

Trabajo Fin de Grado

Grado en Ingeniería de Organización Industrial

Modelos de Programación Lineal para flowshop de permutación con restricciones de disponibilidad

Autor: María del Carmen Barberá Quijano

Tutor: Paz Pérez González

Dep. de Organización Industrial y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de Organización Industrial

Modelos de Programación Lineal para flowshop de permutación con restricciones de disponibilidad

Autor:

María del Carmen Barberá Quijano

Tutor:

Paz Pérez González

Dep. de Organización Industrial y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Modelos de Programación Lineal para flowshop de permutación con restricciones de disponibilidad

Autor: María del Carmen Barberá Quijano
Tutor: Paz Pérez González

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Índice

1	Introducción	1
1.1	Objetivos del Proyecto	1
1.2	Sumario	2
2	Descripción del problema	3
2.1	Introducción	3
2.2	Descripción del problema	3
3	Modelos	5
3.1	Introducción	5
3.2	Datos, índices y variables empleados	5
3.2.1	Datos	5
3.2.2	Índices	6
3.2.3	Variables	6
3.3	Modelos originales y sus adaptaciones	7
3.3.1	Familia de modelos Wagner	7
	Modelo WST	7
	Modelo WST adaptado	8
	Modelo Wilson	10
	Modelo Wilson adaptado	11
	Modelo TS2	12
	Modelo TS2 adaptado	12
3.3.2	Familia de modelos Manne	14
	Modelo SGST	14
	Modelo SGST adaptado	14
	Modelo LYeq	15
	Modelo LYeq adaptado	15
3.3.3	Modelos formulados por Baker (2013)	16
	Idle-time version	16
	Idle-time version adaptado	18
	Completion-time version	19
	Completion-time version adaptado	19
4	Análisis	21
4.1	Resolución del modelo	21
4.2	Análisis estadístico de los resultados	21
4.2.1	Influencia del Modelo en el CPU Time	22
4.2.2	Influencia del periodo de disponibilidad (T) en el <i>CPU time</i>	24
4.2.3	Influencia del modelo y el periodo de disponibilidad (T) en el <i>CPU Time</i>	25
4.2.4	Influencia del modelo y el número de trabajos (N) en el CPU Time	25

4.2.5	Influencia del modelo y el número de máquinas (M) en el <i>CPU Time</i>	27
5	Conclusiones	31
Apéndice A	Anexo	33
A.0.1	Modelo WST	33
A.0.2	Modelo Wilson	37
A.0.3	Modelo TS2	41
A.0.4	Modelo Idle-time	45
A.0.5	Modelo Completion-time	49
A.0.6	Modelo SGST	53
A.0.7	Modelo LYeq	56
	<i>Índice de Figuras</i>	61
	<i>Índice de Tablas</i>	63
	<i>Bibliografía</i>	65

1 Introducción

En la mayoría de los sectores empresariales existe un problema común: la asignación de los recursos existentes a las tareas cuya realización es necesaria para el correcto funcionamiento de la empresa. Este problema es lo que conocemos como programación y su resolución irá encaminada a conseguir unos objetivos preestablecidos de forma óptima.

Existen numerosos enfoques para la programación si partimos de la definición anterior, y este trabajo se centra en uno de ellos, la programación de la producción.

El problema que se plantea es el siguiente: disponemos de un número de máquinas (recursos), las cuales trabajan en un entorno de tipo *flowshop*, que se destinan al procesamiento de una serie de trabajos. El objetivo de esta asignación será minimizar el retraso de todos los trabajos (*tardiness*) con respecto a su fecha de entrega, cuyo cumplimiento no es considerado estrictamente necesario. Asimismo, existen una serie de limitaciones a la hora de resolver este problema: existen periodos de indisponibilidad de las máquinas y los trabajos se consideran ininterrumpibles, por lo que la solución óptima debe cumplir con estos aspectos.

A lo largo de la literatura, se han estudiado múltiples problemas de programación de la producción. Algunos de ellos se emplean como base a la hora de resolver la problemática expuesta, realizando adaptaciones según el objetivo y las restricciones que consideramos.

Para crear este documento, se ha optado por emplear el sistema de composición de textos *LaTeX*. Este sistema presenta numerosas ventajas, como la calidad en la edición de ecuaciones y la facilidad al estructurar el texto, razones que han provocado su elección frente a otros procesadores de texto.

1.1 Objetivos del Proyecto

El objetivo último de este proyecto es la caracterización de un modelo que logre resolver de forma óptima el problema presentado en la introducción. Este modelo, además de proporcionar la mejor solución, debe hacerlo en el menor tiempo posible. Por consiguiente, realizaremos un análisis del desempeño de todos los modelos para alcanzar así la meta final, encontrar el modelo más eficiente.

Podemos desglosar este objetivo en las siguientes fases que desembocan en su consecución:

1. Adaptar los modelos existentes en la literatura para el problema propuesto.
2. Programar los modelos para su posterior resolución.
3. Resolver los modelos para un número pequeño de instancias mediante el software *Gurobi*.
4. Validar los modelos.
5. Resolver de nuevo los modelos para una batería de instancias.
6. Evaluar los tiempos de cómputo para cada uno de los modelos.

7. Analizar los resultados y determinar el modelo más eficiente. En este punto logramos el alcance del objetivo final.

1.2 Sumario

Este primer capítulo introduce el propósito del proyecto, es decir, el problema abordado y la metodología empleada.

El capítulo 2 proporciona los conceptos básicos necesarios para el entendimiento de la problemática expuesta, así como su descripción detallada.

El capítulo 3 describe la metodología empleada para la resolución del problema. En este caso, se presentan los modelos originales y sus adaptaciones para que cumplan con el objetivo y las restricciones propuestas.

El capítulo 4 presenta un análisis del desempeño de los modelos, empleando como variable de medida el tiempo de cómputo requerido en cada uno de ellos.

2 Descripción del problema

2.1 Introducción

Este apartado ofrece una visión general de los conceptos básicos necesarios en la definición de problemas de programación de la producción.

En primer lugar, se definen algunos términos generales Framinan et al. (2014):

- Máquina: Recurso productivo con capacidad de realizar operaciones de transformación o transporte de material.
- Trabajo: Producto que es objeto de una operación.
- Tiempo de proceso: Duración temporal de la operación de transformación de un trabajo en una máquina.

El objetivo de este problema consiste en la asignación en la escala temporal de una serie de trabajos a las máquinas de las que disponemos. El resultado de este proceso es un programa que especifica qué máquina procesará cada trabajo y en qué instante lo hará.

Para alcanzar la solución a un problema de programación del mundo real, lo abstraemos y simplificamos formalmente mediante un modelo de programación. Existe una gran diversidad de modelos, por lo que relacionar aquellos que presenten similitudes puede resultar muy útil a la hora de su resolución. Por este motivo, Graham et al. (1979) presentó una clasificación de modelos según tres aspectos:

- Entorno (α): Indica el número de máquinas y su disposición en la fábrica.
- Restricciones (β): Limitaciones del proceso sobre las máquinas o los trabajos. Podemos considerar restricciones sobre la precedencia, las fechas de llegada o entrega de los trabajos, la existencia de tiempos de setup, etc. Existen unas suposiciones generales que se consideran sin necesidad de ser indicadas en las restricciones:
 - Los trabajos, disponibles al principio del horizonte de programación, se consideran ininterrumpibles.
 - Una máquina no puede procesar más de un trabajo simultáneamente, al igual que un trabajo no puede ser tratado por más de una máquina en un mismo instante.
 - El búfer entre máquinas se supone infinito y el tiempo de transporte despreciable.
- Criterio (γ): Función objetivo del problema.

El problema queda caracterizado mediante el formato: $\alpha|\beta|\gamma$

2.2 Descripción del problema

Utilizando la clasificación descrita en el apartado 2.1, el problema estudiado en este proyecto se caracteriza a continuación:

- Entorno: $\alpha = F_m$. Las máquinas forman un entorno de tipo taller de flujo (*flowshop*), en el que todos los trabajos siguen la misma ruta predeterminada.

- Restricciones: $\beta = prmu, nr - pm$. Además de las suposiciones generales descritas en el apartado 2.1, existen dos limitaciones en este problema:
 - $prmu$. Esta restricción, específica para entornos de tipo *flowshop*, impone que la secuencia de procesamiento de trabajos sea la misma en todas las máquinas.
 - $nr - pm$. Define la existencia de periodos de indisponibilidad de las máquinas, generalmente debidos a mantenimiento preventivo, tras los cuales los trabajos ya comenzados no podrán reanudarse. Consideramos que las máquinas están disponibles durante un periodo T , tras el cual, se produce una parada cuya duración es considerada nula ya que no resulta significativa en la resolución del problema.

A continuación, se ejemplifica la influencia de la restricción $nr - pm$ en una instancia pequeña formada por 3 máquinas y 3 trabajos. En el primer caso, figura 2.1, no existen periodos de indisponibilidad de las máquinas. En cambio, en el segundo caso (figura 2.2), las máquinas estarán disponibles durante un periodo de tiempo constante T , tras el cual sufren una parada. Para este ejemplo, el valor de T es de 4 u.t. De la existencia de paradas se derivan los *bines*, que se definen como los periodos de tiempo existentes entre dos paradas consecutivas. Los trabajos se consideran ininterrumpibles, por lo que únicamente comenzarán a procesarse en un *bin* si pueden terminar su procesado en él. En caso contrario, se programarán en el siguiente *bin*. En la figura 4.2 observamos que el trabajo 3, no comienza en el *bin*1 ya que su duración es de 3 u.t. y disponemos de 1 u.t antes de la primera parada. Esto ocurre para este mismo trabajo en la máquina 2. Por tanto, se han necesitado 3 *bines* para finalizar los 3 trabajos.

Observamos que la existencia de *bines* aumenta el tiempo de procesado de los trabajos, y, por tanto, la complejidad del problema.

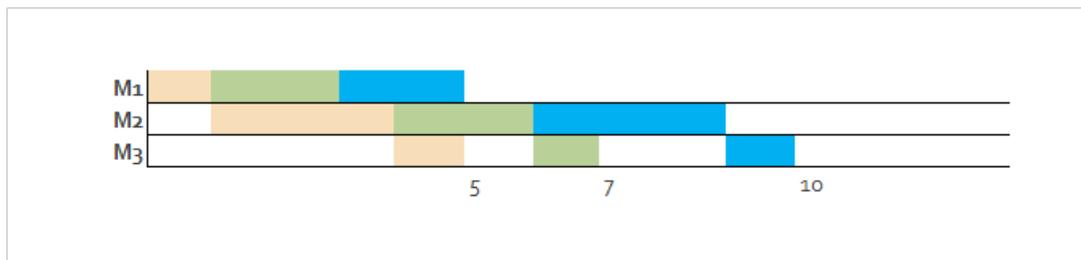


Figura 2.1 Diagrama de Gantt sin restricción de disponibilidad en las máquinas.

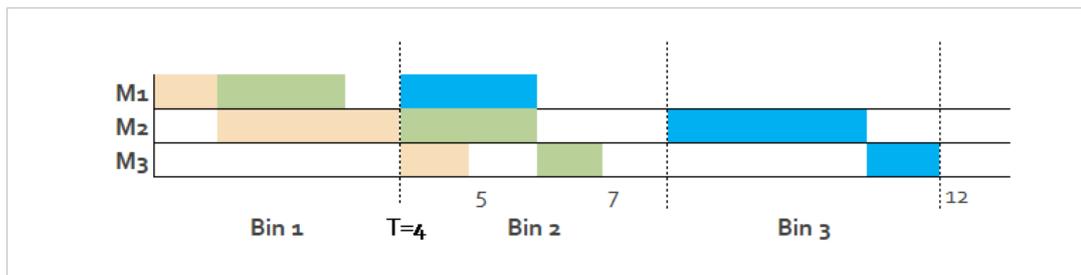


Figura 2.2 Diagrama de Gantt con restricción de disponibilidad en las máquinas.

- Criterio: $\gamma = \sum_j T_j$. Podemos medir numerosos parámetros respecto al desempeño de los trabajos en las máquinas. En este caso, tendremos en cuenta el *tardiness* o retraso de los trabajos respecto a su fecha de entrega (en caso de adelanto será nulo). La solución óptima será aquella que proporcione el menor valor del *tardiness* total.

De esta forma, el problema queda simplificado de la siguiente forma: $F_m | prmu, nr - pm | \sum_j T_j$

3 Modelos

3.1 Introducción

En este capítulo se exponen:

- Los datos, variables e índices empleados en la descripción de los modelos
- Los modelos de programación lineal entera originales junto con las adaptaciones realizadas para que resuelvan el problema descrito: $F_m \mid prmu, nr - pm \mid \sum_j T_j$

Los modelos originales proceden de dos fuentes diferentes. Cinco de ellos fueron evaluados y comparados por Stafford et al. (2005) para el problema $F_m \mid prmu \mid C_{max}$ empleando dos enfoques: dar a la variable de decisión X_{jk} el valor 1 si el trabajo j aparece en la posición k de la secuencia, o en el caso de que el trabajo j preceda al trabajo k . El tiempo de cálculo en el primer caso fue menor, por lo que la segunda fuente, Baker (2013) optó por esta formulación.

Los modelos de Stafford et al. (2005) fueron adaptados por Ramos Salgado (2017) en su Trabajo de Fin de Grado para que cumplieren con la restricción de periodo de indisponibilidad, $nr - pm$, pero minimizando el *makespan*. Por tanto, será necesario adaptarlos para minimizar el *tardiness* total.

Los modelos propuestos por Baker (2013) tienen como objetivo minimizar el *tardiness*, pero sin incluir la restricción de tiempo de indisponibilidad. En este caso, no modificaremos el objetivo, sino añadiremos esta limitación adicional.

De este modo, los siete modelos son adaptados para resolver el problema $F_m \mid prmu, nr - pm \mid \sum_j T_j$.

El procedimiento para obtener la solución óptima, a partir de los modelos ya adaptados, ha sido el siguiente:

1. Desarrollo de un programa en lenguaje C que genere los modelos de programación lineal para cada instancia concreta. El código en lenguaje C para cada uno de los modelos se encuentra en el Anexo.
2. Utilización del software *Gurobi* para obtener la solución óptima para cada instancia.

3.2 Datos, índices y variables empleados

En esta sección se muestra la notación empleada para los datos, conjuntos de índices y variables que aparecen en los modelos. Se ha optado por mantener la notación propuesta por cada autor, en lugar de unificarla, para facilitar la consulta de las referencias originales y el entendimiento de los modelos. Por tanto, clasificamos los datos, índices y variables en tres tipos: comunes a todos los modelos, presentes en los modelos de Stafford et al. (2005) y utilizados por Baker (2013).

3.2.1 Datos

- Comunes a todos los modelos:

M , número de máquinas.

N , número de trabajos.

P , valor suficientemente grande para garantizar que únicamente se cumple una del par de restricciones dicotómicas. En este caso le asignamos el valor de la suma de los tiempos de proceso de todos los trabajos en todas las máquinas, es decir, $P = \sum_{r=1}^M \sum_{i=1}^N T_{ri}$.

$Bines$, cota superior del número de bins. Este valor se calcula a partir de los valores de P y T , según $Bines = \lceil \frac{P}{T} \rceil$.

T , periodo de disponibilidad. Tiempo constante que transcurre entre dos paradas consecutivas de las máquinas.

- Para los modelos propuestos por Stafford et al. (2005):

T_{ri} , tiempo de proceso del trabajo i en la máquina r .

d_i , fecha de entrega del trabajo i .

- Para los modelos propuestos por Baker (2013):

p_{ij} , tiempo de proceso del trabajo j en la máquina i .

d_j , fecha de entrega del trabajo j .

3.2.2 Índices

- Comunes a todos los modelos:

b para los bins en el horizonte temporal ($1 \leq b \leq Bines$).

- Para los modelos propuestos por Stafford et al. (2005):

r para las máquinas ($1 \leq r \leq M$).

i, k , para los trabajos ($1 \leq i \leq k \leq N$).

j para la posición en la secuencia ($1 \leq j \leq N$).

- Para los modelos propuestos por Baker (2013):

i para las máquinas ($1 \leq i \leq M$).

j para los trabajos ($1 \leq j \leq N$).

k para la posición en la secuencia ($1 \leq k \leq N$).

3.2.3 Variables

- Para los modelos propuestos por Stafford et al. (2005):

$B_{r,j}$, tiempo de comienzo del trabajo en la posición j de la secuencia en la máquina r .
($1 \leq r \leq M, 1 \leq j \leq N$)

$$Bin_{r,i(j),b} = \begin{cases} 1 & \text{Si el trabajo } i \text{ (que está en la posición } j \text{ de la secuencia)} \\ & \text{es procesado en la máquina } r \text{ y en el bin } b \\ 0 & \text{En otro caso} \end{cases}$$

($1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq Bines$)

C_{ri} , tiempo de finalización del trabajo i en la máquina r ($1 \leq r \leq M, 1 \leq j \leq N$).

$$D_{i,k} = \begin{cases} 1 & \text{Si el trabajo } i \text{ es procesado antes que el trabajo } k \\ 0 & \text{En otro caso} \end{cases}$$

($1 \leq i \leq k \leq N$)

$E_{r,j}$, tiempo de finalización del trabajo en la posición j en la máquina r ($1 \leq r \leq M, 1 \leq j \leq N$).

$S_{r,i}$, tiempo de comienzo del trabajo i en la máquina r ($1 \leq r \leq M, 1 \leq i \leq N$).

$X_{r,j}$, tiempo ocioso en la máquina r antes del comienzo del procesamiento del trabajo en la posición j de la secuencia ($1 \leq r \leq M, 1 \leq j \leq N$).

$Y_{r,j}$, tiempo de espera del trabajo en la posición j de la secuencia después de ser procesado en la máquina r ($1 \leq r \leq M, 1 \leq j \leq N$).

$$Z_{i,j} = \begin{cases} 1 & \text{Si el trabajo } i \text{ es asignado a la posición } j \text{ de la secuencia} \\ 0 & \text{En otro caso} \end{cases}$$

($1 \leq i \leq N, 1 \leq j \leq N$)

d_j , fecha de entrega del trabajo en la posición j de la secuencia ($1 \leq j \leq N$).

td_j , tardiness del trabajo en la posición j de la secuencia ($1 \leq j \leq N$).

- Para los modelos propuestos por Baker (2013):

$$Bin_{i,j(k),b} = \begin{cases} 1 & \text{Si el trabajo } j \text{ (que está en la posición } k \text{ de la secuencia)} \\ & \text{es procesado en la máquina } i \text{ y en el bin } b \\ 0 & \text{En otro caso} \end{cases}$$

($1 \leq i \leq M, 1 \leq j \leq N, 1 \leq k \leq N, 1 \leq b \leq Bines$)

C_{ik} , tiempo de finalización del trabajo que se encuentra en la posición k de la secuencia en la máquina i ($1 \leq r \leq M, 1 \leq j \leq N$).

$C_{i,k}$, tiempo de finalización del trabajo en la posición k en la máquina i ($1 \leq i \leq M, 1 \leq k \leq N$).

$I_{i,k}$, tiempo ocioso en la máquina i antes del comienzo del procesamiento del trabajo en la posición k de la secuencia ($1 \leq i \leq M, 1 \leq k \leq N$).

$H_{k,i}$, tiempo de espera del trabajo en la posición k de la secuencia después de ser procesado en la máquina i ($1 \leq i \leq M, 1 \leq k \leq N$).

$$x_{j,k} = \begin{cases} 1 & \text{Si el trabajo } j \text{ es asignado a la posición } k \text{ de la secuencia} \\ 0 & \text{En otro caso} \end{cases}$$

($1 \leq j \leq N, 1 \leq k \leq N$)

d_k , fecha de entrega del trabajo en la posición k de la secuencia ($1 \leq k \leq N$).

td_k , tardiness del trabajo en la posición k de la secuencia ($1 \leq k \leq N$).

3.3 Modelos originales y sus adaptaciones

En este apartado se exponen los modelos de partida, las adaptaciones realizadas y los modelos resultantes que proporcionan la solución óptima al problema propuesto.

Como se indicó en el apartado 3.1, estos modelos proceden de dos fuentes diferentes: Stafford et al. (2005) y Baker (2013). Para los cinco modelos procedentes de la primera fuente, vamos a tener en cuenta las adaptaciones realizadas por el alumno Ramos Salgado (2017) en su Trabajo de Fin de Grado para que cumpliesen con la restricción de periodo de indisponibilidad, $nr - pm$. Estas restricciones estarán incluidas en los modelos originales. Adicionalmente, existe una división para los modelos propuestos por Stafford et al. (2005) en dos familias, cuyas características se describen en los subapartados 3.3.1 y 3.3.2.

3.3.1 Familia de modelos Wagner

WST, Wilson y TS2 son los modelos pertenecientes a la familia Wagner. Los tres modelos incorporan el problema de asignación, por lo que las variables se refieren a los trabajos por su posición en la secuencia (subíndice j).

Modelo WST

Este modelo emplea las variables X_{rj} , Y_{rj} , Z_{ij} , E_{rj} y Bin_{rjb} .

$$\text{Min } C_{max} = \sum_{i=1}^N T_{Mi} + \sum_{p=1}^N X_{Mp} \quad (3.1)$$

$$\text{sa : } \sum_{j=1}^N Z_{ij} = 1 \quad (1 \leq i \leq N) \quad (3.2)$$

$$\sum_{i=1}^N Z_{ij} = 1 \quad (1 \leq j \leq N) \quad (3.3)$$

$$\sum_{i=1}^N T_{ri}Z_{i,j+1} + X_{r,j+1} + Y_{r,j+1} = \sum_{i=1}^N T_{r+1,i}Z_{ij} + X_{r+1,j+1} + Y_{rj} \quad (1 \leq r \leq M-1, 1 \leq j \leq N-1) \quad (3.4)$$

$$X_{r+1,1} = X_{r,1} + Y_{r,1} + \sum_{i=1}^N T_{ri}Z_{i1} \quad (1 \leq r \leq M-1) \quad (3.5)$$

$$E_{rj} = \sum_{l=1}^j (X_{rl} + \sum_{i=1}^N T_{ri}Z_{il}) \quad (1 \leq r \leq M, 1 \leq j \leq N) \quad (3.6)$$

$$E_{rj} \leq bT + P(1 - \text{Bin}_{rjb}) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \quad (3.7)$$

$$E_{rj} - \sum_{i=1}^N T_{ri}Z_{ij} + P(1 - \text{Bin}_{rjb}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \quad (3.8)$$

$$\sum_{b=1}^{\text{Bines}} \text{Bin}_{rjb} = 1 \quad (1 \leq r \leq M, 1 \leq j \leq N) \quad (3.9)$$

$$X_{rj}, Y_{rj}, E_{rj} \geq 0; \quad 1 \leq r \leq M, 1 \leq j \leq N$$

$$Z_{ij}, \text{Bin}_{rjb} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}$$

Este modelo tiene como objetivo minimizar el *makespan*, el cual viene dado por la suma de los tiempos de proceso y los tiempos ociosos en la última máquina antes del procesamiento de cada trabajo (3.1).

Los conjuntos de restricciones (3.2 y 3.3) garantizan la asignación de cada trabajo a una única posición en la secuencia, así como que cada posición sea ocupada por un solo trabajo. Las restricciones (3.4 y 3.5) aseguran que el trabajo en la posición $j+1$ no empiece a ser procesado en la máquina r hasta que el trabajo en la posición j haya sido completamente procesado. Asimismo, garantizan que el trabajo en la posición j no comience a ser procesado en la máquina $r+1$ hasta que no finalice su procesamiento en la máquina r .

La variable E_{rj} , incorporada para facilitar la inclusión de la restricción de disponibilidad, se define como la suma de los tiempos de espera y de proceso de los trabajos (3.6). Las restricciones (3.7 y 3.8) proporcionan la cota superior del tiempo de finalización y la cota inferior del tiempo de comienzo de cada trabajo, respectivamente, para que sea procesado dentro del bin que le corresponda.

Finalmente, cada trabajo (referido por su posición j) podrá ser asignado a un único bin en cada máquina r (3.9).

Modelo WST adaptado

El modelo modificado incorpora la variable td_j y las siguientes restricciones:

$$td_j \geq E_{Mj} - d_j \quad (1 \leq j \leq n) \quad (3.10)$$

$$d_j = \sum_{i=1}^N d_i Z_{ij} \quad (1 \leq j \leq n) \quad (3.11)$$

La restricción (3.10) limita el *tardiness*, es decir, la diferencia entre el tiempo de finalización del trabajo en la posición j y su fecha de entrega, la cual se define en la restricción (3.11) teniendo en cuenta la variable de asignación de trabajos a posiciones en la secuencia.

El modelo adaptado quedaría de la siguiente forma:

$$\begin{aligned}
 & \text{Min } \sum_{j=1}^N td_j \\
 & \text{sa : } \sum_{j=1}^N Z_{ij} = 1 \quad (1 \leq i \leq N) \\
 & \quad \sum_{i=1}^N Z_{ij} = 1 \quad (1 \leq j \leq N) \\
 & \quad \sum_{i=1}^N T_{ri}Z_{i,j+1} + X_{r,j+1} + Y_{r,j+1} = \sum_{i=1}^N T_{r+1,i}Z_{ij} + \\
 & \quad \quad + X_{r+1,j+1} + Y_{r,j} \quad (1 \leq r \leq M-1, 1 \leq j \leq N-1) \\
 & \quad X_{r+1,1} = X_{r,1} + Y_{r,1} + \sum_{i=1}^N T_{ri}Z_{i1} \quad (1 \leq r \leq M-1) \\
 & \quad E_{rj} = \sum_{l=1}^j (X_{rl} + \sum_{i=1}^N T_{ri}Z_{il}) \quad (1 \leq r \leq M, 1 \leq j \leq N) \\
 & \quad E_{rj} \leq bT + P(1 - \text{Bin}_{rjb}) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \\
 & \quad E_{rj} - \sum_{i=1}^N T_{ri}Z_{ij} + P(1 - \text{Bin}_{rjb}) \geq T(b-1) \\
 & \quad \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \\
 & \quad \sum_{b=1}^{\text{Bines}} \text{Bin}_{rjb} = 1, \quad (1 \leq r \leq M, 1 \leq j \leq N) \\
 & \quad td_j \geq E_{Mj} - d_j \quad (1 \leq j \leq n) \\
 & \quad d_j = \sum_{i=1}^N d_i Z_{ij} \quad (1 \leq j \leq n) \\
 & \quad X_{rj}, Y_{rj}, E_{rj}, td_j \geq 0; \quad 1 \leq r \leq M, 1 \leq j \leq N \\
 & \quad Z_{ij}, \text{Bin}_{rjb} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}
 \end{aligned}$$

Modelo Wilson

Este modelo emplea las variables $B_{r,j}$, Z_{ij} y $Bin_{r,jb}$.

$$\text{Min } B_{MN} + \sum_{i=1}^N T_{Mi} Z_{iN} \quad (3.12)$$

$$\text{sa : } \sum_{j=1}^N z_{ij} = 1 \quad (1 \leq i \leq N) \quad (3.13)$$

$$\sum_{i=1}^N z_{ij} = 1 \quad (1 \leq j \leq N) \quad (3.14)$$

$$B_{1j} + \sum_{i=1}^N T_{1i} Z_{ij} \leq B_{1,j+1} \quad (1 \leq j \leq N-1) \quad (3.15)$$

$$B_{11} = 0 \quad (3.16)$$

$$B_{r1} + \sum_{i=1}^N T_{ri} Z_{i1} \leq B_{r+1,1} \quad (1 \leq r \leq M-1) \quad (3.17)$$

$$B_{rj} + \sum_{i=1}^N T_{ri} Z_{ij} \leq B_{r+1,j} \quad (1 \leq r \leq M-1, 2 \leq j \leq N) \quad (3.18)$$

$$B_{rj} + \sum_{i=1}^N T_{ri} Z_{ij} \leq B_{r,j+1} \quad (2 \leq r \leq M, 1 \leq j \leq N-1) \quad (3.19)$$

$$B_{rj} + \sum_{i=1}^N T_{ri} Z_{ij} - bT \leq P(1 - Bin_{r,jb}) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq Bines) \quad (3.20)$$

$$B_{rj} + P(1 - Bin_{r,jb}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq Bines) \quad (3.21)$$

$$\sum_{b=1}^{Bines} Bin_{r,jb} = 1, \quad (1 \leq r \leq M, 1 \leq j \leq N) \quad (3.22)$$

$$B_{r,j} \geq 0; \quad 1 \leq r \leq M, 1 \leq j \leq N$$

$$Z_{ij}, Bin_{r,jb} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq Bines$$

La función objetivo (*makespan*) se expresa adicionando al tiempo de comienzo del último trabajo en la última máquina su tiempo de proceso en esta máquina (3.12).

Los dos primeros conjuntos de restricciones (3.13 y 3.14) coinciden con las del modelo WST y tienen el mismo significado. La ecuación (3.15) fuerza que todos los trabajos comiencen en la primera máquina de forma inmediata a la finalización de su predecesor. En cuanto al primer trabajo, será procesado en el instante 0 en la primera máquina (3.16) y comenzará en el resto de máquinas tras ser procesado en la anterior (3.17). El conjunto de restricciones (3.18) actúa como la restricción (3.17) para todas las máquinas y trabajos. Asimismo, cada trabajo (referido por su posición $j+1$) no podrá ser procesado en una máquina hasta que acabe el trabajo j (3.19).

Las restricciones (3.20 y 3.21) proporcionan la cota superior del tiempo de finalización y la cota inferior del tiempo de comienzo de cada trabajo, respectivamente, para que sea procesado dentro del bin que le corresponda.

Finalmente, cada trabajo (referido por su posición j) podrá ser asignado a un único bin en cada máquina r (3.22).

Modelo Wilson adaptado

El modelo modificado incorpora la variable td_j y las siguientes restricciones:

$$td_j \geq B_{Mj} + \sum_{i=1}^N T_{Mi}Z_{ij} - d_j \quad (1 \leq j \leq n) \quad (3.23)$$

$$d_j = \sum_{i=1}^N d_i Z_{ij} \quad (1 \leq j \leq n) \quad (3.24)$$

La restricción (3.23) limita el *tardiness*, es decir, la diferencia entre el tiempo de finalización (en este caso, definido como la suma del tiempo de comienzo y el tiempo de proceso) del trabajo en la posición j y su fecha de entrega, la cual se define en la restricción (3.24) teniendo en cuenta la variable de asignación de trabajos a posiciones en la secuencia.

El modelo adaptado quedaría de la siguiente forma:

$$\begin{aligned} & \text{Min} \sum_{j=1}^N td_j \\ & \text{sa : } \sum_{j=1}^N z_{ij} = 1 \quad (1 \leq i \leq N) \\ & \quad \sum_{i=1}^N z_{ij} = 1 \quad (1 \leq j \leq N) \\ & \quad B_{1j} + \sum_{i=1}^N T_{1i}Z_{ij} \leq B_{1,j+1} \quad (1 \leq j \leq N-1) \\ & \quad B_{11} = 0 \\ & \quad B_{r1} + \sum_{i=1}^N T_{ri}Z_{i1} \leq B_{r+1,1} \quad (1 \leq r \leq M-1) \\ & \quad B_{rj} + \sum_{i=1}^N T_{ri}Z_{ij} \leq B_{r+1,j} \quad (1 \leq r \leq M-1, 2 \leq j \leq N) \\ & \quad B_{rj} + \sum_{i=1}^N T_{ri}Z_{ij} \leq B_{r,j+1} \quad (2 \leq r \leq M, 1 \leq j \leq N-1) \\ & \quad B_{rj} + \sum_{i=1}^N T_{ri}Z_{ij} - bT \leq P(1 - \text{Bin}_{rjb}) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \\ & \quad B_{rj} + P(1 - \text{Bin}_{rjb}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \\ & \quad \sum_{b=1}^{\text{Bines}} \text{Bin}_{rjb} = 1, \quad (1 \leq r \leq M, 1 \leq j \leq N) \\ & \quad td_j \geq B_{Mj} + \sum_{i=1}^N T_{Mi}Z_{ij} - d_j \quad (1 \leq j \leq n) \\ & \quad d_j = \sum_{i=1}^N d_i Z_{ij} \quad (1 \leq j \leq n) \\ & \quad B_{rj}, td_j \geq 0; \quad 1 \leq r \leq M, 1 \leq j \leq N \\ & \quad Z_{ij}, \text{Bin}_{rjb} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq \text{Bines} \end{aligned}$$

Modelo TS2

Este modelo emplea las variables E_{rj} , Z_{ij} y Bin_{rjb} .

$$\text{Min } E_{MN} \quad (3.25)$$

$$\text{sa : } \sum_{j=1}^N z_{ij} = 1 \quad (1 \leq i \leq N) \quad (3.26)$$

$$\sum_{i=1}^N z_{ij} = 1 \quad (1 \leq j \leq N) \quad (3.27)$$

$$E_{rj} + \sum_{i=1}^N T_{ri} Z_{i,j+1} \leq E_{r,j+1} \quad (1 \leq r \leq M, 1 \leq j \leq N-1) \quad (3.28)$$

$$E_{rj} + \sum_{i=1}^N T_{r+1,i} Z_{ij} \leq E_{r+1,j} \quad (1 \leq r \leq M-1, 1 \leq j \leq N) \quad (3.29)$$

$$E_{11} \geq \sum_{i=1}^N T_{1i} Z_{i1} \quad (3.30)$$

$$E_{rj} - bT \leq P(1 - Bin_{rjb}) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq Bines) \quad (3.31)$$

$$E_{rj} - \sum_{i=1}^N T_{ri} Z_{ij} + P(1 - Bin_{rjb}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq Bines) \quad (3.32)$$

$$\sum_{b=1}^{Bines} Bin_{rjb} = 1, \quad (1 \leq r \leq M, 1 \leq j \leq N) \quad (3.33)$$

$$E_{rj} \geq 0; \quad 1 \leq r \leq M, 1 \leq j \leq N$$

$$Z_{ij}, Bin_{rjb} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq Bines$$

Las restricciones son similares a las del modelo TS2, considerando en lugar del tiempo de inicio de cada trabajo (referido por su posición j) en cada máquina r , B_{rj} , el tiempo de finalización, E_{rj} .

La función objetivo (*makespan*) coincide con el tiempo de finalización del último trabajo en la última máquina (3.25).

Los dos primeros conjuntos de restricciones (3.26 y 3.27) coinciden con las de los dos modelos anteriores y tienen el mismo significado. La ecuación (3.28) fuerza que todos los trabajos finalicen en cada máquina tras finalizar su predecesor y ser procesado él mismo. El conjunto de restricciones (3.29) asegura para todos los trabajos que no puedan comenzar en la máquina $r+1$ hasta que finalicen en r . El primer trabajo en la primera máquina comenzará en el instante 0, por lo que su tiempo de finalización coincidirá con su tiempo de proceso (3.30).

Las restricciones (3.31 y 3.32) proporcionan la cota superior del tiempo de finalización y la cota inferior del tiempo de comienzo de cada trabajo, respectivamente, para que sea procesado dentro del bin que le corresponda. Finalmente, cada trabajo (referido por su posición j) podrá ser asignado a un único bin en cada máquina r (3.33).

Modelo TS2 adaptado

El modelo modificado incorpora la variable td_j y las siguientes restricciones:

$$td_j \geq E_{Mj} - d_j \quad (3.34)$$

$$d_j = \sum_{i=1}^N d_i Z_{ij} \quad (3.35)$$

La restricción (3.34) limita el *tardiness*, es decir, la diferencia entre el tiempo de finalización del trabajo en la posición j y su fecha de entrega, la cual se define en la restricción (3.35) teniendo en cuenta la variable de asignación de trabajos a posiciones en la secuencia.

El modelo adaptado quedaría de la siguiente forma:

$$\begin{aligned}
 & \text{Min } \sum_{j=1}^N td_j \\
 & \text{sa: } \sum_{j=1}^N z_{ij} = 1 \quad (1 \leq i \leq N) \\
 & \quad \sum_{i=1}^N z_{ij} = 1 \quad (1 \leq j \leq N) \\
 & \quad E_{rj} + \sum_{i=1}^N T_{ri} z_{i,j+1} \leq E_{r,j+1} \quad (1 \leq r \leq M, 1 \leq j \leq N-1) \\
 & \quad E_{rj} + \sum_{i=1}^N T_{r+1,i} z_{ij} \leq E_{r+1,j} \quad (1 \leq r \leq M-1, 1 \leq j \leq N) \\
 & \quad E_{11} \geq \sum_{i=1}^N T_{1i} z_{i1} \\
 & \quad E_{rj} - bT \leq P(1 - \text{Bin}_{rjb}) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \\
 & \quad E_{rj} - \sum_{i=1}^N T_{ri} z_{ij} + P(1 - \text{Bin}_{rjb}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}) \\
 & \quad \sum_{b=1}^{\text{Bines}} \text{Bin}_{rjb} = 1, \quad (1 \leq r \leq M, 1 \leq j \leq N) \\
 & \quad td_j \geq E_{Mj} - d_j \\
 & \quad d_j = \sum_{i=1}^N d_i z_{ij} \\
 & \quad E_{rj}, td_j \geq 0; \quad 1 \leq r \leq M, 1 \leq j \leq N \\
 & \quad z_{ij}, \text{Bin}_{rjb} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq j \leq N, 1 \leq b \leq \text{Bines}
 \end{aligned}$$

3.3.2 Familia de modelos Manne

SGST y LYeq son los modelos pertenecientes a la familia Manne. El modelo SGST utiliza pares de restricciones dicotómicas para resolver el problema de asignación de los trabajos a las máquinas mientras que el modelo LYeq combina estas dos restricciones utilizando una nueva variable, Q_{rik} . Como consecuencia de esto, nos referimos a los trabajos por el índice i , y no por su posición como en los modelos de la familia Wagner.

Modelo SGST

Este modelo emplea las variables C_{ri} , C_{max} , D_{ik} y Bin_{rib} .

$$\text{Min } C_{max} \quad (3.36)$$

$$\text{sa } : C_{1i} \geq T_{1i} \quad (1 \leq i \leq N) \quad (3.37)$$

$$C_{r+1,i} - C_{ri} \geq T_{r+1,i} \quad (1 \leq r \leq M-1, 1 \leq i \leq N) \quad (3.38)$$

$$C_{ri} - C_{rk} + PD_{ik} \geq T_{ri} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \quad (3.39)$$

$$C_{rk} - C_{ri} + P(1 - D_{ik}) \geq T_{rk} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \quad (3.40)$$

$$C_{max} \geq C_{Mi} \quad (1 \leq i \leq N) \quad (3.41)$$

$$C_{ri} - bT \leq P(1 - Bin_{rib}) \quad (1 \leq r \leq M, 1 \leq i \leq N, i \leq b \leq Bines) \quad (3.42)$$

$$C_{ri} - T_{ri} + P(1 - Bin_{rib}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq i \leq N, 1 \leq b \leq Bines) \quad (3.43)$$

$$\sum_{b=1}^{Bines} Bin_{rib} = 1, \quad (1 \leq r \leq M, 1 \leq i \leq N) \quad (3.44)$$

$$C_{ri}, C_{max} \geq 0; \quad 1 \leq r \leq M, 1 \leq i \leq N$$

$$D_{ik}, Bin_{rib} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq k \leq N, 1 \leq b \leq Bines$$

Para este modelo, la función objetivo coincide con la variable C_{max} (3.36) cuyo valor debe coincidir con el tiempo de finalización del último de los trabajos en la última máquina (3.41).

Cada trabajo debe finalizar en la máquina 1 tras ser procesado (3.37). El conjunto de restricciones (3.38) garantiza para cada trabajo que no pueda finalizar en la máquina $r+1$ tras ser procesado en ella y en la anterior, r . Los pares de restricciones dicotómicas (3.39 y 3.40) relacionan dos trabajos i y k , asegurando que cada trabajo i precede o sigue a k en la secuencia, pero no ambas opciones.

Las restricciones (3.42 y 3.43) proporcionan la cota superior del tiempo de finalización y la cota inferior del tiempo de comienzo de cada trabajo, respectivamente, para que sea procesado dentro del bin que le corresponda. Finalmente, cada trabajo (referido por su posición j) podrá ser asignado a un único bin en cada máquina r (3.44).

Modelo SGST adaptado

El modelo modificado incorpora la variable td_i y cambia la restricción (3.41), que indica la cota inferior del makespan, por la restricción (3.45), que indica la del *tardiness*:

$$td_i \geq C_{Mi} - d_i \quad (1 \leq i \leq N) \quad (3.45)$$

El modelo adaptado quedaría de la siguiente forma:

$$\begin{aligned}
& \text{Min } \sum_{i=1}^N td_i \\
& \text{sa : } C_{1i} \geq T_{1i} \quad (1 \leq i \leq N) \\
& C_{r+1,i} - C_{ri} \geq T_{r+1,i} \quad (1 \leq r \leq M-1, 1 \leq i \leq N) \\
& C_{ri} - C_{rk} + PD_{ik} \geq T_{ri} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \\
& C_{rk} - C_{ri} + P(1 - D_{ik}) \geq T_{rk} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \\
& C_{ri} - bT \leq P(1 - Bin_{rib}) \\
& \quad (1 \leq r \leq M, 1 \leq i \leq N, i \leq b \leq Bines) \\
& C_{ri} - T_{ri} + P(1 - Bin_{rib}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq i \leq N, 1 \leq b \leq Bines) \\
& \sum_{b=1}^{Bines} Bin_{rib} = 1, \quad (1 \leq r \leq M, 1 \leq i \leq N) \\
& td_i \geq C_{Mi} - d_i \quad (1 \leq i \leq N) \\
& C_{ri}, td_i \geq 0; \quad 1 \leq r \leq M, 1 \leq i \leq N \\
& D_{ik}, Bin_{rib} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq k \leq N, 1 \leq b \leq Bines
\end{aligned}$$

Modelo LYeq

Este modelo emplea, además de las variables C_{ri} , C_{max} , D_{ik} y Bin_{rib} del modelo SGST, la variable auxiliar Q_{rik} .

$$\text{Min } C_{max} \quad (3.46)$$

$$\text{sa : } C_{1i} \geq T_{1i} \quad (1 \leq i \leq N) \quad (3.47)$$

$$C_{r+1,i} - C_{ri} \geq T_{r+1,i} \quad (1 \leq r \leq M-1, 1 \leq i \leq N) \quad (3.48)$$

$$C_{max} \geq C_{Mi} \quad (1 \leq i \leq N) \quad (3.49)$$

$$PD_{ik} + C_{ri} - C_{rk} - T_{ri} = Q_{rik} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \quad (3.50)$$

$$Q_{rik} \leq P - T_{ri} - T_{rk} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \quad (3.51)$$

$$C_{ri} - bT \leq P(1 - Bin_{rib}) \quad (1 \leq r \leq M, 1 \leq i \leq N, 1 \leq b \leq Bines) \quad (3.52)$$

$$C_{ri} - T_{ri} + P(1 - Bin_{rib}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq i \leq N, 1 \leq b \leq Bines) \quad (3.53)$$

$$\sum_{b=1}^{Bines} Bin_{rib} = 1, \quad (1 \leq r \leq M, 1 \leq i \leq N) \quad (3.54)$$

$$C_{ri}, C_{max}, Q_{rik} \geq 0; \quad 1 \leq r \leq M, 1 \leq i \leq K \leq N$$

$$D_{ik}, Bin_{rib} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq k \leq N, 1 \leq b \leq Bines$$

La función objetivo (3.46) y las restricciones (3.47, 3.48 y 3.49) se corresponden con las ecuaciones (3.36, 3.37 y 3.41) del modelo SGST, respectivamente. Con respecto a los conjuntos de restricciones (3.50 y 3.51) establecen el par de restricciones dicotómicas. El resto de restricciones actúa del mismo modo que en el resto de modelos, garantizando la restricción de disponibilidad de las máquinas.

Modelo LYeq adaptado

El modelo modificado incorpora la variable td_i y cambia la restricción (3.49), que indica la cota inferior del makespan, por la restricción (3.55), que proporciona la del *tardiness*:

$$td_i \geq C_{Mi} - d_i \quad (1 \leq i \leq N) \quad (3.55)$$

El modelo adaptado quedaría de la siguiente forma:

$$\begin{aligned}
& \text{Min } \sum_{i=1}^N td_i \\
& \text{sa : } C_{1i} \geq T_{1i} \quad (1 \leq i \leq N) \\
& C_{r+1,i} - C_{ri} \geq T_{r+1,i} \quad (1 \leq r \leq M-1, 1 \leq i \leq N) \\
& PD_{ik} + C_{ri} - C_{rk} - T_{ri} = Q_{rik} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \\
& Q_{rik} \leq P - T_{ri} - T_{rk} \quad (1 \leq r \leq M, 1 \leq i < k \leq N) \\
& C_{ri} - bT \leq P(1 - Bin_{rib}) \quad (1 \leq r \leq M, 1 \leq i \leq N, 1 \leq b \leq Bines) \\
& C_{ri} - T_{ri} + P(1 - Bin_{rib}) \geq T(b-1) \quad (1 \leq r \leq M, 1 \leq i \leq N, 1 \leq b \leq Bines) \\
& \sum_{b=1}^{Bines} Bin_{rib} = 1, \quad (1 \leq r \leq M, 1 \leq i \leq N) \\
& td_i \geq C_{Mi} - d_i \quad (1 \leq i \leq N) \\
& C_{ri}, Q_{rik}, td_i \geq 0; \quad 1 \leq r \leq M, 1 \leq i \leq K \leq N \\
& D_{ik}, Bin_{rib} \in \{0,1\}; \quad 1 \leq r \leq M, 1 \leq i \leq N, 1 \leq k \leq N, 1 \leq b \leq Bines
\end{aligned}$$

3.3.3 Modelos formulados por Baker (2013)

Las versiones *Idle-time* y *Completion-time* resuelven el problema de asignación de igual forma que los modelos de la familia Wagner, empleando una variable binaria X_{jk} cuyo valor es 1 en el caso de que el trabajo j sea asignado a la posición k de la secuencia. Como consecuencia, nos referimos a cada trabajo por su posición k .

Idle-time version

El modelo emplea las variables C_k, I_{ik}, H_{ki}, t_k y x_{jk} .

$$\text{Min } \sum_{k=1}^N t_k \quad (3.56)$$

$$\text{sa : } \sum_{j=1}^N x_{jk} = 1 \quad (1 \leq k \leq N) \quad (3.57)$$

$$\sum_{k=1}^N x_{jk} = 1 \quad (1 \leq j \leq N) \quad (3.58)$$

$$I_{i,k+1} + \sum_{u=1}^n p_{iu} x_{u,k+1} + H_{i,k+1} - H_{ik} - \sum_{u=1}^N p_{i+1,u} x_{u,k} - I_{i+1,k+1} = 0 \quad (1 \leq k \leq N-1, 1 \leq i \leq M-1) \quad (3.59)$$

$$I_{i1} + \sum_{u=1}^N p_{iu} x_{u1} + H_{i1} - I_{i+1,1} = 0 \quad (1 \leq i \leq M-1) \quad (3.60)$$

$$C_0 = 0 \quad (3.61)$$

$$t_k \geq C_{k-1} + I_{Mk} + \sum_{j=1}^N p_{Mj} x_{jk} - \sum_{j=1}^N d_j x_{jk} \quad (1 \leq k \leq N) \quad (3.62)$$

$$\begin{aligned}
& C_{ik}, I_{ik}, H_{ki}, t_k \geq 0; \quad 1 \leq i \leq M, 1 \leq k \leq N \\
& x_{jk} \in \{0,1\}; \quad 1 \leq j \leq N, 1 \leq k \leq N
\end{aligned}$$

El objetivo de este modelo es minimizar el *tardiness* total (3.56).

Los conjuntos de restricciones (3.57 y 3.58) garantizan la asignación de cada trabajo a una única posición en la secuencia, así como que cada posición sea ocupada por un solo trabajo.

La relación entre las variables que indican el tiempo ocioso se expresa en la ecuación (3.59) para todas las posiciones en la secuencia, y en la (3.60) para la primera posición. Consideramos un trabajo cuyo índice es 0 y finaliza en el instante 0.

Finalmente, el retraso del trabajo en la posición k es limitado inferiormente en el conjunto de restricciones (3.62).

Idle-time version adaptado

El modelo modificado añade a la variable C_k el subíndice i , obteniendo así el tiempo de finalización para cada posición k en cada máquina i , e incorpora la variable Bin_{ikb} , dando lugar a las restricciones que limitan la disponibilidad de las máquinas:

$$C_{ik} = \sum_{l=1}^k (I_{il} + \sum_{j=1}^N p_{ij}x_{jl}) \quad (1 \leq i \leq M, 1 \leq k \leq N) \quad (3.63)$$

$$C_{ik} \leq bT + P(1 - Bin_{ikb}) \quad (1 \leq i \leq M, 1 \leq k \leq N, 1 \leq b \leq bines) \quad (3.64)$$

$$C_{ik} - \sum_{j=1}^N p_{ij}x_{jk} + P(1 - Bin_{ikb}) \geq T(b-1) \quad (1 \leq i \leq M, 1 \leq k \leq N, 1 \leq b \leq Bines) \quad (3.65)$$

$$\sum_b^{Bines} Bin_{ikb} = 1 \quad (1 \leq i \leq M, 1 \leq k \leq N) \quad (3.66)$$

El valor del *tardiness*, td_k es proporcionado por las siguientes restricciones (3.67 y 3.68):

$$td_k \geq C_{Mk} - d_k \quad (1 \leq k \leq N) \quad (3.67)$$

$$d_k = \sum_{j=1}^N d_j x_{jk} \quad (1 \leq k \leq N) \quad (3.68)$$

Con estas modificaciones, el modelo adaptado viene definido por:

$$\begin{aligned} & \text{Min} \sum_{k=1}^N td_k \\ & \text{sa : } \sum_{j=1}^N x_{jk} = 1 \quad (1 \leq k \leq N) \\ & \quad \sum_{k=1}^N x_{jk} = 1 \quad (1 \leq j \leq N) \\ & \quad I_{i,k+1} + \sum_{u=1}^N p_{iu}x_{u,k+1} + H_{i,k+1} - H_{ik} - \sum_{u=1}^N p_{i+1,u}x_{u,k} - I_{i+1,k+1} = 0 \\ & \quad \quad \quad (1 \leq k \leq N-1, 1 \leq i \leq M-1) \\ & \quad I_{i1} + \sum_{u=1}^N p_{iu}x_{u1} + H_{i1} - I_{i+1,1} = 0 \quad (1 \leq i \leq M-1) \\ & \quad C_{M,0} = 0 \\ & \quad td_k \geq C_{Mk} - d_k \quad (1 \leq k \leq n) \\ & \quad d_k = \sum_{j=1}^N d_j x_{jk} \quad (1 \leq k \leq n) \\ & \quad C_{ik} = \sum_{l=1}^k (I_{il} + \sum_{j=1}^N p_{ij}x_{jl}) \quad (1 \leq i \leq M, 1 \leq k \leq N) \\ & \quad C_{ik} \leq bT + P(1 - Bin_{ikb}) \quad (1 \leq i \leq M, 1 \leq k \leq N, 1 \leq b \leq bines) \\ & \quad C_{ik} - \sum_{j=1}^N p_{ij}x_{jk} + P(1 - Bin_{ikb}) \geq T(b-1) \quad (1 \leq i \leq M, 1 \leq k \leq N, 1 \leq b \leq Bines) \\ & \quad \sum_b^{Bines} Bin_{ikb} = 1 \quad (1 \leq i \leq M, 1 \leq k \leq N) \end{aligned}$$

$$C_{ik}, I_{ik}, H_{ki}, td_k, d_k \geq 0; \quad 1 \leq i \leq M, 1 \leq k \leq N$$

$$x_{jk}, Bin_{ikb} \in \{0,1\}; \quad 1 \leq j \leq N, 1 \leq k \leq N$$

Completion-time version

El modelo emplea las variables C_{ik} , t_k y x_{jk} .

$$Min \sum_{k=1}^n t_k \quad (3.69)$$

$$sa : \sum_{j=1}^n x_{ij} = 1 \quad (1 \leq i \leq N) \quad (3.70)$$

$$\sum_{i=1}^N x_{ij} = 1 \quad (1 \leq j \leq N) \quad (3.71)$$

$$C_{ik} \geq C_{i-1,k} + \sum_{j=1}^n p_{ij} x_{jk} \quad (1 \leq i \leq m, 1 \leq k \leq n) \quad (3.72)$$

$$C_{ik} \geq C_{i,k-1} + \sum_{j=1}^n p_{ij} x_{jk} \quad (1 \leq i \leq m, 1 \leq k \leq n) \quad (3.73)$$

$$t_k \geq C_{mk} - \sum_{j=1}^n d_j x_{jk} \quad (1 \leq k \leq n) \quad (3.74)$$

$$C_{ik}, t_k \geq 0; \quad 1 \leq i \leq M, 1 \leq k \leq N$$

$$x_{jk} \in \{0,1\}; \quad 1 \leq j \leq N, 1 \leq k \leq N$$

En este modelo, la función objetivo (3.69) y las restricciones (3.70 y 3.71) se explican de la misma forma que las de la versión *Idle-time* (3.56, 3.57 y 3.58). El conjunto de restricciones (3.72) asegura para cada trabajo cuya posición es k , que no comience a ser procesado en la máquina i hasta que acabe en la máquina $i - 1$. Adicionalmente, no podrá comenzar hasta que no acabe el trabajo en la posición anterior (3.73).

Para finalizar, el *tardiness* se define de igual forma que en modelos anteriores (3.74).

Completion-time version adaptado

El modelo modificado incorpora la variable Bin_{ikb} , para restringir la disponibilidad de las máquinas mediante las restricciones (3.75, 3.76 y 3.77) que se muestran a continuación:

$$C_{ik} \leq P(1 - Bin_{ikb}) + bT \quad (1 \leq i \leq m, 1 \leq k \leq n, 1 \leq b \leq Bines) \quad (3.75)$$

$$C_{ik} - \sum_{j=1}^n p_{ij} x_{jk} + P(1 - Bin_{ikb}) \geq T(b - 1) \quad (1 \leq i \leq n, 1 \leq k \leq n, 1 \leq b \leq Bines) \quad (3.76)$$

$$\sum_{b=1}^{Bines} Bin_{ikb} = 1 \quad (1 \leq i \leq n, 1 \leq k \leq n) \quad (3.77)$$

Teniendo en cuenta las modificaciones consideradas, el modelo adaptado es el siguiente:

$$\begin{aligned}
 & \text{Min } \sum_{k=1}^n td_k \\
 & \text{sa : } \sum_{j=1}^n x_{jk} = 1 \quad (1 \leq k \leq N) \\
 & \quad \sum_{k=1}^N x_{jk} = 1 \quad (1 \leq j \leq N) \\
 & \quad C_{ik} \geq C_{i-1,k} + \sum_{j=1}^n p_{ij}x_{jk} \quad (1 \leq i \leq m, 1 \leq k \leq n) \\
 & \quad C_{ik} \geq C_{i,k-1} + \sum_{j=1}^n p_{ij}x_{jk} \quad (1 \leq i \leq m, 1 \leq k \leq n) \\
 & \quad td_k \geq C_{mk} - d_k \quad (1 \leq k \leq n) \\
 & \quad d_k = \sum_{j=1}^n d_j x_{jk} \quad (1 \leq k \leq n) \\
 & \quad C_{ik} \leq P(1 - \text{Bin}_{ikb}) + bT \quad (1 \leq i \leq m, 1 \leq k \leq n, 1 \leq b \leq \text{Bines}) \\
 & \quad C_{ik} - \sum_{j=1}^n p_{ij}x_{jk} + P(1 - \text{Bin}_{ikb}) \geq T(b-1) \quad (1 \leq i \leq n, 1 \leq k \leq n, 1 \leq b \leq \text{Bines}) \\
 & \quad \sum_{b=1}^{\text{Bines}} \text{Bin}_{ikb} = 1 \quad (1 \leq i \leq n, 1 \leq k \leq n)
 \end{aligned}$$

$$\begin{aligned}
 & C_{ik}, td_k, d_k \geq 0; \quad 1 \leq i \leq M, 1 \leq k \leq N \\
 & x_{jk}, \text{Bin}_{ikb} \in \{0,1\}; \quad 1 \leq j \leq n, 1 \leq k \leq n, 1 \leq b \leq \text{Bines}
 \end{aligned}$$

4 Análisis

4.1 Resolución del modelo

El hito último de este proyecto es el análisis de los siete modelos presentados en el apartado 3 según el tiempo empleado para hallar la solución óptima, o en su defecto, la mejor solución encontrada hasta agotar el tiempo disponible.

Para presentar este análisis, es necesario exponer el método empleado para la resolución de los modelos. Cada modelo resuelve una batería de 1.920 instancias, que difieren en el número de máquinas (M), trabajos (N) y el periodo de disponibilidad (T). El número de máquinas varía entre 2, 3, 4 y 5, el de trabajos entre 5, 10, 15 y 20 y el periodo de disponibilidad es de 100, 200 o 300 u.t. Se han realizado cruces de estos tres datos. Para cada una de las combinaciones resultantes se resuelven a su vez 10 instancias diferentes. Para cada una de estas instancias, los tiempos de proceso vienen dados según la distribución uniforme discreta entre 1 y 99 u.t.

$$T_{ri} \sim U[1,99]$$

Asimismo, las fechas de entrega se han generado con la distribución uniforme:

$$U[p(1 - T - R/2), p(1 - T + R/2)],$$

donde p se define como la suma de los tiempos de proceso para todos los trabajos para todas las máquinas y los parámetros T y R son 0.2 y 0.6, respectivamente.

La existencia de una batería de 1.920 instancias por modelo implica la resolución de 13.440 problemas diferentes. Se han seguido las siguientes fases para su ejecución:

- Utilización del optimizador *Gurobi-7.0.2-win64* en una plataforma *PC* con procesador *Intel(R) Core(TM) i5 de 1.80 GHz* y sistema operativo de 64 bits *Windows 10 home* de *Microsoft Corporation*. Los tiempos de cómputo requeridos para alcanzar la solución óptima pueden ser excesivamente altos debido a la complejidad de los problemas en algunos casos. Para evitar este hecho, el tiempo de ejecución para cada instancia ha sido limitado a 900s (15 minutos). Si el *software* no ha alcanzado la solución óptima en este tiempo, devolverá la mejor solución factible encontrada hasta entonces. Los datos obtenidos de la resolución tienen el formato “.log”.
- Exportación de los resultados a *Excel*.
- Tratamiento de datos mediante *RStudio* y lenguaje de programación *R*.

4.2 Análisis estadístico de los resultados

Como se ha indicado en el apartado anterior, hemos obtenido un total de 1.920 observaciones de la resolución de los problemas para los siete modelos. En este apartado se realiza un estudio estadístico para concluir qué modelo resuelve con mayor rapidez esta problemática.

Tabla 4.1 Modelos objeto de estudio.

Modelos propuestos por Stafford et al. (2005)				Modelos propuestos por Baker (2013)		
WST	Wilson	TS2	SGST	LYeq	Idle-time	Completion-time

Tabla 4.2 *CPU Time* (s) medio, desviación estándar y porcentaje de instancias para las que se garantiza optimalidad por modelo..

	CPU Time (s)		
	Media	Desv. Est.	Porcentaje optimalidad (%)
WST	39.95	156.55	97.50
Wilson	7.13	68.75	99.43
TS2	9.00	97.00	99.43
SGST	1.48	18.81	100
LYeq	6.55	33.44	99.95
Idle-time	41.73	257.89	97.60
Completion-time	7.32	69.34	99.53

La variable objeto de estudio es el tiempo de cómputo (*CPU Time*). Esta variable depende del modelo, del número de máquinas (M) y del periodo de disponibilidad (T). Por tanto, estudiaremos como influyen cada uno de estos factores en el *CPU Time*, el cual se mide en segundos (s).

El tratamiento de los datos se ha llevado a cabo mediante el entorno *RStudio*, con lenguaje de programación *R*.

La tabla 4.1 muestra los modelos objeto de estudio. Estos modelos se describen con detalle en el apartado 3.3.

4.2.1 Influencia del Modelo en el CPU Time

En este apartado se analiza la influencia de los modelos en el tiempo de cómputo. Es necesario tener en cuenta que el tiempo de cómputo ha sido limitado a 900s. Si el *software* no ha alcanzado la solución óptima en este periodo, devolverá la mejor solución encontrada hasta entonces, o en su defecto, no proporcionará solución factible.

En primer lugar, obtenemos las medias, las desviaciones estándar y el porcentaje de instancias para las que se garantiza optimalidad por modelo (tabla 4.2). Este porcentaje se calcula a partir del número de instancias que no consumen el tiempo límite de computación, es decir, aquellas para las que podemos asegurar que alcanzan el óptimo. A partir de estos datos, representamos las medias de cada modelo y la media total de todos los modelos en la figura 4.1.

En segundo lugar, representamos los datos mediante un Diagrama de cajas y bigotes (figura 4.2). Este gráfico consiste en una caja rectangular, donde los lados más largos muestran el recorrido intercuartílico. Este rectángulo está dividido por un segmento horizontal que indica donde se posiciona la mediana y por lo tanto su relación con los cuartiles primero y tercero, Q_1 y Q_3 , respectivamente. Las líneas que sobresalen de la caja se llaman bigotes. Estos bigotes tienen un límites inferiores y superiores de prolongación, que se calculan como muestran las ecuaciones (4.1 y 4.2):

$$LI = Q_1 - 1.5(Q_3 - Q_1) \quad (4.1)$$

$$LS = Q_1 + 1.5(Q_3 - Q_1) \quad (4.2)$$

Los datos que se encuentran fuera de los límites suelen considerarse *outliers*, aunque en este caso se deben a los distintos cruces de máquinas, trabajos y periodo de disponibilidad. Estos cruces desembocan en instancias dispares, cuyo tiempo de computación puede ser muy diferente. Como vemos en la figura 4.2, existen numerosos valores fuera de los límites, por lo que filtramos para eliminar parte de ellos y ver los datos con claridad. Esto no indica que no sean relevantes, sino que de este modo podemos ver las medianas y el

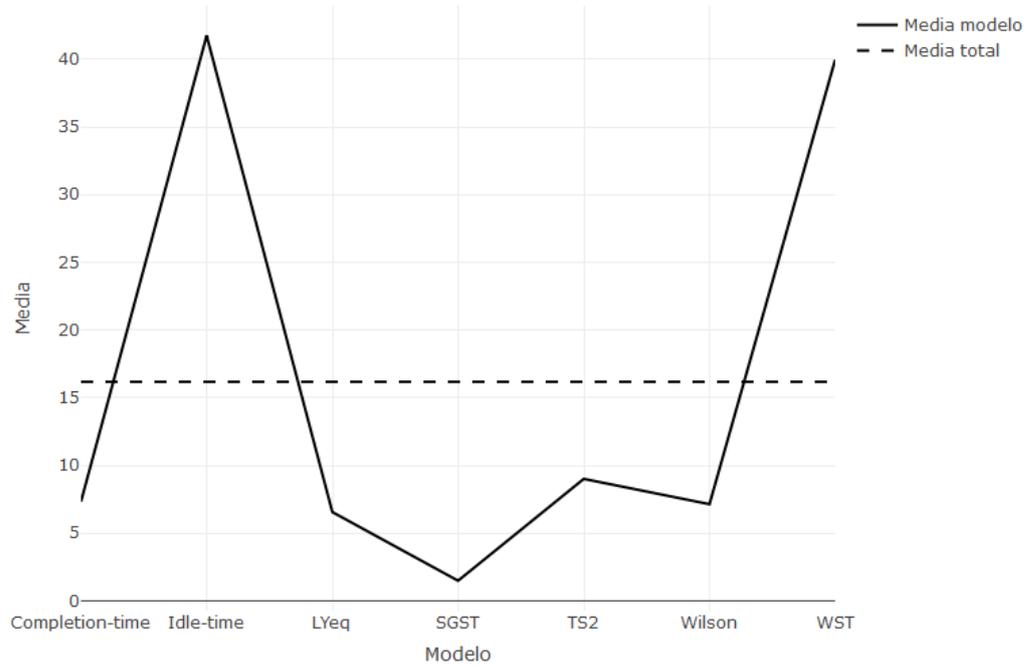


Figura 4.1 Medias de cada modelo y media total.

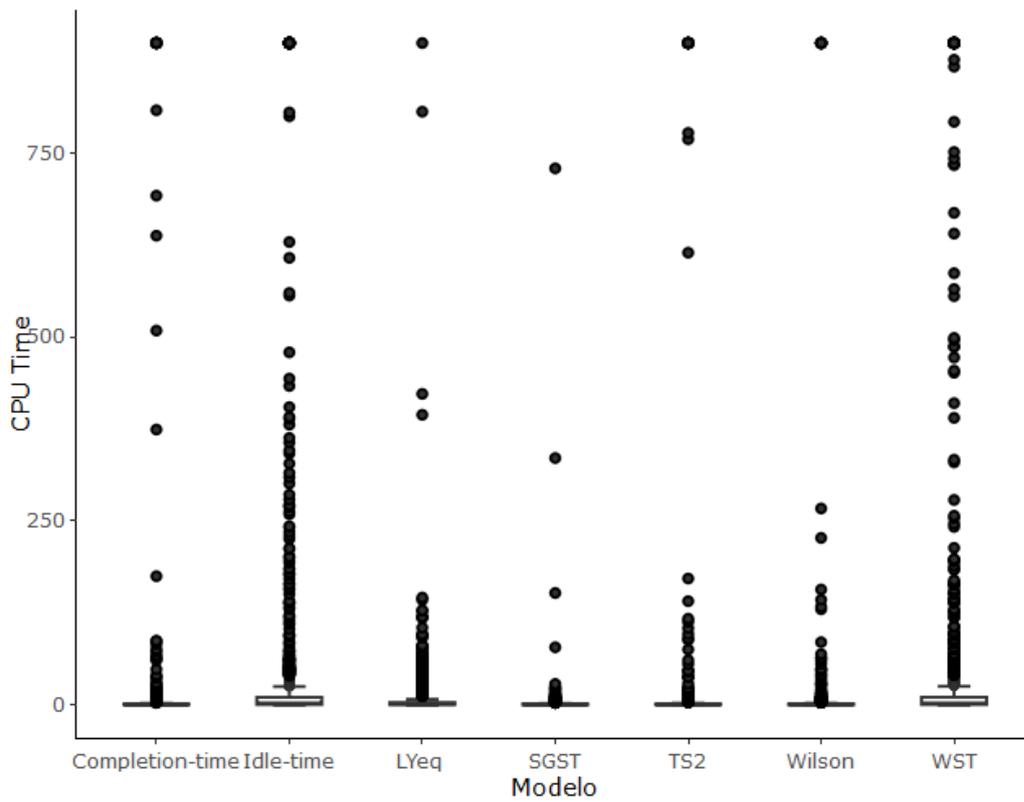


Figura 4.2 Diagrama de cajas y bigotes. CPU Time por modelo.

recorrido intercuartílico de mejor forma. El 54.36% de las observaciones eliminadas pertenecen a instancias de 20 máquinas y el 34.74% a instancias de 15 máquinas. El porcentaje restante será de 10 máquinas. Por tanto, deducimos que el tiempo de cómputo aumenta con el número de máquinas. La gráfica resultante se muestra en la figura 4.3.

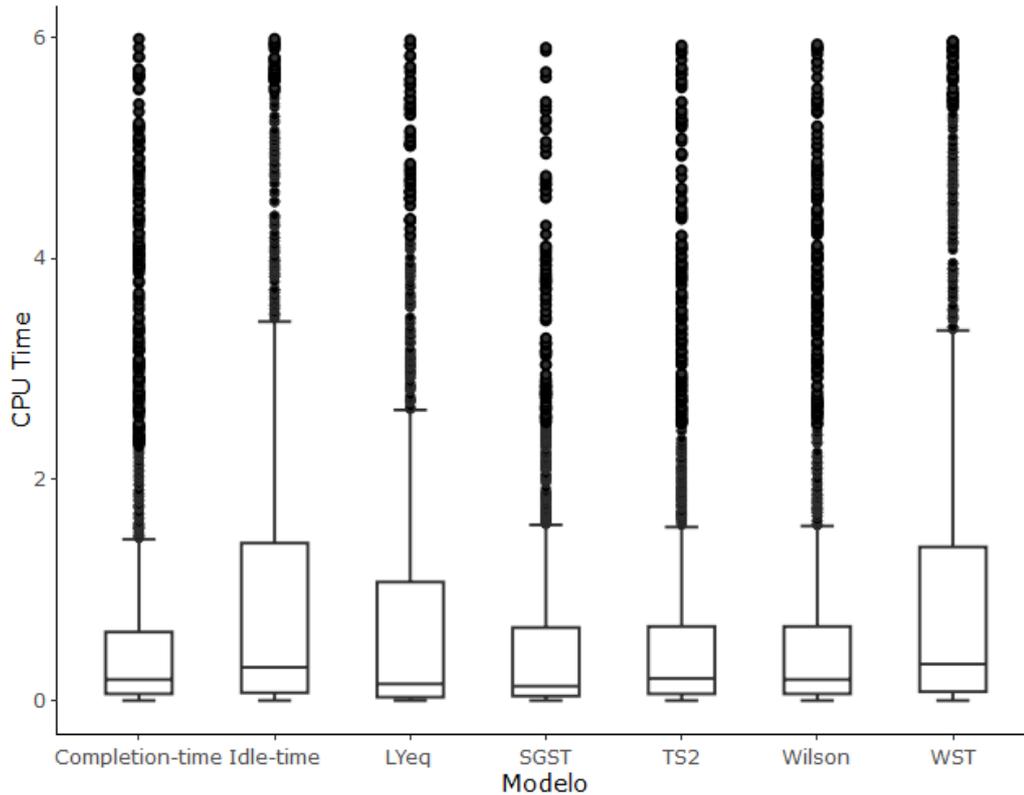


Figura 4.3 Diagrama de cajas y bigotes reducido. *CPU Time* por modelo.

Con estos datos y representaciones podemos llegar a las siguientes conclusiones:

- La media del tiempo de cómputo es notablemente superior para los modelos WST e Idle-time (39.95s y 41.73s, respectivamente) y tiene un valor muy bajo para el modelo SGST (1.48s).
- La dispersión de los datos es llamativa principalmente para los modelos Idle-time y WST, a pesar de que su valor es alto para la mayoría de modelos. Por tanto, podemos afirmar para los modelos Idle-Time y WST que el tiempo de cómputo para algunas instancias ha sido destacadamente alto. Esto lo observamos tanto en el Diagrama de Cajas y Bigotes como en la desviación estándar.
- Finalmente podemos concluir que el modelo SGST es el más eficiente en cuanto a *CPU Time* empleado. Asimismo, es el único que garantiza la optimalidad de todas las soluciones. Los modelos Idle-time y WST los que obtienen los peores resultados.

4.2.2 Influencia del periodo de disponibilidad (T) en el *CPU time*

En este apartado se analiza la influencia del periodo de disponibilidad en el tiempo de cómputo.

Para realizar este análisis, basta con representar la función de densidad del tiempo empleado para los tres valores que puede tomar el periodo de disponibilidad, 100, 200 y 300 (figura 4.4). La función de densidad representa la frecuencia (número de observaciones) con que aparecen las mediciones.

De la figura 4.4 podemos llegar a la siguiente conclusión:

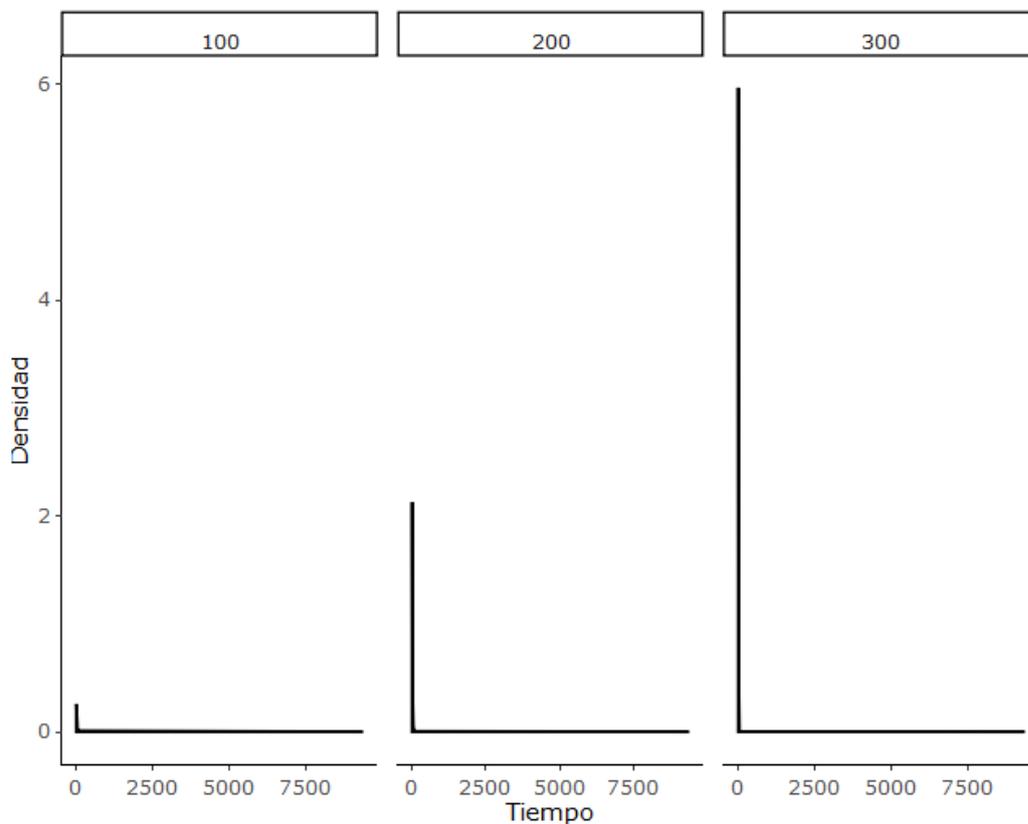


Figura 4.4 Función de densidad para cada periodo de disponibilidad (T).

- A medida que aumenta el periodo de disponibilidad, la frecuencia de las observaciones para tiempos pequeños aumenta, es decir, la mayoría de las instancias requieren un tiempo de cómputo menor. Este resultado se debe a que mayor valor de T requiere menor número de Bines, por tanto la resolución del modelo se simplifica.

4.2.3 Influencia del modelo y el periodo de disponibilidad (T) en el CPU Time

En este apartado se analiza la influencia del periodo de disponibilidad y el modelo en el tiempo de cómputo.

Representamos los datos en el Diagrama de cajas y bigotes (figura 4.5) y vemos que presentan gran dispersión. Por tanto, al igual que en los casos anteriores, filtramos las observaciones de tiempos altos para ver los datos con claridad (figura 4.6).

Por otra parte, representamos las medias y sus desviaciones estándar en la tabla 4.3.

Con estos datos, llegamos a las siguientes conclusiones:

- Con periodos de disponibilidad de 200 y 300 los modelos se comportan de forma similar en cuanto a media de tiempo empleado. Para T=100 observamos que la diferencia de medias entre modelos es muy alta. Para los modelos Idle-time y WST su valor es notablemente superior al resto.
- Con relación a la dispersión de los datos, disminuye de forma inversa a T.

4.2.4 Influencia del modelo y el número de trabajos (N) en el CPU Time

En este apartado se estudia la influencia del modelo y el número de trabajos en el CPU Time.

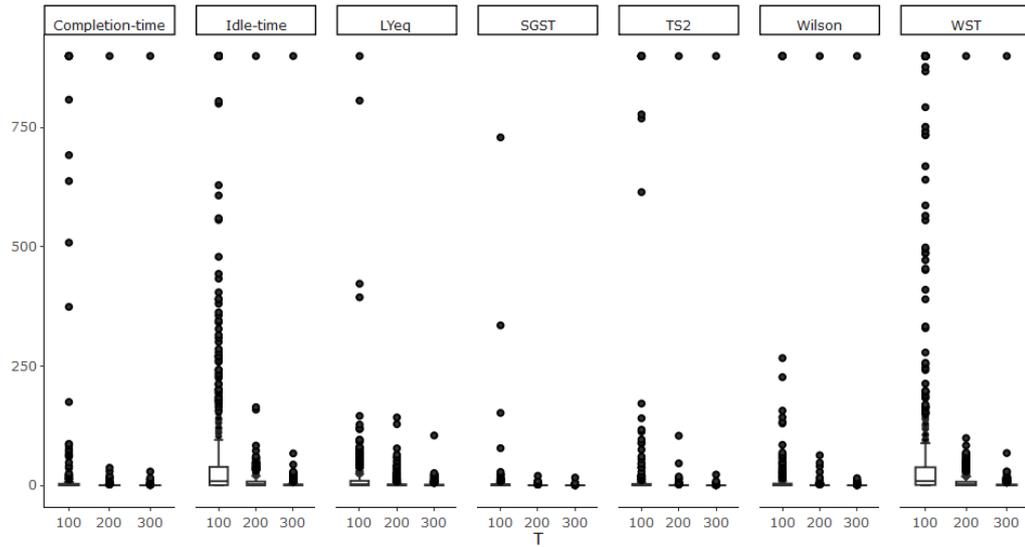


Figura 4.5 Diagrama de cajas y bigotes. *CPU Time* por modelo y periodo de disponibilidad (T).

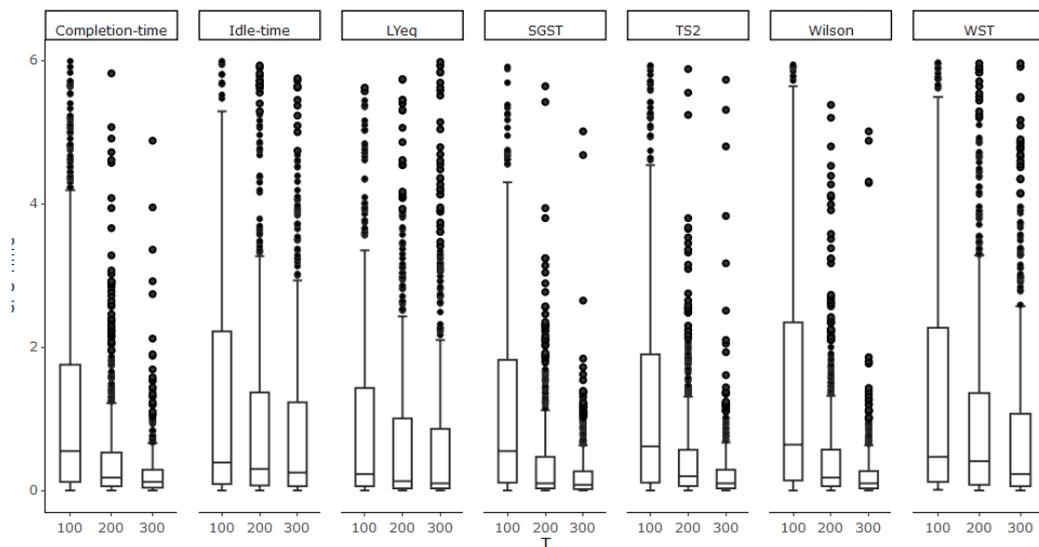


Figura 4.6 Diagrama de cajas y bigotes. *CPU Time* por modelo y periodo de disponibilidad (T) para valores del *CPU Time* menores que 6..

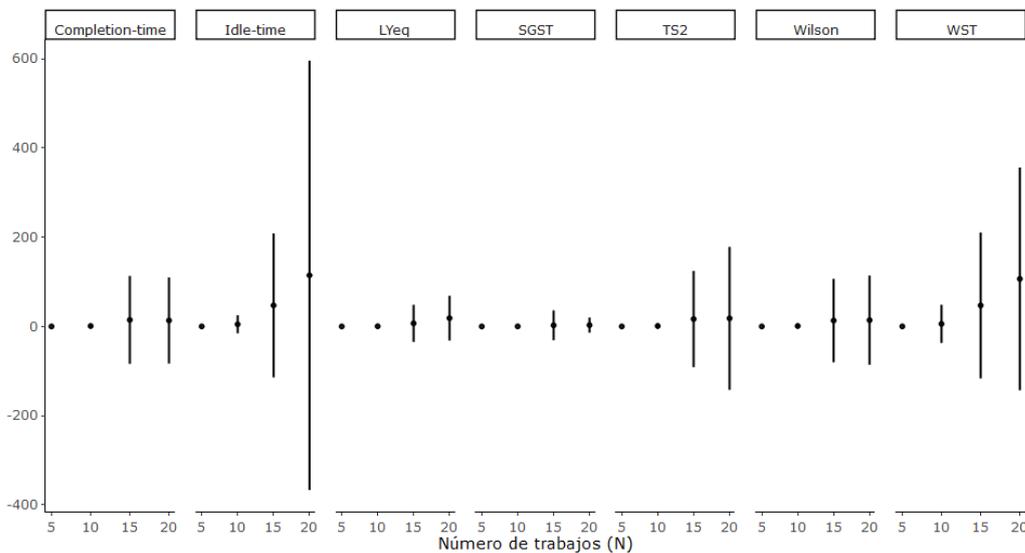
Para llevar a cabo este análisis, calculamos las medias y sus desviaciones estándar en la tabla 4.4 y las representamos para cada uno de los modelos y número de trabajos a asignar (figura 4.7).

De este estudio podemos extraer las siguientes conclusiones:

- Para $N = 5$, todos los modelos actúan de forma similar y no presentan desviaciones destacables con respecto a la media, es decir, no hay ningún valor atípico. Para $N = 10$ los valores medios del *CPU time* son levemente superiores, así como las desviaciones. No obstante, podemos concluir que no presentan notables diferencias.
- A medida que aumenta N, podemos observar con mayor claridad las diferencias entre modelos, concluyendo, como en apartados anteriores, que el modelo Idle-time presenta los peores resultados para estos valores, seguido del WST.

Tabla 4.3 CPU Time(s) medio y desviación estándar en función del modelo y el periodo de disponibilidad (T).

		CPU Time (s)	
		Media	Desv. Est.
WST	T=100	107.54	253.00
	T=200	8.88	37.80
	T=300	3.43	35.80
Wilson	T=100	17.42	107.20
	T=200	2.26	35.75
	T=300	1.70	35.58
TS2	T=100	23.05	159.39
	T=200	2.23	35.85
	T=300	1.71	35.58
SGST	T=100	3.70	32.45
	T=200	0.47	1.16
	T=300	0.27	0.77
LYeq	T=100	14.12	55.83
	T=200	3.61	11.03
	T=300	1.91	5.62
Idle-Time	T=100	112.63	435.20
	T=200	8.96	38.44
	T=300	3.61	35.86
Completion-time	T=100	18.06	108.30
	T=200	2.12	35.63
	T=300	1.76	35.60

**Figura 4.7** Media y desviación estándar de cada modelo para cada número de trabajos.

- Asimismo, cabe destacar que el aumento del número de trabajos no implica en todos los casos el aumento del tiempo medio de computación, ya que en modelos como el Completion-time esto no ocurre.

4.2.5 Influencia del modelo y el número de máquinas (M) en el CPU Time

En este apartado se estudia la influencia del modelo y el número de trabajos en el CPU Time.

Tabla 4.4 CPU Time(s) medio y desviación estándar en función del modelo y el número de trabajos (N)..

		CPU Time (s)	
		Media	Desv. Est.
WST	N=5	23.62	130.68
	N=10	12.42	62.26
	N=15	50.80	180.16
	N=20	72.97	206.24
Wilson	N=5	18.77	122.46
	N=10	5.04	58.16
	N=15	1.79	7.99
	N=20	2.92	17.53
TS2	N=5	22.49	173.77
	N=10	5.30	58.43
	N=15	1.85	10.06
	N=20	6.36	61.06
SGST	N=5	3.22	37.42
	N=10	0.63	2.11
	N=15	0.82	1.91
	N=20	1.24	2.10
LYeq	N=5	8.03	61.66
	N=10	3.47	9.91
	N=15	5.80	14.96
	N=20	8.90	18.38
Idle-Time	N=5	21.13	121.44
	N=10	14.08	70.66
	N=15	60.89	453.93
	N=20	70.83	195.57
Completion-time	N=5	16.86	113.61
	N=10	4.20	47.38
	N=15	1.75	9.29
	N=20	6.45	62.38

Para llevar a cabo este análisis, calculamos las medias y sus desviaciones estándar en la tabla 4.5 y las representamos para cada uno de los modelos y número de trabajos a asignar (figura 4.8).

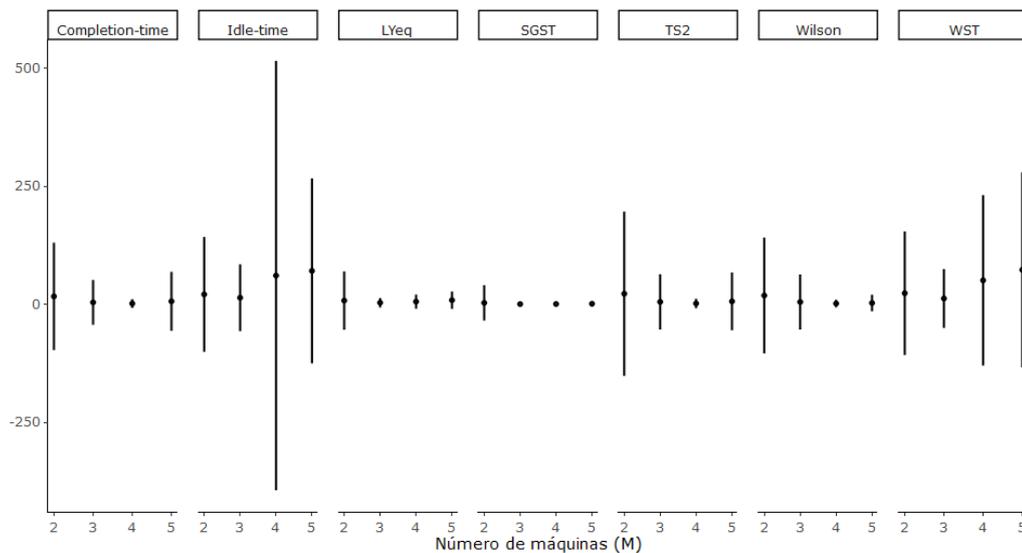


Figura 4.8 Media y desviación estándar de cada modelo para cada número de máquinas.

Tabla 4.5 CPU Time(s) medio y desviación estándar en función del modelo y el número de máquinas (M).

		CPU Time (s)	
		Media	Desv. Est.
WST	M=2	23.62	130.68
	M=3	12.42	62.26
	M=4	50.80	180.16
	M=5	72.97	206.24
Wilson	M=2	18.77	122.46
	M=3	5.04	58.16
	M=4	1.79	7.99
	M=5	2.92	17.53
TS2	M=2	22.49	173.77
	M=3	5.30	58.43
	M=4	1.85	10.06
	M=5	6.36	61.06
SGST	M=2	3.22	37.42
	M=3	0.63	2.11
	M=4	0.82	1.91
	M=5	1.24	2.10
LYeq	M=2	8.03	61.66
	M=3	3.47	9.91
	M=4	5.80	14.96
	M=5	8.90	18.38
Idle-Time	M=2	21.13	121.44
	M=3	14.08	70.66
	M=4	60.89	453.93
	M=5	70.83	195.57
Completion-time	M=2	16.86	113.61
	M=3	4.20	47.38
	M=4	1.75	9.29
	M=5	6.45	62.38

Como vemos en los datos, la influencia de las máquinas no permite sacar ideas concluyentes. Únicamente observamos que los modelos Idle-time y WST se comportan de forma muy similar

5 Conclusiones

En este trabajo se han adaptado siete modelos diferentes existentes en la literatura para resolver el siguiente problema de programación de la producción: la asignación de máquinas, las cuales trabajan en un entorno de tipo *flowshop* de permutación, al procesamiento de trabajos, con el objetivo de minimizar el *tardiness* total y teniendo en cuenta las restricciones de disponibilidad de las máquinas. Este problema queda caracterizado de la siguiente forma: $F_m | pmu, nr - pm | \sum_j T_j$.

Todos los modelos deben obtener la solución óptima según el modelado previo. No obstante, no resultan igual de eficientes a la hora de lograrlo. Por tanto, el objetivo último de este proyecto es analizar el desempeño de los modelos en cuanto a tiempo de cómputo empleado, para tratar determinar qué modelo resulta más eficiente.

Para lograr este objetivo, los modelos se han programado en lenguaje C para que puedan resolver una batería de 13.440 instancias mediante el *software* de optimización *Gurobi*.

A partir de las observaciones alcanzadas, se ha realizado un análisis estadístico evaluando el desempeño de los modelos según los factores que pueden influir en él: el periodo de disponibilidad, el número de máquinas y el número de trabajos. En todos los casos se ha observado que el modelo SGST es el más eficiente a la hora de resolver el problema propuesto.

Apéndice A

Anexo

En esta apartado se adjunta el código en lenguaje C que implementa cada uno de los siete modelos desarrollados para resolver la problemática expuesta.

A.0.1 Modelo WST

```
modelo_WST(MAT_INT p_ri, int n, int m, int Bines, int T, int P, MAT_INT di, char
    **argv)
{
    int j,i,r,l,b;
    FILE *fichero=fopen(argv[5],"w");
    fprintf(fichero,"Minimize\n");
    //F0
    fprintf(fichero,"td[1]");
    for(j=2;j<=n;j++)
    {
        fprintf(fichero, " + td[%d]",j);
    }

    //Restricciones
    fprintf(fichero, "\nSubject To");
    //R1
    for(i=1; i<=n;i++)
    {
        fprintf(fichero,"\nR1.%d:",i);
        fprintf(fichero," Z[%d][1]",i);
        for(j=2;j<=n;j++)
        {
            fprintf(fichero, " + Z[%d][%d]",i,j);
        }
        fprintf(fichero, " = 1");
    }
    //R2
    for(j=1; j<=n;j++)
    {
        fprintf(fichero,"\nR2.%d:",j);
        fprintf(fichero," Z[1][%d]",j);
        for(i=2;i<=n;i++)
        {
            fprintf(fichero, " + Z[%d][%d]",i,j);
        }
    }
}
```

```

    fprintf(fichero, " = 1");
}
//R3
for(r=1;r<=m-1;r++)
{
    for(j=1;j<=n-1;j++)
    {
        fprintf(fichero, "\nR3.%d.%d: %d Z[1] [%d]", r, j, p_ri[r-1][0], j+1);
        for(i=2;i<=n;i++)
        {
            fprintf(fichero, " + %d Z[%d] [%d]", p_ri[r-1][i-1], i, j+1);
        }
        fprintf(fichero, " + X[%d] [%d] + Y[%d] [%d]", r, j+1, r, j+1);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " - %d Z[%d] [%d]", p_ri[r][i-1], i, j);
        }
        fprintf(fichero, " - X[%d] [%d] - Y[%d] [%d] = 0", r+1, j+1, r, j);
    }
}
//R4
for(r=1;r<=m-1;r++)
{
    fprintf(fichero, "\nR4.%d:", r);
    fprintf(fichero, " X[%d] [1] - X[%d] [1] - Y[%d] [1]", r+1, r, r);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " - %d Z[%d] [1]", p_ri[r-1][i-1], i);
    }
    fprintf(fichero, " = 0");
}
//R5
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nR5.%d.%d:", r, j);
        fprintf(fichero, " E[%d] [%d]", r, j);
        for(l=1;l<=j;l++)
        {
            fprintf(fichero, " - X[%d] [%d]", r, l);
            {
                for(i=1;i<=n;i++)
                {
                    fprintf(fichero, " - %d Z[%d] [%d]", p_ri[r-1][i-1], i, l);
                }
            }
        }
        fprintf(fichero, " = 0");
    }
}
//R6
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)

```

```

        {
            fprintf(fichero, "\nR6.%d.%d.%d: E[%d] [%d] + %d Bin[%d] [%d] [%d]
                <= %d",r,j,b,r,j,P,r,j,b,b*T+P);
        }
    }
}
//R7
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero,"\nR7.%d.%d.%d: E[%d] [%d]",r,j,b,r,j);
            for(i=1;i<=n;i++)
            {
                fprintf(fichero, " - %d Z[%d] [%d]", p_ri[r-1][i-1],i,j);
            }
            fprintf(fichero, " - %d Bin[%d] [%d] [%d] >= %d",P,r,j,b, T*(b-1)-
                P);
        }
    }
}
//R8
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nR8.%d.%d: Bin[%d] [%d] [1] ",r,j,r,j);
        for(b=2;b<=Bines;b++)
        {
            fprintf(fichero, " + Bin[%d] [%d] [%d]",r,j,b);
        }
        fprintf(fichero, " = 1");
    }
}
//R9
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nR9.%d: td[%d] - E[%d] [%d] + d[%d] >= 0",j,j,m,j,j);
}
//R10
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nR10.%d: d[%d]",j,j);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " - %d Z[%d] [%d]",di[0][i-1],i,j);
    }
    fprintf(fichero, " = 0");
}
}
//Bounds
fprintf(fichero, "\nBounds");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {

```

```

        fprintf(fichero, "\nX[%d] [%d] >= 0",r,j);
        fprintf(fichero, "\nY[%d] [%d] >= 0",r,j);
        fprintf(fichero, "\nE[%d] [%d] >= 0",r,j);
    }
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\ntd[%d] >= 0",j);
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nd[%d] >= 0",j);
}
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\n0 <= Z[%d] [%d] <= 1",i,j);
    }
}
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",r,j,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "Z[%d] [%d] ",i,j);
    }
}
fprintf(fichero, "\n");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",r,j,b);
        }
    }
}
//General
fprintf(fichero, "\nGeneral\n");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)

```

```

    {
        fprintf(fichero, "X[%d] [%d] ",r,j);
        fprintf(fichero, "Y[%d] [%d] ",r,j);
        fprintf(fichero, "E[%d] [%d] ",r,j);
    }
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "td[%d] ",j);
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "d[%d] ",j);
}
//fprintf(fichero, "X Y E t");
fprintf(fichero, "\nEnd");
fclose(fichero);
return 0;
}

```

A.0.2 Modelo Wilson

```

modelo_Wilson(MAT_INT p_ri, int n, int m, int Bines,int T, int P, MAT_INT di,
char **argv)
{
    int i,j,r,b;
    FILE *fichero=fopen(argv[5],"w");
    fprintf(fichero,"Minimize\n");
    //F0
    fprintf(fichero,"td[1]");
    for(j=2;j<=n;j++)
    {
        fprintf(fichero, " + td[%d]",j);
    }
    //Restricciones
    fprintf(fichero, "\nSubject To");
    //R1
    for(i=1; i<=n;i++)
    {
        fprintf(fichero,"\nR1.%d:",i);
        fprintf(fichero," Z[%d] [1]",i);
        for(j=2;j<=n;j++)
        {
            fprintf(fichero, " + Z[%d] [%d]",i,j);
        }
        fprintf(fichero, " = 1");
    }
    //R2
    for(j=1; j<=n;j++)
    {
        fprintf(fichero,"\nR2.%d:",j);
        fprintf(fichero," Z[1] [%d]",j);
        for(i=2;i<=n;i++)
        {
            fprintf(fichero, " + Z[%d] [%d]",i,j);
        }
    }
}

```

```

    }
    fprintf(fichero, " = 1");

}
//R3
for(j=1;j<=n-1;j++)
{
    fprintf(fichero, "\nR3.%d: B[1] [%d]",j,j);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " + %d Z[%d] [%d]",p_ri[0][i-1],i,j);
    }
    fprintf(fichero, " - B[1] [%d] <= 0", j+1);
}
//R4
fprintf(fichero, "\nR4: B[1] [1] = 0");
//R5
for(r=1;r<=m-1;r++)
{
    fprintf(fichero, "\nR5.%d: B[%d] [1]",r,r);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " + %d Z[%d] [1]",p_ri[r-1][i-1],i);
    }
    fprintf(fichero, " - B[%d] [1] <= 0",r+1);
}
//R6
for(r=1;r<=m-1;r++)
{
    for(j=2;j<=n;j++)
    {
        fprintf(fichero,"\nR6.%d.%d: B[%d] [%d]",r,j,r,j);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " + %d Z[%d] [%d]",p_ri[r-1][i-1],i,j);
        }
        fprintf(fichero, " - B[%d] [%d] <= 0",r+1,j);
    }
}
//R7
for(r=2;r<=m;r++)
{
    for(j=1;j<=n-1;j++)
    {
        fprintf(fichero,"\nR7.%d.%d: B[%d] [%d]",r,j,r,j);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " + %d Z[%d] [%d]",p_ri[r-1][i-1],i,j);
        }
        fprintf(fichero, " - B[%d] [%d] <= 0",r,j+1);
    }
}
//R8
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {

```

```

    for(b=1;b<=Bines;b++)
    {
        fprintf(fichero,"\nR8.%d.%d.%d: B[%d] [%d]",r,j,b,r,j);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " + %d Z[%d] [%d]",p_ri[r-1][i-1],i,j);
        }
        fprintf(fichero, " + %d Bin[%d] [%d] [%d] <= %d",P,r,j,b,P+(b*T));
    }
}
//R9
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero,"\nR9.%d.%d.%d: B[%d] [%d] - %d Bin[%d] [%d] [%d]
                >= %d",r,j,b,r,j,P,r,j,b,T*(b-1)-P);
        }
    }
}
//R10
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nR10.%d.%d: Bin[%d] [%d] [1]",r,j,r,j);
        for(b=2;b<=Bines;b++)
        {
            fprintf(fichero, " + Bin[%d] [%d] [%d]",r,j,b);
        }
        fprintf(fichero, " = 1");
    }
}
//R11
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nR11.%d: td[%d] - B[%d] [%d]",j,j,m,j);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " - %d Z[%d] [%d]",p_ri[m-1][i-1],i,j);
    }
    fprintf(fichero, " + d[%d] >= 0",j);
}
//R12
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nR12.%d: d[%d]",j,j);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " - %d Z[%d] [%d]",di[0][i-1],i,j);
    }
    fprintf(fichero, " = 0");
}
//Bounds

```

```

fprintf(fichero, "\nBounds");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nB[%d] [%d] >= 0",r,j);
    }
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\ntd[%d] >= 0",j);
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nd[%d] >= 0",j);
}
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\n0 <= Z[%d] [%d] <= 1",i,j);
    }
}
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",r,j,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "Z[%d] [%d] ",i,j);
    }
}
fprintf(fichero, "\n");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",r,j,b);
        }
    }
}
//General
fprintf(fichero, "\nGeneral\n");

```

```

for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "B[%d] [%d] ",r,j);
    }
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "td[%d] ",j);
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "d[%d] ",j);
}
fprintf(fichero, "\nEnd");
fclose(fichero);
return 0;
}

```

A.0.3 Modelo TS2

```

modelo_TS2(MAT_INT p_ri, int n, int m, int Bines,int T, int P, MAT_INT di,char
**argv)
{
    int i,j,r,b;
    FILE *fichero=fopen(argv[5],"w");
    fprintf(fichero,"Minimize\n");
    //F0
    fprintf(fichero,"td[1]");
    for(j=2;j<=n;j++)
    {
        fprintf(fichero, " + td[%d]",j);
    }
    //Restricciones
    fprintf(fichero, "\nSubject To");
    //R1
    for(i=1; i<=n;i++)
    {
        fprintf(fichero,"\nR1.%d:",i);
        fprintf(fichero," Z[%d] [1]",i);
        for(j=2;j<=n;j++)
        {
            fprintf(fichero, " + Z[%d] [%d]",i,j);
        }
        fprintf(fichero, " = 1");
    }
    //R2
    for(j=1; j<=n;j++)
    {
        fprintf(fichero,"\nR2.%d:",j);
        fprintf(fichero," Z[1] [%d]",j);
        for(i=2;i<=n;i++)
        {

```

```

        fprintf(fichero, " + Z[%d] [%d]", i, j);
    }
    fprintf(fichero, " = 1");

}
//R3
for(r=1;r<=m;r++)
{
    for(j=1;j<=n-1;j++)
    {
        fprintf(fichero, "\nR3.%d.%d: E[%d] [%d]", r, j, r, j);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " + %d Z[%d] [%d]", p_ri[r-1][i-1], i, j+1);
        }
        fprintf(fichero, " - E[%d] [%d] <= 0", r, j+1);
    }
}
//R4
for(r=1;r<=m-1;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nR4.%d.%d: E[%d] [%d]", r, j, r, j);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " + %d Z[%d] [%d]", p_ri[r][i-1], i, j);
        }
        fprintf(fichero, " - E[%d] [%d] <= 0", r+1, j);
    }
}
//R5
fprintf(fichero, "\nR5: E[1] [1]");
for(i=1;i<=n;i++)
{
    fprintf(fichero, " - %d Z[%d] [1]", p_ri[0][i-1], i);
}
fprintf(fichero, " >= 0");
//R6
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "\nR6.%d.%d.%d: E[%d] [%d] + %d Bin[%d] [%d] [%d]
                <= %d", r, j, b, r, j, P, r, j, b, P+(b*T));
        }
    }
}
//R7
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {

```

```

        fprintf(fichero, "\nR7.%.%d.%.%d: E[%d] [%d]",r,j,b,r,j);
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, " - %d Z[%d] [%d]",p_ri[r-1][i-1],i,j);
        }
        fprintf(fichero, " - %d Bin[%d] [%d] [%d] >= %d",P,r,j,b, T*(b-1)-
            P);
    }
}
//R8
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nR8.%.%d.%.%d: Bin[%d] [%d] [1]",r,j,r,j);
        for(b=2;b<=Bines;b++)
        {
            fprintf(fichero, " + Bin[%d] [%d] [%d]",r,j,b);
        }
        fprintf(fichero, " = 1");
    }
}
//R9
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nR11.%.%d: td[%d] - E[%d] [%d] + d[%d] >= 0",j,j,m,j,j);
}
//R10
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nR12.%.%d: d[%d]",j,j,di[0],j);
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, " - %d Z[%d] [%d]",di[0][i-1],i,j);
    }
    fprintf(fichero, " = 0");
}
//Bounds
fprintf(fichero, "\nBounds");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\nE[%d] [%d] >= 0",r,j);
    }
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\ntd[%d] >= 0",j);
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "\nd[%d] >= 0",j);
}
for(i=1;i<=n;i++)

```

```

{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "\n0 <= Z[%d] [%d] <= 1",i,j);
    }
}
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",r,j,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(i=1;i<=n;i++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "Z[%d] [%d] ",i,j);
    }
}
fprintf(fichero, "\n");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",r,j,b);
        }
    }
}
//General
fprintf(fichero, "\nGeneral\n");
for(r=1;r<=m;r++)
{
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, "E[%d] [%d] ",r,j);
    }
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "td[%d] ",j);
}
for(j=1;j<=n;j++)
{
    fprintf(fichero, "d[%d] ",j);
}
fprintf(fichero, "\nEnd");
fclose(fichero);

```

```

return 0;
}

```

A.0.4 Modelo Idle-time

```

modelo_idle_time(MAT_INT p_ri, int n, int m, int Bines, int T, int P, MAT_INT
di, char **argv)
{
    int k,j,i,u,l,b;
    FILE *fichero=fopen(argv[5],"w");
    fprintf(fichero,"Minimize\n");
    //F0
    fprintf(fichero,"td[1]");
    for(k=2;k<=n;k++)
    {
        fprintf(fichero, " + td[%d]",k);
    }

    //Restricciones
    fprintf(fichero, "\nSubject To");
    //R1
    for(k=1; k<=n;k++)
    {
        fprintf(fichero,"\nR1.%d:",k);
        fprintf(fichero," x[1] [%d]",k);
        for(j=2;j<=n;j++)
        {
            fprintf(fichero, " + x[%d] [%d]",j,k);
        }
        fprintf(fichero, " = 1");
    }
    //R2
    for(j=1; j<=n;j++)
    {
        fprintf(fichero,"\nR2.%d:",j);
        fprintf(fichero," x[%d] [1]",j);
        for(k=2;k<=n;k++)
        {
            fprintf(fichero, " + x[%d] [%d]",j,k);
        }
        fprintf(fichero, " = 1");
    }
    //R3
    for(k=1;k<=n-1;k++)
    {
        for(i=1;i<=m-1;i++)
        {
            fprintf(fichero, "\nR3.%d.%d: I[%d] [%d]",k,i,i,k+1);
            for(u=1;u<=n;u++)
            {
                fprintf(fichero, " + %d x[%d] [%d]",p_ri[i-1][u-1],u,k+1);
            }
            fprintf(fichero, " + H[%d] [%d] - H[%d] [%d]",i,k+1,i,k);
            for(u=1;u<=n;u++)
            {

```

```

        fprintf(fichero, " - %d x[%d] [%d]", p_ri[i][u-1],u,k);
    }
    fprintf(fichero, " - I[%d] [%d] = 0",i+1,k+1);
}
}
//R4
for(i=1;i<=m-1;i++)
{
    fprintf(fichero, "\nR4.%d: I[%d] [1]",i,i);
    for(u=1;u<=n;u++)
    {
        fprintf(fichero, " + %d x[%d] [1]", p_ri[i-1][u-1],u);
    }
    fprintf(fichero, " + H[%d] [1] - I[%d] [1] = 0",i,i+1);
}
//R5
fprintf(fichero, "\nR5: C[%d] [0] = 0",m);
//R6
for(k=1;k<=n;k++)
{
    fprintf(fichero, "\nR6.%d: td[%d] - C[%d] [%d] + d[%d] >= 0",k,k,m,k,k);
}
//R7
for(k=1;k<=n;k++)
{
    fprintf(fichero, "\nR7.%d: d[%d]",k,k);
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, " - %d x[%d] [%d]", di[0][j-1],j,k);
    }
    fprintf(fichero, " = 0");
}
//R8
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\nR8.%d.%d: C[%d] [%d]",i,k,i,k);
        for(l=1;l<=k;l++)
        {
            fprintf(fichero, " - I[%d] [%d]",i,l);
            for(j=1;j<=n;j++)
            {
                fprintf(fichero, " - %d x[%d] [%d]",p_ri[i-1][j-1],j,l);
            }
        }
        fprintf(fichero, " = 0");
    }
}
//R9
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)
        {

```

```

        fprintf(fichero, "\nR9.%d.%d.%d: C[%d] [%d] + %d Bin[%d] [%d] [%d]
                <= %d",i,k,b,i,k,P,i,k,b,b*T+P);
    }
}
//R10
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "\nR10.%d.%d.%d: C[%d] [%d]",i,k,b,i,k);
            for(j=1;j<=n;j++)
            {
                fprintf(fichero, " - %d x[%d] [%d]", p_ri[i-1][j-1],j,k);
            }
            fprintf(fichero, " - %d Bin[%d] [%d] [%d] >= %d",P,i,k,b,T*(b-1)-P
                    );
        }
    }
}
//R11
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\nR11.%d.%d: Bin[%d] [%d] [1] ",i,k,i,k);
        for(b=2;b<=Bines;b++)
        {
            fprintf(fichero, " + Bin[%d] [%d] [%d]",i,k,b);
        }
        fprintf(fichero, " = 1");
    }
}
//Bounds
fprintf(fichero, "\nBounds");
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\nC[%d] [%d] >= 0",i,k);
    }
}
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\nI[%d] [%d] >= 0",i,k);
    }
}
for(k=1;k<=m;k++)
{
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, "\nH[%d] [%d] >= 0",k,i);
    }
}

```

```

}
for(k=1;k<=n;k++)
{
    fprintf(fichero, "\ntd[%d] >= 0",k);
}
for(k=1;k<=n;k++)
{
    fprintf(fichero, "\nd[%d] >= 0",k);
}
for(j=1;j<=n;j++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\n0 <= x[%d] [%d] <= 1",j,k);
    }
}
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",i,k,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(j=1;j<=n;j++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "x[%d] [%d] ",j,k);
    }
}
fprintf(fichero, "\n");
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",i,k,b);
        }
    }
}
//General
fprintf(fichero, "\nGeneral\n");
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "C[%d] [%d] ",i,k);
    }
}
}

```

```

for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "I[%d] [%d] ",i,k);
    }
}
for(k=1;k<=m;k++)
{
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, "H[%d] [%d] ",k,i);
    }
}
for(k=1;k<=n;k++)
{
    fprintf(fichero, "td[%d] ",k);
}
for(k=1;k<=n;k++)
{
    fprintf(fichero, "d[%d] ",k);
}
fprintf(fichero, "\nEnd");
fclose(fichero);
return 0;
}

```

A.0.5 Modelo Completion-time

```

modelo_completion_time(MAT_INT p_ri, int n, int m, int Bines, int T, int P,
    MAT_INT di, char **argv)
{
    int k,i,j,b;
    FILE *fichero=fopen(argv[5],"w");
    fprintf(fichero,"Minimize\n");
    //F0
    fprintf(fichero,"td[1]");
    for(k=2;k<=n;k++)
    {
        fprintf(fichero, " + td[%d]",k);
    }

    //Restricciones
    fprintf(fichero, "\nSubject To");
    //R1
    for(k=1; k<=n;k++)
    {
        fprintf(fichero,"\nR1.%d:",k);
        fprintf(fichero," x[1] [%d]",k);
        for(j=2;j<=n;j++)
        {
            fprintf(fichero, " + x[%d] [%d]",j,k);
        }
        fprintf(fichero, " = 1");
    }
    //R2
}

```

```

for(j=1; j<=n;j++)
{
    fprintf(fichero, "\nR2.%d:",j);
    fprintf(fichero, " x[%d] [1]",j);
    for(k=2;k<=n;k++)
    {
        fprintf(fichero, " + x[%d] [%d]",j,k);
    }
    fprintf(fichero, " = 1");
}
//R3
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\nR3.%d.%d: C[%d] [%d] - C[%d] [%d]",i,k,i,k,i-1,k);
        for(j=1;j<=n;j++)
        {
            fprintf(fichero, " - %d x[%d] [%d]",p_ri[i-1][j-1],j,k);
        }
        fprintf(fichero, " >= 0");
    }
}
//R4
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\nR4.%d.%d: C[%d] [%d] - C[%d] [%d]",i,k,i,k,i,k-1);
        for(j=1;j<=n;j++)
        {
            fprintf(fichero, " - %d x[%d] [%d]",p_ri[i-1][j-1],j,k);
        }
        fprintf(fichero, " >= 0");
    }
}
//R5
for(k=1;k<=n;k++)
{
    fprintf(fichero, "\nR5.%d: td[%d] - C[%d] [%d] + d[%d] >= 0",k,k,m,k,k);
}
//R6
for(k=1;k<=n;k++)
{
    fprintf(fichero, "\nR6.%d: d[%d]",k,k);
    for(j=1;j<=n;j++)
    {
        fprintf(fichero, " - %d x[%d] [%d]", di[0][j-1],j,k);
    }
    fprintf(fichero, " = 0");
}
//R7
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)

```

```

        {
            fprintf(fichero, "\nR7.%d.%d.%d: C[%d] [%d] + %d Bin[%d] [%d] [%d]
                <= %d", i, k, b, i, k, P, i, k, b, b*T+P);
        }
    }
}
//R8
for(i=1; i<=m; i++)
{
    for(k=1; k<=n; k++)
    {
        for(b=1; b<=Bines; b++)
        {
            fprintf(fichero, "\nR8.%d.%d.%d: C[%d] [%d]", i, k, b, i, k);
            for(j=1; j<=n; j++)
            {
                fprintf(fichero, " - %d x[%d] [%d]", p_ri[i-1][j-1], j, k);
            }
            fprintf(fichero, " - %d Bin[%d] [%d] [%d] >= %d", P, i, k, b, T*(b-1)-
                P);
        }
    }
}
//R9
for(i=1; i<=m; i++)
{
    for(k=1; k<=n; k++)
    {
        fprintf(fichero, "\nR9.%d.%d: Bin[%d] [%d] [1]", i, k, i, k);
        for(b=2; b<=Bines; b++)
        {
            fprintf(fichero, " + Bin[%d] [%d] [%d]", i, k, b);
        }
        fprintf(fichero, " = 1");
    }
}
//Bounds
fprintf(fichero, "\nBounds");
for(i=1; i<=m; i++)
{
    for(k=1; k<=n; k++)
    {
        fprintf(fichero, "\nC[%d] [%d] >= 0", i, k);
    }
}
for(k=1; k<=n; k++)
{
    fprintf(fichero, "\ntd[%d] >= 0", k);
}
for(k=1; k<=n; k++)
{
    fprintf(fichero, "\nd[%d] >= 0", k);
}
for(j=1; j<=n; j++)
{
    for(k=1; k<=n; k++)
    {

```

```

        fprintf(fichero, "\n0 <= x[%d] [%d] <= 1",j,k);
    }
}
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",i,k,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(j=1;j<=n;j++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "x[%d] [%d] ",j,k);
    }
}
fprintf(fichero, "\n");
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",i,k,b);
        }
    }
}
//General
fprintf(fichero, "\nGeneral\n");
for(i=1;i<=m;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "C[%d] [%d] ",i,k);
    }
}
for(k=1;k<=n;k++)
{
    fprintf(fichero, "td[%d] ",k);
}
for(k=1;k<=n;k++)
{
    fprintf(fichero, "d[%d] ",k);
}
fprintf(fichero, "\nEnd");
fclose(fichero);
return 0;
}

```

A.0.6 Modelo SGST

```

modelo_SGST(MAT_INT p_ri, int n, int m, int Bines,int T, int P, MAT_INT di,char
**argv)
{
    int i,r,k,b;
    FILE *fichero=fopen(argv[5],"w");
    fprintf(fichero,"Minimize\n");
    //F0
    fprintf(fichero,"td[1]");
    for(i=2;i<=n;i++)
    {
        fprintf(fichero, " + td[%d]",i);
    }

    //Restricciones
    fprintf(fichero, "\nSubject To");
    //R1
    for(i=1; i<=n;i++)
    {
        fprintf(fichero, "\nR1.%d: C[1][%d] >= %d",i,i,p_ri[0][i-1]);
    }
    //R2
    for(r=1;r<=m-1;r++)
    {
        for(i=1;i<=n;i++)
        {
            fprintf(fichero, "\nR2.%d.%d: C[%d][%d] - C[%d][%d] >= %d",r,i,r+1,i
                ,r,i,p_ri[r][i-1]);
        }
    }
    //R3
    {
        for(r=1;r<=m;r++)
        {
            for(i=1;i<=n-1;i++)
            {
                for(k=i+1;k<=n;k++)
                {
                    fprintf(fichero, "\nR3.%d.%d.%d: C[%d][%d] - C[%d][%d] + %d D
                        [%d][%d] >= %d",r,i,k,r,i,r,k,P,i,k,p_ri[r-1][i-1]);
                }
            }
        }
    }
    //R4
    {
        for(r=1;r<=m;r++)
        {
            for(i=1;i<=n-1;i++)
            {
                for(k=i+1;k<=n;k++)
                {
                    fprintf(fichero, "\nR4.%d.%d.%d: C[%d][%d] - C[%d][%d] - %d D
                        [%d][%d] >= %d",r,i,k,r,k,r,i,P,i,k,p_ri[r-1][k-1]-P);
                }
            }
        }
    }
}

```

```

    }
  }
}
//R5
for(r=1;r<=m;r++)
{
  for(i=1;i<=n;i++)
  {
    for(b=1;b<=Bines;b++)
    {
      fprintf(fichero, "\nR5.%d.%d.%d: C[%d] [%d] + %d Bin[%d] [%d] [%d]
        <= %d",r,i,b,r,i,P,r,i,b, P+(b*T));
    }
  }
}
//R6
for(r=1;r<=m;r++)
{
  for(i=1;i<=n;i++)
  {
    for(b=1;b<=Bines;b++)
    {
      fprintf(fichero, "\nR6.%d.%d.%d: C[%d] [%d] - %d Bin[%d] [%d] [%d]
        >= %d",r,i,b,r,i,P,r,i,b,p_ri[r-1][i-1]-P+T*(b-1));
    }
  }
}
//R7
for(r=1;r<=m;r++)
{
  for(i=1;i<=n;i++)
  {
    fprintf(fichero, "\nR7.%d.%d: Bin[%d] [%d] [1]",r,i,r,i);
    for(b=2;b<=Bines;b++)
    {
      fprintf(fichero, " + Bin[%d] [%d] [%d]",r,i,b);
    }
    fprintf(fichero, " = 1");
  }
}
//R8
for(i=1;i<=n;i++)
{
  fprintf(fichero, "\nR8.%d: td[%d] - C[%d] [%d] >= %d",i,i,m,i,-di[0][i
    -1]);
}
//Bounds
fprintf(fichero, "\nBounds");
for(r=1;r<=m;r++)
{
  for(i=1;i<=n;i++)
  {
    fprintf(fichero, "\nC[%d] [%d] >= 0",r,i);
  }
}
for(i=1;i<=n;i++)

```

```

{
    fprintf(fichero, "\ntd[%d] >= 0",i);
}
for(i=1;i<=n;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\n0 <= D[%d] [%d] <= 1",i,k);
    }
}
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",r,i,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(i=1;i<=n;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "D[%d] [%d] ",i,k);
    }
}
fprintf(fichero, "\n");
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",r,i,b);
        }
    }
}
//General
fprintf(fichero, "\nGeneral\n");
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, "C[%d] [%d] ",r,i);
    }
}
for(i=1;i<=n;i++)
{
    fprintf(fichero, "td[%d] ",i);
}
fprintf(fichero, "\nEnd");
fclose(fichero);

```

```

return 0;
}

```

A.0.7 Modelo LYeq

```

modelo_LYeq(MAT_INT p_ri, int n, int m, int Bines,int T, int P, MAT_INT di,char
**argv)
{
int i,r,k,b;
FILE *fichero=fopen(argv[5],"w");
fprintf(fichero,"Minimize\n");
//F0
fprintf(fichero,"td[1]");
for(i=2;i<=n;i++)
{
fprintf(fichero, " + td[%d]",i);
}

//Restricciones
fprintf(fichero, "\nSubject To");
//R1
for(i=1; i<=n;i++)
{
fprintf(fichero, "\nR1.%d: C[1] [%d] >= %d",i,i,p_ri[0][i-1]);
}
//R2
for(r=1;r<=m-1;r++)
{
for(i=1;i<=n;i++)
{
fprintf(fichero, "\nR2.%d.%d: C[%d] [%d] - C[%d] [%d] >= %d",r,i,r+1,i
,r,i,p_ri[r][i-1]);
}
}
//R3
{
for(r=1;r<=m;r++)
{
for(i=1;i<=n-1;i++)
{
for(k=i+1;k<=n;k++)
{
fprintf(fichero, "\nR3.%d.%d.%d: %d D[%d] [%d] + C[%d] [%d] - C
[%d] [%d] - Q[%d] [%d] [%d] = %d",r,i,k,P,i,k,r,i,r,k,r,i,k,
p_ri[r-1][i-1]);
}
}
}
}
//R4
{
for(r=1;r<=m;r++)
{
for(i=1;i<=n-1;i++)

```

```

        {
            for(k=i+1;k<=n;k++)
            {
                fprintf(fichero, "\nR4.%d.%d.%d: Q[%d] [%d] [%d] <= %d",r,i,k,r
                    ,i,k,P-p_ri[r-1][i-1]-p_ri[r-1][k-1]);
            }
        }
    }
}
//R5
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "\nR5.%d.%d.%d: C[%d] [%d] + %d Bin[%d] [%d] [%d]
                <= %d",r,i,b,r,i,P,r,i,b,P+(b*T));
        }
    }
}
//R6
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "\nR6.%d.%d.%d: C[%d] [%d] - %d Bin[%d] [%d] [%d]
                >= %d",r,i,b,r,i,P,r,i,b,T*(b-1)+p_ri[r-1][i-1]-P);
        }
    }
}
//R7
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, "\nR7.%d.%d: Bin[%d] [%d] [1] ",r,i,r,i);
        for(b=2;b<=Bines;b++)
        {
            fprintf(fichero, " + Bin[%d] [%d] [%d]",r,i,b);
        }
        fprintf(fichero, " = 1");
    }
}
//R8
for(i=1;i<=n;i++)
{
    fprintf(fichero, "\nR8.%d: td[%d] - C[%d] [%d] >= %d",i,i,m,i,-di[0][i
        -1]);
}
//Bounds
fprintf(fichero, "\nBounds");
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)

```

```

    {
        fprintf(fichero, "\nC[%d] [%d] >= 0",r,i);
    }
}
for(r=1;r<=m;r++)
{
    for(i=1;i<=m;i++)
    {
        for(k=1;k<=n;k++)
        {
            fprintf(fichero, "\nQ[%d] [%d] [%d] >= 0",r,i,k);
        }
    }
}
for(i=1;i<=n;i++)
{
    fprintf(fichero, "\ntd[%d] >= 0",i);
}
for(i=1;i<=n;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "\n0 <= D[%d] [%d] <= 1",i,k);
    }
}
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        for(b=1;b<=Bines;b++)
        {
            {
                fprintf(fichero, "\n0 <= Bin[%d] [%d] [%d] <= 1",r,i,b);
            }
        }
    }
}
//Binary
fprintf(fichero, "\nBinary\n");
for(i=1;i<=n;i++)
{
    for(k=1;k<=n;k++)
    {
        fprintf(fichero, "D[%d] [%d] ",i,k);
    }
}
fprintf(fichero, "\n");
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        for(b=1;b<=Bines;b++)
        {
            fprintf(fichero, "Bin[%d] [%d] [%d] ",r,i,b);
        }
    }
}
}

```

```
//General
fprintf(fichero, "\nGeneral\n");
for(r=1;r<=m;r++)
{
    for(i=1;i<=n;i++)
    {
        fprintf(fichero, "C[%d] [%d] ",r,i);
    }
}
for(r=1;r<=m;r++)
{
    for(i=1;i<=m;i++)
    {
        for(k=1;k<=n;k++)
        {
            fprintf(fichero, "Q[%d] [%d] [%d] ",r,i,k);
        }
    }
}
for(i=1;i<=n;i++)
{
    fprintf(fichero, "td[%d] ",i);
}
fprintf(fichero, "\nEnd");
fclose(fichero);
return 0;
}
```


Índice de Figuras

2.1	Diagrama de Gantt sin restricción de disponibilidad en las máquinas	4
2.2	Diagrama de Gantt con restricción de disponibilidad en las máquinas	4
4.1	Medias de cada modelo y media total	23
4.2	Diagrama de cajas y bigotes. <i>CPU Time</i> por modelo	23
4.3	Diagrama de cajas y bigotes reducido. <i>CPU Time</i> por modelo	24
4.4	Función de densidad para cada periodo de disponibilidad (T)	25
4.5	Diagrama de cajas y bigotes. <i>CPU Time</i> por modelo y periodo de disponibilidad (T)	26
4.6	Diagrama de cajas y bigotes. <i>CPU Time</i> por modelo y periodo de disponibilidad (T) para valores del <i>CPU Time</i> menores que 6.	26
4.7	Media y desviación estándar de cada modelo para cada número de trabajos	27
4.8	Media y desviación estándar de cada modelo para cada número de máquinas	28

Índice de Tablas

4.1	Modelos objeto de estudio	22
4.2	<i>CPU Time (s)</i> medio, desviación estándar y porcentaje de instancias para las que se garantiza optimalidad por modelo.	22
4.3	<i>CPU Time(s)</i> medio y desviación estándar en función del modelo y el periodo de disponibilidad (T)	27
4.4	<i>CPU Time(s)</i> medio y desviación estándar en función del modelo y el número de trabajos (N).	28
4.5	<i>CPU Time(s)</i> medio y desviación estándar en función del modelo y el número de máquinas (M)	29

Bibliografía

- Baker, K. R. (2013). Computational results for the flowshop tardiness problem. *Computers and Industrial Engineering*, 64(3):812–816.
- Framinan, J. M., Leisten, R., and Ruiz García, R. (2014). *Manufacturing Scheduling Systems. An Integrated View on Models, Methods and Tools*.
- Graham, R., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. (1979). Optimization and heuristic in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326.
- Ramos Salgado, C. (2017). Soluciones exactas para el problema de flowshop de permutación con restricciones de disponibilidad periódicas.
- Stafford, E. F., Tseng, F. T., and Gupta, J. N. (2005). Comparative evaluation of MILP flowshop models. *Journal of the Operational Research Society*, 56(1):88–101.