tiempo de procesamiento promedio más largo  $\hat{p}_j = \frac{1}{n} \sum_{i=1}^m p_{ij}$ , luego se crea una RCL construida con los  $a$  primeros elementos y se selecciona un trabajo aleatoriamente de ella. El trabajo seleccionado se inserta en el mejor punto de inserción en la solución parcial, de modo que aumente menos el valor de la función objetivo del *makespan*. Luego, la RCL se actualiza y este proceso continúa hasta que se hayan asignado todos los trabajos.

Al final, se evalúa cuál es el tiempo total de terminación de la solución construida.

**Ejemplo constructivo 1:**

Para este constructivo utilizaremos el concepto de *span* de una máquina  $i$  denotado como  $O_i$ , que representa el tiempo al cual la máquina termina de procesar su última tarea asignada. Denotemos también por  $\Delta_i$  el incremento del span  $O_i$  de la máquina  $i$  para el mejor punto de inserción para esa máquina y a  $O'_i = O_i + \Delta_i$  como el span de la máquina si se realizara la inserción del trabajo en dicha máquina, entonces la mejor máquina para insertar el trabajo  $j$  será  $i^* = \operatorname{argmin}\{O'_i\}$ , es decir, la máquina en donde se obtenga el menor span.

A continuación ejemplificaremos paso a paso cómo se lleva a cabo la construcción de una solución utilizando el constructivo 1, para ello, utilizaremos los datos de la instancia que mostramos en las figuras 3.1 y 3.2. Primero inicializamos la solución y el span de las máquinas

$$S = \{\{0, 0\}, \{0, 0\}\}$$

$$O = \{0, 0\}$$

Enseguida calculamos el tiempo de procesamiento promedio de cada tarea y ordenamos las tareas según el tiempo de procesamiento mas largo:

Tareas	1	2	3	4	5	6
$\hat{p}_j$	2.5	54	48	24.5	40.5	28.5
Tareas ordenadas	2	3	5	6	4	1
$\hat{p}_j$	54	48	40.5	28.5	24.5	2.5

Después, creamos la RCL con los candidatos con  $a$  primeros elementos a partir del ordenamiento anterior, en este ejemplo tomaremos de valor  $a = 4$ , y una vez formada la RCL, se seleccionará de manera aleatoria uno de los elementos de la lista:

$$\text{RCL} = \{2, 3, 5, 6\}$$

Una vez teniendo el elemento a evaluar (3), se inserta en la mejor posición de la solución parcial, de modo que se inserte en la máquina con el menor span después de la inserción. El valor de inserción del trabajo  $j$  en la mejor posición en cada máquina se da de la siguiente manera. Como el span es cero en ambas máquinas, al igual que los tiempos de preparación para la primer tarea en una máquina, la evaluación se obtiene directamente de los tiempos de procesamiento, quedándonos de la siguiente manera:

$$\begin{aligned} O'_0 &: 28 \\ O'_1 &: 68 \end{aligned}$$

El span mínimo después de la inserción se obtiene en la máquina 0 en la posición 1. Actualizamos la solución

$$\begin{aligned} S &= \{\{0, 3, 0\}, \{0, 0\}\} \\ O &= \{28, 0\} \end{aligned}$$

A continuación volvemos al paso donde construimos la RCL, como aquí seleccionamos un elemento al azar y este fue colocado ya en una máquina, ahora nos quedan 3 elementos en la RCL, por tanto, seleccionaremos la siguiente tarea con tiempo de procesamiento más largo para incluirla a la RCL y posteriormente procedemos a seleccionar de manera aleatoria un elemento de la nueva RCL:

$$\text{RCL} = \{2, 5, 6, 4\}$$

Una vez teniendo el elemento a evaluar (4), se inserta en la mejor posición de la solución parcial, de modo que se inserte en la máquina con el menor span después de la inserción:

$$\begin{aligned} O'_0 &= \text{mín}\{63, 62\} \\ O'_1 &= 17 \end{aligned}$$

El span mínimo después de la inserción se obtiene en la máquina 1 en la posición 1. Actualizamos la solución

$$\begin{aligned} S &= \{\{0, 3, 0\}, \{0, 4, 0\}\} \\ O &= \{28, 17\} \end{aligned}$$

Una vez actualizada la solución, repetimos el procedimiento anterior hasta que asignemos todas las tareas en las máquinas. Insertando respectivamente las tareas 6, 5, 1 y 2, en ese orden. La solución final que obtendremos, nos da el ordenamiento de las tareas en cada una de las máquinas:

$$\begin{aligned} S &= \{0, 6, 3, 1, 0\}, \{0, 2, 4, 5, 0\} \\ O &= \{46, 89\} \end{aligned}$$

Finalmente calculamos nuestra función objetivo para la solución construida, dándonos  $TCT = 248$ .

**Constructivo 2 (C2)** Se evalúa en el mejor punto de inserción en la solución parcial, de modo que aumente menos el valor de la función objetivo del *makespan*. De los trabajos evaluados nos quedamos con el que ofrezca el menor incremento.

**Constructivo 3 (C3)** Los trabajos se ordenan de acuerdo con el tiempo de procesamiento promedio más corto  $\hat{p}_j = \frac{1}{n} \sum_{i=1}^m p_{ij}$ , luego se conforma la RCL con los  $a$  primeros elementos y se selecciona un elemento de manera aleatoria. El trabajo seleccionado se inserta en el mejor punto de inserción en la solución parcial, de modo que aumente menos el valor de la función objetivo del TCT. Luego, la RCL se actualiza y este proceso continúa hasta que se hayan asignado todos los trabajos.

**Constructivo 4 (C4)** En el procedimiento constructivo, se seleccionan  $a$  elementos para conformar la RCL. Cada trabajo seleccionado se evalúa en el mejor punto de inserción en la solución parcial, de modo que aumente menos el valor de la función objetivo del TCT. De los trabajos evaluados nos quedamos con el que ofrezca el menor incremento. Los incrementos son evaluados de la misma forma que en el constructivo 3.

**Ejemplo del constructivo 4:**

Para este constructivo consideraremos la misma representación de la solución pero ya no será necesario definir el span de las máquinas. Denotaremos además el TCT como el tiempo total de terminación, y a  $\Delta_i$  como el incremento del tiempo del TCT si se realizara la inserción en la máquina  $i$  en su mejor punto de inserción. Consideremos las 6 tareas a asignar:

Tareas: 1 2 3 4 5 6

Construimos la RCL seleccionando  $a$  elementos al azar que serán los que tomemos a evaluar, donde  $a = 4$ :

$$RCL = \{2, 3, 1, 6\}$$

Una vez obtenidas nuestras tareas a evaluar, pasamos a evaluar la inserción de cada una de las tareas en cada máquina, seleccionaremos la tarea donde obtengamos el menor incremento:

Tarea 2	Tarea 3	Tarea 1	Tarea 6
$\Delta_0 : 87$	$\Delta_0 : 28$	$\Delta_0 : 1$	$\Delta_0 : 9$
$\Delta_1 : 21$	$\Delta_1 : 68$	$\Delta_1 : 4$	$\Delta_1 : 48$

El span mínimo después de la inserción se obtiene de la tarea 1, en la máquina 0 y en la posición 1. Actualizamos la solución:

$$S = \{\{0, 1\}, \{0\}\}$$

$$TCT = 1$$



Una vez seleccionada la tarea y colocada en una máquina, tendremos que seleccionar otros 4 elemento al azar para actualizar la RCL:

$$RCL = \{4, 6, 5, 2\}$$

Nuevamente, pasamos a insertar cada una de las tareas en cada máquina, donde seleccionaremos la tarea donde obtengamos el menor incremento en la FO:

Tarea 4	Tarea 6	Tarea 5	Tarea 2
$\Delta_0 : \min\{67, 34\}$	$\Delta_0 : \min\{26, 19\}$	$\Delta_0 : \min\{84, 42\}$	$\Delta_0 : \min\{178, 89\}$
$\Delta_1 : 17$	$\Delta_1 : 48$	$\Delta_1 : 43$	$\Delta_1 : 21$

El span mínimo después de la inserción se obtiene de la tarea 4, en la máquina 1 y en la posición 1. Actualizamos la solución:

$$S = \{\{0, 1\}, \{0, 4\}\}$$

$$TCT = 18$$

Una vez actualizada la solución, repetimos el procedimiento anterior hasta que asignemos todas las tareas en las máquinas.

Insertando respectivamente las tareas 6, 2, 3 y 5 , en ese orden. La solución final que obtendremos, nos da el ordenamiento de las tareas en cada una de las máquinas:

$$S = \{0, 1, 6, 3, 5\}, \{0, 4, 2\}$$

$$TCT = \{219\}$$

# Capítulo 4

## Experimentación

El modelo matemático propuesto se implementó en Visual C++ y se utilizó CPLEX 12.6. La implementación de los 4 algoritmos constructivos se llevó a cabo en lenguaje C/C++ para la experimentación computacional con el modelo matemático y los algoritmos constructivos se utilizó una computadora HP con un procesador Intel Core i5 de 4Gb de RAM y un disco duro de 840 Gb.

### 4.1. Instancias

La experimentación computacional se llevó a cabo sobre tres conjuntos de instancias pequeñas, medianas y grandes cuyas características describimos a continuación.

Para las instancias pequeñas, se consideraron las siguientes combinaciones de trabajos y máquinas, donde para las tareas a realizar se tenía que  $n \in \{6, 8, 10, 12\}$  y en cuanto al número de máquinas se tiene que  $m \in \{2, 3, 4, 5\}$ . El tiempo de procesamiento se distribuye uniformemente en el rango  $[1, 99]$  y los tiempos de preparación en rangos  $[1,9]$ ,  $[1,49]$ ,  $[1,99]$ ,  $[1,124]$ , para cada rango de tiempo de preparación se tienen 10 instancias por lo que cada combinación de  $n$  y  $m$  se tienen 40 diferentes instancias, dándonos un total de 160 instancias por cada valor de  $n$ , lo que finalmente nos daría un total de 640 instancias pequeñas.

Mientras que para las instancias medianas, vamos a considerar que el número de tareas son  $n \in \{20, 30, 40\}$  y el número de máquinas sería de  $m \in \{2, 3, 4, 5\}$ . El tiempo de procesamiento se distribuye uniformemente en el rango  $[1, 99]$  y los tiempos de procesamientos en rangos  $[1,49]$ ,  $[1,99]$ ,  $[1,124]$ , para cada rango de tiempo de preparación se tienen 10 instancias por lo que cada combinación de  $n$  y  $m$  se tienen 30, obteniendo 120 instancias, lo que nos da un total de 360 instancias medianas.

Para las instancias grandes, vamos a considerar que el número de tareas son  $n \in \{50, 100, 150, 200, 250\}$  y el número de máquinas sería de  $m \in \{10, 15, 20, 25, 30\}$ . El tiempo de procesamiento se distribuye uniformemente en el rango  $[1, 99]$  y los tiempos de preparación en rangos  $[1,9]$ ,  $[1,49]$ ,  $[1,99]$ ,  $[1,124]$ , para cada rango de tiempo de preparación se tienen 10 instancias por lo que cada combinación de  $n$  y  $m$  se tienen 40, obteniendo 200 instancias, lo que nos da un total de 1000 instancias grandes.

## 4.2. Desempeño del modelo

A continuación mostramos los resultados obtenidos al resolver el modelo matemático propuesto (M1). Esta experimentación se realiza sobre las instancias medianas, es decir, el número de máquinas varía entre 20, 30, 40, mientras que el número de tareas varía de 2 a 5. Cabe destacar que también se realizó experimentación con las instancias pequeñas pero dado que el modelo fue capaz de resolver todas las instancias pequeñas sin ningún problema, no se incluyó la correspondiente tabla de resultados.

La tabla 4.1 muestra los resultados para dicho conjunto de instancias. Las columnas 1 y 2, hacen referencia al tamaño de la instancia. Las columnas 3 y 4 reportan la desviación respecto al óptimo (desv) y el tiempo (en segundos) para la relajación lineal (RL). Finalmente las columnas 5, 6 y 7 muestran el número de instancias no resueltas ( $\#nr$ ), el gap, y el tiempo de ejecución (en segundos).

Tabla 4.1: Desempeño del modelo y de su relajación lineal para resolver las instancias medianas minimizando el TCT

Tamaño		M1				
$n$	$m$	RL		Opt		
		desv	tiempo	$\#nr$	gap	tiempo
20	2	1.31	0.19	0	0	1.60
	3	1.73	0.25	0	0	1.94
	4	1.48	0.28	0	0	1.70
	5	0.89	0.23	0	0	1.45
30	2	1.05	1.02	0	0	47.58
	3	1.82	1.46	0	0	37.43
	4	1.63	1.80	0	0	14.74
	5	1.54	1.62	0	0	11.63
40	2	1.30	2.72	1	1.2	755.88
	3	1.49	4.14	0	0	371.06
	4	1.60	5.75	0	0	121.86
	5	1.82	6.87	0	0	115.49

En la tabla 4.1 podemos observar los diferentes datos sobre las instancias medianas, donde tomamos por tamaños de máquinas con 20, 30, 40, y en cada una con 2, 3, 4 ó 5 tareas respectivamente.

Como podemos observar, en los tamaños de 20 máquinas, su relajación lineal es muy buena, ya que tiene desviaciones cercanas al 0, oscilando entre .9 la mas baja y 1.7 la mas alta, además la relajación lineal se resuelve en tiempos relativamente pequeños.

Se observa que el tiempo de cómputo comienza a incrementar considerablemente conforme crece el tamaño de la instancia y pudimos observar que por ejemplo para las instancias de 40 tareas ya hay una instancia sin resolver a optimalidad. Esto nos lleva a la necesidad de desarrollar algoritmos heurísticos para aquellos tamaños de instancia que el modelo no es capaz de resolver.

### 4.3. Desempeño de las heurísticas

A continuación mostramos los resultados obtenidos para comparar el desempeño de las diferentes heurísticas. Esta experimentación se realiza sobre las instancias chicas y medianas en donde se incluye la comparación con los soluciones óptimas obtenidas con el modelo. Todas las heurísticas dependen de un parámetro  $a$  que corresponde a la cantidad de elementos en la RCL. En los experimentos que realizaremos a continuación se consideraron dos valores  $a = 2$  y  $a = 4$ .

#### 4.3.1. En instancias pequeñas y medianas

En las tabla 4.2 y 4.3 se presentan las comparativas de los constructivos respecto a las soluciones óptimas obtenidas con el modelo en instancias chicas para los valores de  $a = 2$  y  $a = 4$ . En las tabla 4.4 y 4.5 se presentan las comparativas de los constructivos respecto a las soluciones óptimas obtenidas con el modelo en instancias medianas para los valores de  $a = 2$  y  $a = 4$ .

En las cuatro tablas, las columnas 1 y 2, hacen referencia al tamaño de la instancia, es decir, número de tareas ( $n$ ) y número de máquinas ( $m$ ), respectivamente. En las columnas 3, 5, 7, y 9 contienen la desviación promedio de las soluciones obtenidas con las heurísticas C1, C2, C3 y C4, respectivamente, respecto a las soluciones óptimas. Mientras que en las columnas 4, 6, 8, y 10 contienen el tiempo promedio de cada grupo de instancias por cada valor de  $n$  y  $m$ , para las cuatro heurísticas.

Al final de la tabla se encuentra una línea con el nombre de “Promedio” donde podemos encontrar el promedio global de las desviaciones y tiempos de cada una de las heurísticas respectivamente.

Tabla 4.2: Resultados de las heurísticas en  $n$  instancias pequeñas con parámetro de  $a = 2$

$n$	$m$	C1		C2		C3		C4	
		Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo
6	2	23.08	0.00E+00	7.15	6.75E-04	18.85	0.00E+00	6.69	5.00E-05
	3	19.62	0.00E+00	10.57	8.50E-04	16.86	0.00E+00	6.52	0.00E+00
	4	17.44	0.00E+00	7.63	9.00E-04	16.39	0.00E+00	9.15	2.50E-05
	5	12.73	1.50E-04	13.24	7.75E-04	16.71	1.00E-04	10.97	0.00E+00
8	2	23.66	0.00E+00	7.78	7.75E-04	19.22	0.00E+00	5.93	1.25E-04
	3	22.36	0.00E+00	11.27	1.03E-03	20.68	5.00E-05	8.84	2.50E-05
	4	24.17	0.00E+00	13.45	5.25E-04	18.56	5.00E-05	8.43	5.00E-05
	5	20.67	0.00E+00	14.01	1.00E-04	16.76	5.00E-05	8.23	1.25E-04
10	2	28.37	0.00E+00	11.79	2.25E-04	26.31	2.50E-05	10.23	2.50E-05
	3	28.28	0.00E+00	12.73	1.25E-04	20.69	5.00E-05	7.03	5.00E-05
	4	26.35	0.00E+00	12.80	1.75E-04	18.36	5.00E-05	7.40	7.50E-05
	5	21.21	2.50E-05	13.27	7.50E-05	19.07	0.00E+00	9.53	0.00E+00
12	2	26.00	0.00E+00	9.68	3.00E-04	25.62	7.50E-05	7.87	1.75E-04
	3	25.90	0.00E+00	10.66	1.50E-04	22.91	0.00E+00	9.28	1.50E-04
	4	32.30	0.00E+00	13.95	7.50E-05	20.03	1.00E-04	8.12	0.00E+00
	5	23.52	0.00E+00	14.88	1.25E-04	22.20	0.00E+00	10.90	5.00E-05
<b>Promedio</b>		<b>23.48</b>	<b>1.09E-05</b>	<b>11.55</b>	<b>4.30E-04</b>	<b>19.95</b>	<b>3.44E-05</b>	<b>8.44</b>	<b>5.78E-05</b>

Tabla 4.3: Resultados de las heurísticas en  $n$  instancias pequeñas con parámetro de  $a = 4$ 

$n$	$m$	C1		C2		C3		C4	
		Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo
6	2	24.80	0.00E+00	9.41	8.75E-04	17.07	2.50E-05	5.04	0.00E+00
	3	19.70	2.50E-05	8.84	7.25E-04	16.23	5.00E-05	5.52	5.00E-05
	4	15.80	0.00E+00	6.86	9.50E-04	14.65	2.50E-05	7.60	2.50E-05
	5	19.25	0.00E+00	11.26	5.25E-04	14.28	5.00E-05	8.64	0.00E+00
8	2	23.67	0.00E+00	9.23	1.00E-04	18.14	0.00E+00	4.86	0.00E+00
	3	25.02	0.00E+00	10.85	2.75E-04	19.85	1.00E-04	6.21	0.00E+00
	4	22.41	0.00E+00	10.63	2.00E-04	18.61	1.50E-04	9.05	5.00E-05
	5	20.24	0.00E+00	13.15	2.50E-04	16.48	1.00E-04	9.82	1.25E-04
10	2	29.98	2.50E-05	12.64	7.50E-04	22.46	5.00E-05	7.75	5.00E-05
	3	29.51	2.50E-05	10.73	7.75E-04	21.25	2.00E-04	7.97	5.00E-05
	4	23.38	0.00E+00	11.36	9.50E-04	16.36	1.25E-04	6.03	1.50E-04
	5	23.52	0.00E+00	12.63	1.03E-03	15.60	5.00E-05	7.42	7.50E-05
12	2	30.96	7.50E-05	11.05	1.05E-03	22.46	1.50E-04	6.10	7.50E-05
	3	28.00	2.00E-04	13.66	1.03E-03	21.57	1.50E-04	7.00	2.50E-05
	4	31.14	2.50E-05	13.07	7.25E-04	21.22	0.00E+00	6.79	5.00E-05
	5	23.56	0.00E+00	16.45	9.75E-04	16.57	0.00E+00	8.20	0.00E+00
<b>Promedio</b>		<b>24.43</b>	<b>2.34E-05</b>	<b>11.36</b>	<b>6.98E-04</b>	<b>18.30</b>	<b>7.66E-05</b>	<b>7.13</b>	<b>4.53E-05</b>

Tabla 4.4: Resultados de las heurísticas en  $n$  instancias medianas con parámetro de  $a = 2$ 

$n$	$m$	C1		C2		C3		C4	
		Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo
20	2	32.51	0.00000	15.91	0.00130	29.18	0.00003	13.31	0.00020
	3	34.41	0.00000	18.18	0.00153	29.44	0.00010	13.58	0.00003
	4	34.86	0.00003	20.85	0.00180	30.96	0.00007	16.75	0.00007
	5	34.25	0.00000	21.76	0.00123	28.90	0.00007	17.68	0.00000
30	2	35.55	0.00000	18.00	0.00083	32.77	0.00007	14.79	0.00003
	3	38.01	0.00033	18.84	0.00143	34.03	0.00003	17.11	0.00007
	4	37.98	0.00000	21.70	0.00170	32.76	0.00000	18.58	0.00003
	5	38.70	0.00057	22.96	0.00217	35.06	0.00007	19.59	0.00013
40	2	37.02	0.00000	17.85	0.00127	34.40	0.00017	17.84	0.00033
	3	40.17	0.00000	22.29	0.00010	35.96	0.00003	19.19	0.00010
	4	41.85	0.00000	25.11	0.00223	36.26	0.00033	20.70	0.00023
	5	40.30	0.00000	23.29	0.00337	35.14	0.00000	19.31	0.00007
<b>Promedio</b>		<b>37.13</b>	<b>0.00008</b>	<b>20.56</b>	<b>0.00158</b>	<b>32.90</b>	<b>0.00008</b>	<b>17.37</b>	<b>0.00011</b>

En las 4 tablas, tanto en las desviaciones promedio como en los promedios finales de las heurísticas, podemos observar que en la heurística 4 la desviación promedio es la más baja de las 4 heurísticas, seguida de la 2, pero la diferencia de tiempos de CPU (en segundos) entre ambas es considerable, por lo que podemos concluir que la heurística 4 es la mejor de todas por su desviación pequeña y su tiempo de ejecución corto, y además sus tiempos de ejecución no incrementan tanto respecto al tiempo de las otras heurísticas.

Del análisis de las 4 tablas, se observa que la heurística 4 alcanza el mejor desempeño con el parámetro  $a = 4$ , lo mismo pasa en general para las otras heurísticas. Esto refleja que es

Tabla 4.5: Resultados de las heurísticas en  $n$  instancias medianas con parámetro de  $a = 4$ 

$n$	$m$	C1		C2		C3		C4	
		Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo
20	2	32.50	0.00010	15.79	0.00117	26.02	0.00003	10.62	0.00010
	3	33.51	0.00000	17.48	0.00160	28.39	0.00000	10.60	0.00007
	4	32.28	0.00007	18.65	0.00183	24.57	0.00010	12.82	0.00017
	5	31.36	0.00000	22.22	0.00127	27.74	0.00000	13.86	0.00020
30	2	36.54	0.00007	17.79	0.00010	30.37	0.00000	13.27	0.00020
	3	36.73	0.00007	19.41	0.00007	28.50	0.00003	15.06	0.00023
	4	36.25	0.00000	21.99	0.00017	27.26	0.00020	13.84	0.00017
	5	40.12	0.00000	22.61	0.00080	31.43	0.00000	14.98	0.00027
40	2	37.85	0.00000	18.26	0.00177	32.43	0.00007	13.72	0.00023
	3	40.28	0.00010	22.46	0.00243	32.99	0.00017	17.22	0.00020
	4	41.82	0.00030	23.72	0.00310	31.28	0.00003	16.07	0.00050
	5	40.44	0.00000	24.03	0.00340	31.95	0.00017	17.48	0.00033
<b>Promedio</b>		<b>36.64</b>	<b>0.00006</b>	<b>20.37</b>	<b>0.00148</b>	<b>29.41</b>	<b>0.00007</b>	<b>14.13</b>	<b>0.00022</b>

importante realizar un ajuste de dicho parámetro para obtener el valor que nos de el mejor desempeño. Cabe destacar que la desviación global (**Promedio**) de la heurística 4 con  $a = 4$  tiene un valor de 7.13 y 14.13 en instancias pequeñas y medianas, respectivamente. Mientras que la máxima desviación promedio por grupo es de 9.82 y 17.48 en las instancias pequeñas y medianas, respectivamente. Podemos decir que estos valores son considerablemente buenos para una heurística.

### 4.3.2. En instancias grandes

Para las instancias grandes comparamos el desempeño de las heurísticas únicamente entre las soluciones que se obtienen de ellas, ya que para este caso no se conocen soluciones óptimas.

En las tabla 4.6 y 4.7 se presentan las comparativas de los constructivos respecto a las mejores soluciones encontradas en instancias grandes para los valores de  $a = 2$  y  $a = 4$ . En las dos tablas, las columnas 1 y 2, hacen referencia al tamaño de la instancia, es decir, número de tareas ( $n$ ) y número de máquinas ( $m$ ), respectivamente. En las columnas 3, 5, 7, y 9 contienen la desviación promedio de las soluciones obtenidas con las heurísticas C1, C2, C3 y C4, respectivamente, respecto a las mejores soluciones encontradas. Mientras que en las columnas 4, 6, 8, y 10 contienen el tiempo promedio de cada grupo de instancias por cada valor de  $n$  y  $m$ , para las cuatro heurísticas. Al final de la tabla se encuentra una línea con el nombre de "Promedio" donde podemos encontrar el promedio global de las desviaciones y tiempos de cada una de las heurísticas respectivamente.

De los resultados de las tabla 4.6 y 4.7 podemos observar que se mantiene el mismo comportamiento de las heurísticas, es decir, la heurística 4 tiene la desviación promedio más baja de las 4 heurísticas, seguida de la 2, y además sus tiempos de ejecución no incrementan tanto respecto al tiempo de las otras heurísticas. También se observa que la heurística 4, el mejor desempeño lo alcanza con el parámetro  $a = 4$ , lo mismo pasa con la heurística 3 mientras al contrario ocurre con las heurísticas 1 y 2. Esto refleja que el mejor desempeño de la heurística 4 puede afectar las desviaciones de las otras heurísticas.

Tabla 4.6: Resultados de las heurísticas en  $n$  instancias grandes con parámetro de  $a = 2$ 

$n$	$m$	C1		C2		C3		C4	
		Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo
50	10	26.82	0.0004	6.81	0.0060	20.82	0.0005	0.42	0.0011
	15	20.51	0.0008	7.33	0.0072	16.93	0.0001	0.81	0.0001
	20	20.32	0.0003	8.83	0.0064	12.88	0.0012	0.56	0.0004
	25	14.93	0.0000	9.78	0.0050	9.25	0.0011	1.11	0.0001
	30	11.60	0.0009	9.82	0.0070	8.70	0.0009	1.30	0.0008
100	10	29.61	0.0008	5.58	0.0089	22.83	0.0005	0.23	0.0017
	15	27.11	0.0013	5.22	0.0084	19.59	0.0016	0.47	0.0017
	20	25.64	0.0004	5.61	0.0091	17.58	0.0002	0.30	0.0032
	25	22.50	0.0005	7.41	0.0062	14.89	0.0014	0.12	0.0030
	30	19.38	0.0004	8.04	0.0017	13.70	0.0026	0.07	0.0035
150	10	32.04	0.0008	5.42	0.0055	25.08	0.0028	0.09	0.0059
	15	29.02	0.0016	7.83	0.0045	21.90	0.0033	0.00	0.0048
	20	27.36	0.0032	6.32	0.0054	19.52	0.0020	0.20	0.0048
	25	26.42	0.0000	8.10	0.0034	18.90	0.0042	0.17	0.0063
	30	23.66	0.0016	7.38	0.0038	17.08	0.0037	0.08	0.0032
200	10	33.16	0.0016	4.69	0.0078	27.52	0.0041	0.15	0.0142
	15	30.77	0.0006	5.83	0.0085	22.87	0.0044	0.07	0.0102
	20	28.53	0.0024	7.65	0.0076	21.60	0.0046	0.00	0.0095
	25	28.09	0.0025	6.62	0.0076	21.37	0.0046	0.18	0.0089
	30	26.90	0.0073	8.49	0.0069	19.10	0.0040	0.00	0.0081
250	10	33.42	0.0044	5.13	0.0141	27.04	0.0042	0.04	0.0246
	15	30.85	0.0044	5.89	0.0125	24.58	0.0081	0.09	0.0196
	20	29.79	0.0030	6.27	0.0120	21.07	0.0063	0.02	0.0194
	25	28.53	0.0058	7.93	0.0116	21.00	0.0078	0.00	0.0149
	30	27.65	0.0033	7.77	0.0113	18.85	0.0091	0.10	0.0156
<b>Promedio</b>		<b>22.18</b>	<b>0.0019</b>	<b>7.03</b>	<b>0.0075</b>	<b>19.39</b>	<b>0.0033</b>	<b>0.26</b>	<b>0.0074</b>

Cabe destacar que la desviación global (**Promedio**) de la heurística 4 tiene un valor de 0.10, mientras que la máxima desviación promedio es de 0.66 en las instancias grandes. Podemos decir que estos valores son considerablemente menores que los obtenidos por las otras heurísticas, lo que indica que dicha heurística obtiene las mejores soluciones encontradas en la mayoría de las instancias.

En conclusión, para las instancias pequeñas, medianas y grandes, el método con menor tiempo promedio de ejecución resultó ser la heurística 4 siendo esta también la que menos desviación promedio tiene de todas.

Tabla 4.7: Resultados de las heurísticas en  $n$  instancias grandes con parámetro de  $a = 4$ 

$n$	$m$	C1		C2		C3		C4	
		Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo	Desv. Prom.	Tiempo
50	10	29.94	0.0002	10.06	0.0056	18.81	0.0005	0.00	0.0014
	15	23.72	0.0001	9.47	0.0068	13.47	0.0007	0.59	0.0021
	20	17.39	0.0004	9.56	0.0068	10.94	0.0003	0.33	0.0005
	25	16.27	0.0000	9.39	0.0073	9.30	0.0012	0.48	0.0008
	30	12.35	0.0001	10.37	0.0058	6.51	0.0011	0.66	0.0002
100	10	34.01	0.0003	8.74	0.0049	21.65	0.0017	0.04	0.0024
	15	29.87	0.0004	10.61	0.0017	19.70	0.0025	0.08	0.0017
	20	27.89	0.0008	11.52	0.0015	15.40	0.0022	0.00	0.0036
	25	25.25	0.0023	11.40	0.0009	14.37	0.0033	0.18	0.0021
	30	23.32	0.0001	11.17	0.0019	13.00	0.0016	0.02	0.0005
150	10	35.43	0.0009	10.20	0.0026	22.44	0.0053	0.00	0.0086
	15	33.32	0.0006	11.97	0.0050	19.24	0.0032	0.04	0.0056
	20	31.48	0.0027	12.48	0.0036	19.91	0.0059	0.00	0.0061
	25	29.58	0.0025	12.56	0.0026	16.42	0.0044	0.00	0.0040
	30	27.85	0.0021	11.72	0.0029	15.24	0.0050	0.00	0.0057
200	10	36.08	0.0009	9.66	0.0095	25.39	0.0047	0.00	0.0238
	15	34.52	0.0026	11.80	0.0059	22.99	0.0079	0.00	0.0216
	20	32.79	0.0027	12.61	0.0059	19.17	0.0102	0.00	0.0181
	25	31.41	0.0025	12.03	0.0048	18.44	0.0091	0.00	0.0158
	30	31.17	0.0030	13.48	0.0092	16.44	0.0086	0.00	0.0139
250	10	36.24	0.0040	9.21	0.0134	24.86	0.0086	0.00	0.0446
	15	35.02	0.0046	10.88	0.0121	21.84	0.0113	0.00	0.0365
	20	33.22	0.0043	11.55	0.0085	19.98	0.0117	0.00	0.0295
	25	32.47	0.0073	12.40	0.0107	18.22	0.0128	0.00	0.0272
	30	31.94	0.0066	13.31	0.0063	17.71	0.0149	0.00	0.0264
<b>Promedio</b>		<b>29.30</b>	<b>0.0021</b>	<b>11.13</b>	<b>0.0058</b>	<b>17.66</b>	<b>0.0055</b>	<b>0.10</b>	<b>0.0121</b>



# Capítulo 5

## Conclusiones y trabajo futuro

Se propuso un modelo entero mixto para el problema de secuenciación en máquinas paralelas con tiempos de preparación dependientes de la secuencia y la minimización del tiempo total de terminación. El modelo puede resolver instancias de tamaños considerables respecto a las instancias que se resuelven en la literatura con otros modelos.

Dado que el problema es NP-hard y que el optimizador comercial no puede resolver instancias grandes se propusieron cuatro heurísticas constructivas para encontrar soluciones factibles.

De la experimentación computacional, podemos concluir que la combinación de selección y evaluación que conforman la heurística 4 es la mejor de las 4 posibles combinaciones, donde, la selección consistió en elegir aleatoriamente  $a$  elementos para conformar la RCL, evaluar la contribución de los elementos como el menor incremento en el TCT, y elegir aquel con el menor incremento.

Las heurísticas cuya forma de selección se basan en reglas de prioridad son superadas por la forma de selección aleatoria y vale la pena el tiempo invertido de programación en la forma de evaluación del TCT ya que se obtienen resultados satisfactorios en comparación con una evaluación basada en el makespan.

Como trabajo a futuro se podría considerar comparar con una heurística tipo GRASP, agregar un procedimiento de búsqueda local para mejorar las soluciones construidas, explorar modificaciones del modelo para otras funciones objetivo, o minimizar el TCT en otros ambientes de producción.

Además, parece haber una estrecha relación entre la eficiencia del modelo matemático y la eficiencia de la heurística 4 en relación al modo de evaluación de la función objetivo. Por lo que también sería interesante ver si esa misma relación se mantiene cuando se consideran otras funciones objetivo.

# Bibliografía

- Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research*, 246(2):345–378.
- Allahverdi, A., Gupta, J. N., y Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2):219–239.
- Allahverdi, A., Ng, C., Cheng, T. E., y Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187(3):985–1032.
- Angel-Bello, F., Alvarez, A., y García, I. (2013). Two improved formulations for the minimum latency problem. *Applied Mathematical Modelling*, 37(4):2257–2266.
- Angel-Bello, F., Cardona-Valdés, Y., y Álvarez, A. (2017). Mixed integer formulations for the multiple minimum latency problem. *Operational Research*, pages 1–30.
- Bigras, L.-P., Gamache, M., y Savard, G. (2008). The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization*, 5(4):685–699.
- Fischetti, M., Laporte, G., y Martello, S. (1993). The delivery man problem and cumulative matroids. *Operations Research*, 41(6):1055–1064.
- Fowler, J., Horng, S. M., y Cochran, J. K. (2003). A hybridized genetic algorithm to solve parallel machine scheduling problems with sequence dependent setups. *International Journal of Industrial Engineering: Theory Applications and Practice*, 10(3):232–243.
- Gouveia, L. y Voß, S. (1995). A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research*, 83(1):69–82.
- Lucena, A. (1990). Time-dependent traveling salesman problem—the deliveryman case. *Networks*, 20(6):753–763.
- Méndez-Díaz, I., Zabala, P., y Lucena, A. (2008). A new formulation for the traveling deliveryman problem. *Discrete Applied Mathematics*, 156(17):3223–3237.
- Nessah, F., Yalaoui, F., y Chu, C. (2005). New heuristics for identical parallel machine scheduling with sequence dependent setup times and dates. In *Proceedings of the international conference on industrial engineering and systems management, Marrakech, Morocco*, pages 32–41.

- Nucamendi-Guillén, S., Martínez-Salazar, I., Angel-Bello, F., y Moreno-Vega, J. M. (2016). A mixed integer formulation and an efficient metaheuristic procedure for the k-travelling repairmen problem. *Journal of the Operational Research Society*.
- Picard, J.-C. y Queyranne, M. (1978). The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110.
- Pinedo, M. (2005). *Planning and scheduling in manufacturing and services*. Springer.
- Pinedo, M. L. (2016). *Scheduling: theory, algorithms, and systems*. Springer.
- Queyranne, M. y Wang, Y. (1991). Single-machine scheduling polyhedra with precedence constraints. *Mathematics of Operations Research*, 16(1):1–20.
- Taha, H. A. (2012). *Investigación de operaciones*. Pearson Educación, 9na edition.
- Weng, M. X., Lu, J., y Ren, H. (2001). Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *International journal of production economics*, 70(3):215–226.