

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

Análisis de un problema de secuenciación en un taller de flujo con dos conjuntos de trabajos

Autor: Jesús Jiménez Blázquez

Tutora: Paz Pérez González

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

Sevilla, 2020



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Análisis de un problema de secuenciación en un taller de flujo con dos conjuntos de trabajos

Autor:

Jesús Jiménez Blázquez

Tutora:

Paz Pérez González

Dep. Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Grado: Análisis de un problema de secuenciación en un taller de flujo con dos conjuntos de trabajos

Autor: Jesús Jiménez Blázquez

Tutor: Paz Pérez González

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A todo aquel que siempre estuvo conmigo.

Agradecimientos

Tras cuatro intensos años de sacrificio y perseverancia, han sido muchas las personas que me han ayudado a lidiar con los sinsabores que entraña la carrera. Es por ello que estoy enormemente agradecido a mis familiares y amigos por haberse volcado conmigo en esta dura andadura estudiantil. Sin vuestro apoyo todo hubiera sido más complejo y menos llevadero, por lo que no sé si hubiera llegado hasta aquí.

De igual forma, han sido muchos los profesores que he conocido desde que comencé mi formación académica cuando era un crío hasta la presente fecha. Me siento muy afortunado de haber podido adquirir gran parte de los conocimientos que me habéis tratado de inculcar.

A todos vosotros, simplemente gracias.

Jesús Jiménez Blázquez

Graduado en Ingeniería de Tecnologías Industriales

Sevilla, 2020

Resumen

En este presente documento se aborda el análisis de un problema de secuenciación en un taller de flujo con dos máquinas y dos conjuntos de trabajos. Más concretamente, el objetivo es estudiar la resolución en un entorno *Flow Shop* de un problema multicliente cuando los objetivos son la tardanza total y el tiempo de finalización máximo.

Para llevar a cabo la investigación, se han configurado las baterías de problemas mediante un modelo de programación lineal entera mixta capaz de resolverlas de manera eficiente. La creación de dicho modelo se ha realizado mediante la herramienta MATLAB[®] mientras que la resolución de cada problema se ha efectuado mediante el optimizador de GUROBI[®].

Finalmente, se detallan los resultados obtenidos mediante tablas y se procede al análisis del tiempo computacional de GUROBI[®], así como al de los valores de las funciones objetivo.

Abstract

This document deals with the analysis of a two-agent scheduling problem in a two-machine flowshop. More specifically, the goal is to study the resolution of a flowshop environment of a multi-agent problem when the objectives are the total tardiness and the maximum completion time.

To carry out the research, the problem batteries have been configured using a mixed integer linear programming model able to solve them efficiently. The creation of said model have been conducted using the MATLAB[®] tool, while the resolution of each problem has been carried out using the GUROBI[®] optimizer.

Finally, the results obtained by means of tables are detailed and the computational time of GUROBI[®] is analyzed, as well as that of the values of the objective functions.

Agradecimientos	viii
Resumen	x
Abstract	xii
Índice	xiii
Índice de tablas	xv
Índice de figuras	xvi
1 Introducción	1
1.1 <i>Objetivos</i>	1
1.2 <i>Estructura del documento</i>	1
2 Programación de la Producción	3
2.1 <i>Introducción</i>	3
2.2 <i>Fundamentación teórica</i>	5
2.2.1 <i>Conceptos básicos</i>	5
2.2.2 <i>Modelos de programación de la producción</i>	6
2.2.3 <i>Complejidad computacional</i>	11
2.2.4 <i>Métodos de programación de la producción</i>	12
2.2.5 <i>Programación multiobjetivo</i>	13
3 Descripción del problema	15
3.1 <i>Descripción del problema</i>	15
3.2 <i>Modelo matemático de programación lineal entera mixta</i>	16
3.3 <i>Modelado matemático del problema</i>	17
3.4 <i>Generación de los parámetros del modelo</i>	19
3.4.1 <i>Generación del valor de Q</i>	19
3.4.2 <i>Generación de los valores de d_j</i>	20
3.5 <i>Ejemplo numérico</i>	20
4 Resolución exacta del problema	25
4.1 <i>Introducción</i>	25
4.2 <i>Matlab</i>	25
4.3 <i>Gurobi</i>	26
4.4 <i>Batería de problemas</i>	28
4.5 <i>Resultados obtenidos</i>	31
5 Análisis de resultados	37
5.1 <i>Batería pequeña</i>	37
5.2 <i>Batería mediana</i>	40
5.3 <i>Batería grande</i>	44
5.4 <i>Análisis de resultados entre las distintas baterías de problemas</i>	49
6 Conclusiones	53
Bibliografía	54
Anexo A	55

ÍNDICE DE TABLAS

Tabla 2.1 Cálculo de las principales funciones objetivo (Framiñán, Leisten and Ruiz, 2014).	10
Tabla 3.1 Datos del ejemplo numérico del problema.	20
Tabla 4.1 Tiempo de ejecución de Gurobi (s). Batería pequeña.	31
Tabla 4.2 Total Tardiness de A (u.t). Batería pequeña.	32
Tabla 4.3 Makespan de B (u.t). Batería pequeña.	32
Tabla 4.4 Valores de Q (u.t). Batería pequeña.	32
Tabla 4.5 Optimalidad (%). Batería pequeña.	33
Tabla 4.6 Tiempo de ejecución de Gurobi (s). Batería mediana.	33
Tabla 4.7 Total Tardiness de A (u.t). Batería mediana.	33
Tabla 4.8 Makespan de B (u.t). Batería mediana.	34
Tabla 4.9 Valores de Q (u.t). Batería mediana.	34
Tabla 4.10 Optimalidad (%). Batería mediana.	34
Tabla 4.11 Tiempo de ejecución de Gurobi (s). Batería grande.	35
Tabla 4.12 Total Tardiness de A (u.t). Batería grande.	35
Tabla 4.13 Makespan de B (u.t). Batería grande.	35
Tabla 4.14 Valores de Q (u.t). Batería grande.	35
Tabla 4.15 Optimalidad (%). Batería grande.	35

ÍNDICE DE FIGURAS

Figura 2.1 Diagramas de Gantt orientados a máquina y trabajo (Framiñán, Leisten and Ruiz, 2014)	3
Figura 2.2 Fases del sistema de producción (Pérez González, Paz; Framiñán Torres, José M; Fernández-Viagas, 2014).	4
Figura 2.3 Clases de programas no interrumpibles para job shop (Framiñán, Leisten and Ruiz, 2014).	6
Figura 2.4 Clasificación de los modelos de programación de la producción (Framiñán, Leisten and Ruiz, 2014).	6
Figura 2.5 Máquina única (Framiñán, Leisten and Ruiz, 2014).	7
Figura 2.6 Máquinas paralelas idénticas (Framiñán, Leisten and Ruiz, 2014).	7
Figura 2.7 Flow Shop (Framiñán, Leisten and Ruiz, 2014).	8
Figura 2.8 Job Shop (Framiñán, Leisten and Ruiz, 2014).	8
Figura 2.9 Open Shop (Framiñán, Leisten and Ruiz, 2014).	8
Figura 2.10 Hybrid layout (Framiñán, Leisten and Ruiz, 2014).	9
Figura 2.11 Tipología de problemas de programación de la producción (Maldonado, 2013).	12
Figura 2.12 Espacio de búsqueda de un Branch and Bound (Clausen, 1999).	13
Figura 2.13 Frente de Pareto (Framiñán, Leisten and Ruiz, 2014).	14
Figura 3.1 Evolución del número de programas admisibles.	15
Figura 3.2 Pseudocódigo de la regla de Johnson (Framiñán, Leisten and Ruiz, 2014).	19
Figura 3.3 Resolución del ejemplo en Excel para $\varepsilon = 0$.	21
Figura 3.4 Diagrama de Gantt del ejemplo para $\varepsilon = 0$.	22
Figura 3.5 Resolución del ejemplo en Excel para $\varepsilon = -0.25$.	22
Figura 3.6 Diagrama de Gantt del ejemplo para $\varepsilon = -0.25$.	23
Figura 4.1 Esquema funcionamiento de Gurobi (http://www.gurobi.com).	26
Figura 4.2 Ejemplo de instancia de 8 trabajos.	28
Figura 4.3 Funcionamiento de ejecución.	29
Figura 4.4 Flujograma del código en Matlab.	30
Figura 4.5 Ejemplo de archivo de salida para 8 trabajos y $\varepsilon = 0$.	30
Figura 5.1 Tiempo de ejecución medio de Gurobi (s). Batería pequeña.	37
Figura 5.2 Comparativa de la optimalidad (%). Batería pequeña.	38
Figura 5.3 Comparativa de la tardanza total del cliente A (u.t). Batería pequeña.	38
Figura 5.4 Comparativa del makespan del cliente B (u.t). Batería pequeña.	39
Figura 5.5 Comparativa de los valores de Q (u.t). Batería pequeña.	39
Figura 5.6 Comparativa entre C_{\max} y Q (u.t). Batería pequeña.	40
Figura 5.7 Tiempo de ejecución medio de Gurobi (s). Batería mediana.	40
Figura 5.8 Comparativa de la optimalidad (%). Batería mediana.	41
Figura 5.9 Tiempos de ejecución de Gurobi (s) para $n=28$.	41
Figura 5.10 Tiempos de ejecución de Gurobi (s) para $n=30$.	42
Figura 5.11 Comparativa de la tardanza total del cliente A (u.t). Batería mediana.	42

Figura 5.12 Comparativa del makespan del cliente B (u.t).	43
Figura 5.13 Comparativa de los valores de Q (u.t). Batería mediana.	43
Figura 5.14 Comparativa entre C_{\max} y Q (u.t). Batería mediana.	44
Figura 5.15 Tiempo de ejecución medio de Gurobi (s). Batería grande.	44
Figura 5.16 Tiempos de ejecución de Gurobi (s) para $n=35$.	45
Figura 5.17 Tiempos de ejecución de Gurobi (s) para $n=40$.	45
Figura 5.18 Tiempos de ejecución de Gurobi (s) para $n=45$.	46
Figura 5.19 Tiempos de ejecución de Gurobi (s) para $n=50$.	46
Figura 5.20 Comparativa de la optimalidad (%). Batería grande.	47
Figura 5.21 Comparativa de la tardanza total del cliente A (u.t). Batería grande.	47
Figura 5.22 Comparativa del makespan del cliente B (u.t). Batería grande.	48
Figura 5.23 Comparativa de los valores de Q (u.t). Batería grande.	48
Figura 5.24 Comparativa entre C_{\max} y Q (u.t). Batería grande.	49
Figura 5.25 Evolución de los tiempos medios (s) de Gurobi.	49
Figura 5.26 Análisis de la optimalidad (%) en función de ϵ .	50
Figura 5.27 Análisis general de la optimalidad (%) de Gurobi.	50
Figura 5.28 Análisis general de la tardanza total del cliente A (u.t).	51
Figura 5.29 Análisis general de C_{\max} y Q (u.t).	51

1 INTRODUCCIÓN

1.1 Objetivos

El principal objetivo de este Trabajo Fin de Grado es estudiar, analizar y resolver un problema de programación de la producción. En concreto, se pretende secuenciar los trabajos de manera eficiente de dos clientes distintos en un taller de flujo. Para lograr esto, se asignarán los distintos recursos a los diferentes trabajos provenientes de ambos entornos y cuyos tiempos de proceso y fechas de entrega vienen previamente establecidos.

La programación de la producción es una tarea compleja pues, en la mayoría de ocasiones, no existen soluciones exactas para un número de trabajos elevado, tal y como ocurre en una empresa real. Cada industria realiza la programación de la producción de manera particular, sin atender a las reglas que dicta la programación. Esto conlleva a errores y a pérdidas abismales de dinero ya que se evita el cambio a la programación por el mero hecho de que hasta la presente fecha la empresa ha trabajado de una forma determinada. Por esta razón, es de vital importancia la implantación de la programación, lo que supone un aumento de beneficios, optimizando el rendimiento de la planta. Actualmente, es primordial cumplir con los plazos de entrega, siendo necesario para ello cumplir, a su vez, con los plazos de producción. Por consiguiente, la programación de la producción se convierte en una herramienta de primera necesidad dentro de cualquier industria, tanto en empresas de servicios para disminuir la mano de obra como en empresas de logística para reducir al máximo los costes de transporte (y logística inversa).

En una planta productiva, los recursos que aparecen van desde las propias máquinas que posee la empresa hasta los operarios. Los trabajos pueden ser operaciones simples, como por ejemplo el cortado de una chapa, u operaciones concatenadas que se realicen en una determinada sección de la fábrica. Los objetivos principales son muy diversos, por ejemplo, minimizar la tardanza total, el tiempo de finalización máximo o los retrasos. Además, es muy relevante el entorno de producción ya que no es lo mismo trabajar en un entorno de varias máquinas en paralelo o en un taller de flujo; la manera de secuenciación es completamente diferente.

En este documento se pretende dar solución a un problema de secuenciación en un taller de flujo (*Flow Shop*) con dos máquinas y dos conjuntos de trabajos o clientes diferentes. El objetivo es analizar cómo varían los tiempos computacionales del problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{\text{máx}}^B \leq Q$. Además, los criterios que se pretenden minimizar son: la tardanza total del primer cliente y el tiempo de finalización máximo del segundo cliente. Para dar solución a este problema, se ha modelado, generando un modelo de programación lineal entera mixta (MILP).

1.2 Estructura del documento

Este documento se compone de un total de 6 capítulos, en los que se estudia y se analiza el problema de programación de la producción, y dos anexos, en los que se muestra el código del modelo de programación lineal en la herramienta Matlab (*Anexo A*) y todos los datos obtenidos de la resolución del problema (*Anexo B*).

En el capítulo 2 se describe la programación de la producción, empezando por una breve reseña histórica y continuando con los conceptos básicos empleados en este documento relacionados con ella.

Bajo este contexto teórico, en el capítulo 3 se procede a la explicación de manera detallada de un taller de flujo, enlazando con el concepto de programación multiobjetivo. Tras esto, se explica el problema que es objeto de estudio en este Trabajo Fin de Grado, realizando una breve descripción del mismo. Además, se explica el concepto de modelo de programación lineal entera mixta para entender bien el modo de resolución del problema en cuestión. En suma, se expresa el modelo matemático del problema, así como su implementación mediante la herramienta Matlab. Finalmente, se ilustra el problema con un ejemplo numérico sencillo que expone la idea de resolución y cómo se ordenan los trabajos dentro de la secuencia. El ejemplo se resuelve con el solver de Excel dado que se trata de un ejemplo sencillo con un número de trabajos muy reducido.

En el capítulo 4, se mostrarán los programas necesarios que permitirán visualizar los resultados obtenidos. Concretamente, se ha utilizado el lenguaje de programación de Matlab para generar el modelo de programación lineal que se pasa al solver. Este modelo será resuelto posteriormente por el optimizador de Gurobi, proporcionando el valor de la función objetivo para los conjuntos de trabajos de los dos clientes y el tiempo de cómputo necesario para obtener la solución para cada instancia del problema. Después de ello, se procede a explicar cómo se han ejecutado las tres baterías de problemas. Una vez presentados los programas necesarios para resolver de forma óptima el problema y las características de las instancias de los problemas que se van a resolver, el capítulo concluye con la presentación de los resultados obtenidos.

En el capítulo 5 se analizan los distintos resultados obtenidos en cada una de las tres baterías de problemas mediante gráficas que muestran la evolución de las funciones objetivo y el tiempo de cómputo conforme varían los parámetros del problema. Además, se compararán las tres baterías entre sí y se estudiará para qué valores del número de trabajos el problema empieza a obstruirse, en otras palabras, se analizará para qué valores del número de trabajos el tiempo de cómputo es demasiado grande. En base a ello, el análisis del problema es interesante para descubrir entre qué valores del número de trabajos es conveniente emplear un modelo de programación lineal entera mixta y para cuáles es aconsejable utilizar una metaheurística para resolver el problema.

A continuación, el capítulo 6 enuncia las conclusiones obtenidas en este proyecto y expresa las posibles líneas de mejora para futuras investigaciones en este entorno y bajo este problema concreto.

Para concluir, aparecen los dos anexos que sirven de apoyo al documento. En el *Anexo A* se muestra el código elaborado para la creación del MILP, mientras que en el *Anexo B* se exhiben los resultados obtenidos para las tres baterías de problemas.

2 PROGRAMACIÓN DE LA PRODUCCIÓN

2.1 Introducción

La programación de la producción comenzó a mediados del siglo XX, después la Segunda Guerra Mundial, pues se produjo una importante evolución en la investigación operativa. Entre los principales autores, conviene resaltar los trabajos de Johnson (1954), Jackson (1955) y Smith (1956), cuyo principal objetivo era la minimización del número medio de trabajos existentes en el taller (ver Velasco Sánchez, 2013). Años más tarde, Garey y Johnson (1979) formulan la teoría de la complejidad computacional, dado que, para muchos problemas, aún no existía un algoritmo capaz de resolver el problema de manera exacta en un tiempo polinomial acotado (Garey and Johnson, 1979). Por ello, dependiendo del número de recursos empleados para su resolución, los problemas se pueden clasificar en polinomiales (P) y no polinomiales (NP) y, dentro de estos, se encuentran los no polinomiales más complejos (NP-hard), tal y como se verá en el subcapítulo 2.4 con más detalle (Pinedo, 2016).

Gracias al avance computacional, se añadió cierta incertidumbre a los modelos de optimización para investigar su repercusión. De esta manera, se distinguen dos grandes grupos de modelos de optimización en programación de la producción: los deterministas y los estocásticos (Pinedo, 2016). Por un lado, los modelos deterministas son aquellos en los que las mismas entradas darían lugar a las mismas salidas, sin tener presente el azar o incertidumbre. Por otro lado, los modelos estocásticos son aquellos que introducen componentes aleatorios que producen incertidumbre, viéndose alteradas las salidas ante las mismas entradas, tal y como ocurre en la propia naturaleza. En general, los primeros son menos realistas, pero también es más sencillo su estudio, aunque, en muchas ocasiones, ofrecen soluciones que se encuentran cerca de las que se podrían obtener en los modelos estocásticos.

Actualmente, la programación de la producción es un proceso que representa un papel primordial en el sector industrial y que consiste en la toma de un número elevado de decisiones a lo largo del tiempo para asegurarse la entrega al cliente de los productos con la máxima calidad, mínimo coste y menor tiempo de entrega (Pérez González, Paz; Framiñán Torres, José M; Fernández-Viagas, 2014). Por tanto, la finalidad de la programación de la producción es establecer un instante de inicio y fin de cada trabajo en cada una de las máquinas o estaciones necesarias con el objeto de obtener el producto final (Pinedo, 2016). El resultado de este proceso es, idealmente, un programa en el que se especifica qué trabajo se debe introducir en cada recurso, y cuándo (Framiñán, Leisten and Ruiz, 2014). Evidentemente, el programa debe ser factible, es decir, tiene que cumplir con las restricciones del problema de tal forma que la asignación de las operaciones en las distintas máquinas sea válida.

Históricamente, para representar los programas se han utilizado los llamados *Diagramas de Gantt*. Estos consisten en la representación de los trabajos en cada una de las máquinas o viceversa, dependiendo si están orientados a la máquina o al trabajo, frente al tiempo. Ambos casos se encuentran representados en la *Figura 2.1*. A lo largo de todo el documento, siempre haremos referencia a los diagramas orientados al trabajo.

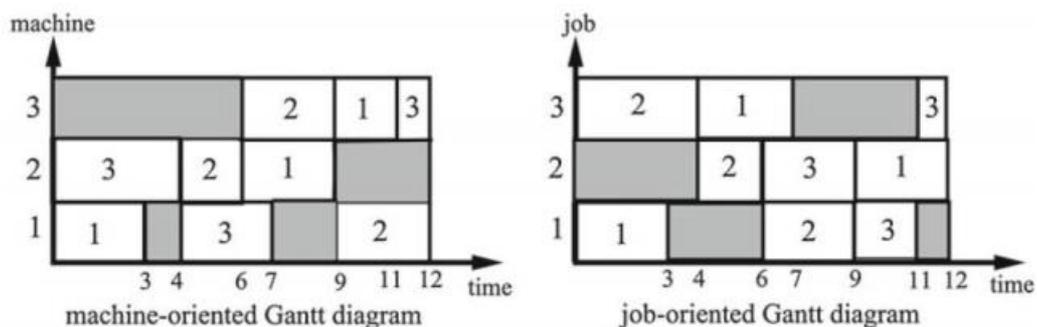


Figura 2.1 Diagramas de Gantt orientados a máquina y trabajo (Framiñán, Leisten and Ruiz, 2014)

Tal y como se mencionó anteriormente, la programación de la producción se basa en la toma de diversas decisiones en el tiempo de manera estructurada y jerárquica a través de diferentes etapas (distribución y fabricación por ejemplo) y frecuencia temporal, corto (operativo), medio (táctico) o largo (estratégico) (Framiñán, José M; Carrasco, 2017)

La programación de la producción es una decisión puramente operativa, ya que en las plantas productivas es fundamental saber el orden de producción para cumplir con la calidad de servicio (Framiñán, José M; Carrasco, 2017). En la *Figura 2.2* se muestran los principales procesos del sistema productivo en base a las etapas de actuación (corto, medio y largo plazo) citadas con anterioridad.

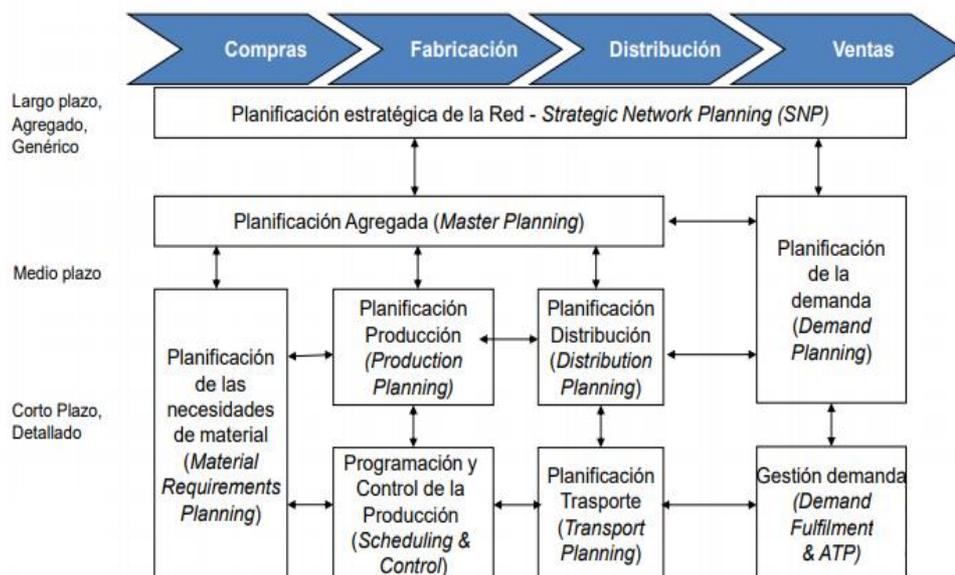


Figura 2.2 Fases del sistema de producción (Pérez González, Paz; Framiñán Torres, José M; Fernández-Viagas, 2014).

Como puede observarse, las decisiones juegan un papel muy relevante en la programación de la producción así que conviene resaltar las siguientes características (Pérez González, Paz; Framiñán Torres, José M; Fernández-Viagas, 2014):

- Deben ser exhaustivas y con un gran nivel de detalle: el volumen de datos que se maneja puede ser muy grande así que es necesario definir el instante de tiempo en que cada trabajo entra en cada recurso de la planta productiva.
- Dado que el tiempo desempeña un papel fundamental en la producción, las decisiones se deben tomar en intervalos cortos de tiempo:
 - ❖ Horizonte rodante o rotatorio: se obtiene un programa para un número de intervalos de tiempo de los cuales solo se ejecuta el primero. Gracias a ello, se consiguen incorporar todas las posibles alteraciones originadas en cada periodo de tiempo.
 - ❖ Reprogramación: aparecen imprevistos que obligan a modificar el programa original dada la estocasticidad del entorno de programación.
- Presenta una gran influencia en los resultados finales.
- La empresa depende íntimamente de las condiciones de decisión, restricciones y objetivos ya que son distintos para cada una.
- Decisiones relativamente estructuradas, tal y como puede apreciarse en la *Figura 2.2*. Además, en la mayoría de ocasiones pueden verse representadas mediante técnicas cuantitativas:
 - ❖ Modelos matemáticos de programación: formados por una serie de restricciones que se deben cumplir, además de la función objetivo, la cual intentaremos satisfacer de manera exacta o aproximada, dependiendo del problema en cuestión. A veces, se tendrán tantas restricciones que será misión imposible alcanzar el óptimo. De hecho, bastará con encontrar una solución admisible (programa factible) al problema.

- ❖ Los datos necesarios para la programación de la producción pueden recogerse en la planta de la fábrica de manera sencilla (tiempos de proceso, fechas de entrega, etc).

2.2 Fundamentación teórica

En este subcapítulo se describirán los principales conceptos que es necesario tener en cuenta cuando se hace referencia a la programación de la producción, así como la notación empleada a la hora de remitirse a cada uno de ellos. Por consiguiente, se van a definir varios subcapítulos para explicar detalladamente toda la base teórica a partir de la cual se desarrollará este presente Trabajo Fin de Grado. Además, se explicarán los modelos de programación de la producción, así como el significado de cada uno de sus campos, la complejidad computacional que estos presentan y los métodos que se utilizan para su resolución.

2.2.1 Conceptos básicos

En los problemas de programación de la producción se supone que el número de trabajos y el número de máquinas con los que se trabaja están acotados o son finitos. Conviene resaltar los siguientes conceptos (Pinedo, 2016):

- **Máquina:** recurso productivo con capacidad para realizar operaciones de transformación o transporte de material. Se trata de la abstracción de diversos objetos reales que pueden representarla (una fábrica, una sección completa, un horno, etc). El número total de trabajos se denota por $M = \{1 \dots m\}$ y se empleará el índice i para hacer referencia a alguna máquina que pertenezca a ese conjunto.
- **Trabajo:** producto que es objeto de operación en alguna de las máquinas. El número total de trabajos se define como $N = \{1 \dots n\}$ y se emplearán los índices j o k para hacer referencia a algún trabajo de ese conjunto.
- **Tiempo de proceso:** duración temporal en la que la máquina i está ocupada en procesar el trabajo j . Se define como p_{ij} .
- **Fecha de entrega:** instante temporal en el que el trabajo j debe estar terminado. Se denota como d_j .
- **Fecha de llegada:** instante de tiempo a partir del cual el trabajo j puede empezar a ser procesado. Se expresa como r_j .
- **Peso/importancia:** prioridad que tiene el trabajo j en nuestro problema. Aparecerán en la función objetivo o criterio que se tenga en cuestión. Se define como w_j .
- **Programa:** contiene la información sobre el instante de inicio y fin de cada trabajo en cada una de las máquinas. A priori, se podría pensar que se tienen dos variables: el tiempo de inicio y el tiempo de fin. Dado que los tiempos de proceso (p_{ij}) son dato, esas dos variables se reducen a una sola. Conocido el instante temporal de inicio y el tiempo de proceso se puede inferir el tiempo de fin y viceversa.

Además, cabe destacar el concepto de programa admisible o programa factible pues se trata de aquel que cumple con las restricciones del proceso productivo. En principio, el objetivo de la programación de la producción es encontrar un programa que cumpla con las restricciones establecidas o programa admisible. El procedimiento a partir del cual se obtiene un programa se denomina algoritmo. Si existe más de un programa admisible, se establecen una serie de objetivos para evaluar cada uno de ellos y discernir sobre cuál es el óptimo. El objetivo de nuestro problema o función objetivo se verá en el subcapítulo 2.2.2 con mayor profundidad.

Llegados a este punto, es necesario realizar una clasificación para conocer cuáles son los tipos de programas que existen en la programación de la producción. De esta manera, si se tienen programas factibles cuyas operaciones no se puedan interrumpir, se distinguen los siguientes programas (Framiñán, Leisten and Ruiz, 2014):

Programa semi-activo: una vez las operaciones se han asignado en todas las máquinas en un orden concreto, no es posible adelantar ninguna operación sin cambiar dicho orden.

- ❖ Programa activo: no se puede adelantar una operación sin retrasar otra.
- ❖ Programa sin retraso: no se mantiene ninguna operación en espera mientras la máquina asignada a esta operación está disponible para procesarla.

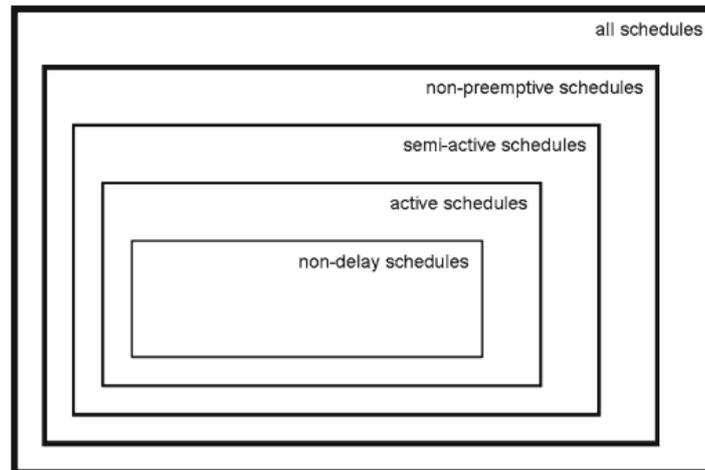


Figura 2.3 Clases de programas no interrumpibles para Job Shop (Framiñán, Leisten and Ruiz, 2014).

- **Ruta:** orden en el que cada trabajo pasa por cada una de las máquinas. Se define R_j como un vector en el que el primer elemento es la primera máquina por la que tendría que pasar el trabajo j y así sucesivamente.
- **Secuencia:** orden en el que cada máquina recibe los trabajos.

2.2.2 Modelos de programación de la producción

Un modelo de programación de la producción trata de abstraer formalmente un problema de programación de la producción (Pinedo, 2016). De esta manera, es necesario definir el número y tipo de máquinas con las que se va a trabajar, el número de trabajos que van a ser procesados, las restricciones que limitan el problema y la función objetivo o criterio que se desea optimizar. Por ello, para recoger esta información se establecen tres campos: α , β , γ . Entonces, conocidos cada uno de los campos, nuestro problema quedaría detallado como sigue: $\alpha | \beta | \gamma$, tal y como puede ser apreciado en la *Figura 2.4*.

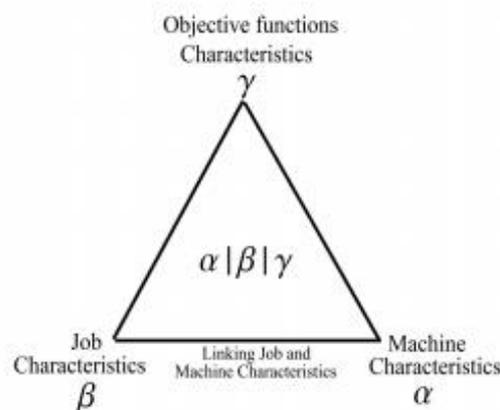


Figura 2.4 Clasificación de los modelos de programación de la producción (Framiñán, Leisten and Ruiz, 2014).

El significado de cada uno de estos campos es el siguiente (Framiñán, Leisten and Ruiz, 2014):

- 1) α : está relacionado con la característica propia de la máquina. Se trata del tipo de entorno a considerar. En base al valor de α se tiene la siguiente clasificación:
- ❖ $\alpha = 1$: Cada trabajo debe procesarse en esa máquina en una sola operación. Además, los trabajos no siguen una ruta concreta.

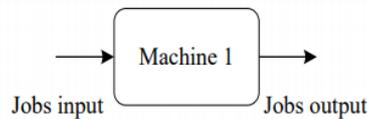


Figura 2.5 Máquina única (Framiñán, Leisten and Ruiz, 2014).

- ❖ $\alpha = P$: máquinas paralelas idénticas. Replican el entorno de una máquina para el incremento de la producción. Todas las máquinas trabajan a la misma velocidad así que el tiempo de proceso de cada trabajo no depende de la máquina en la que se procese. Los trabajos no siguen una ruta concreta. Para definir el número de máquinas que existen, se expresa como sigue: $\alpha = Pm$, donde m es un número natural.

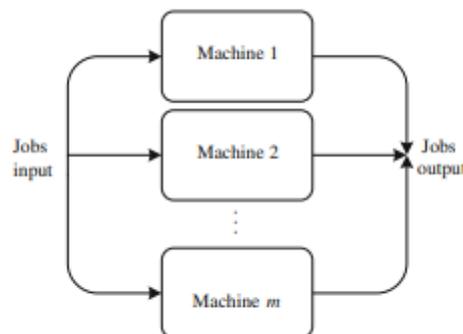


Figura 2.6 Máquinas paralelas idénticas (Framiñán, Leisten and Ruiz, 2014).

- ❖ $\alpha = Q$: máquinas paralelas uniformes. Para definir el número de máquinas que existen, se suele expresar como sigue: $\alpha = Qm$, donde m es un número natural. Cada trabajo debe procesarse en una máquina en una sola operación. Los trabajos no siguen una ruta concreta. Cada máquina tiene diferentes velocidades de procesado (v_i), así que habrá algunas más rápidas y otras que vayan más despacio. el tiempo de procesado (p_{ij}) de cada trabajo en cada máquina se calcula como: $p_{ij} = \frac{p_j}{v_i}$.
- ❖ $\alpha = R$: máquinas paralelas no relacionadas. Las máquinas son diferentes entre sí, no existe una relación de velocidad entre ellas. Por tanto, cada trabajo tendrá un tiempo de proceso distinto según en la máquina en la que se procese. Para definir el número de máquinas que existen, se expresa como sigue: $\alpha = Rm$, donde m es un número natural.
- ❖ $\alpha = F$: taller de flujo (*Flow Shop*). Las máquinas están dispuestas en serie en este entorno, de tal forma que todos los trabajos pasarán por cada una de estas siguiendo una ruta constante. El vector de ruta (R) es idéntico para todos los trabajos. Para cada trabajo, su procesamiento en la máquina m no puede comenzar si en la $m-1$ aún no ha concluido. Además, existe un caso particular denominado taller de flujo regular de permutación en el que la secuencia en cada máquina no varía. Para definir el número de máquinas que hay, se expresa de la siguiente manera: $\alpha = Fm$, donde m es un número natural.

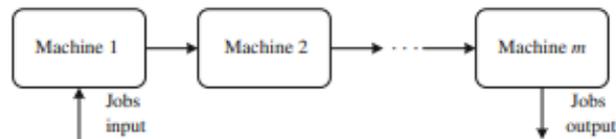


Figura 2.7 Flow Shop (Framiñán, Leisten and Ruiz, 2014).

- ❖ $\alpha = J$: taller trabajos (*Job Shop*). Las máquinas están dispuestas en serie en este entorno de manera tal que los trabajos pasarán por cada una de estas máquinas siguiendo una ruta que va a ir cambiando para cada trabajo. De este modo, el vector de ruta (R_j) varía para cada trabajo, dando lugar a una matriz. El número de máquinas se expresa de la siguiente manera: $\alpha = Jm$; donde m es un número natural.

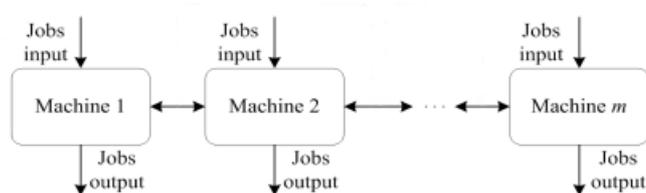


Figura 2.8 Job Shop (Framiñán, Leisten and Ruiz, 2014).

- ❖ $\alpha = O$: taller abierto (*Open Shop*). Las máquinas están dispuestas en serie, cada una realiza un proceso diferente, así que es necesario que cada trabajo sea procesado en todas las máquinas. La ruta de los trabajos no está determinada, de hecho, es lo que se busca en este tipo de entorno. El número de máquinas se expresa de la siguiente manera: $\alpha = Om$; donde m es un número natural.

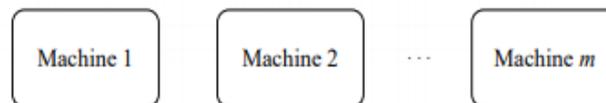


Figura 2.9 Open Shop (Framiñán, Leisten and Ruiz, 2014).

- ❖ $\alpha = HFm$: entorno híbrido (*Hybrid layout*). Se trata de un taller formado de tantas etapas como indique el valor de m , que será un valor entero. Cada etapa está formada por una única máquina o varias máquinas en paralelo (idénticas, relacionadas o uniformes).

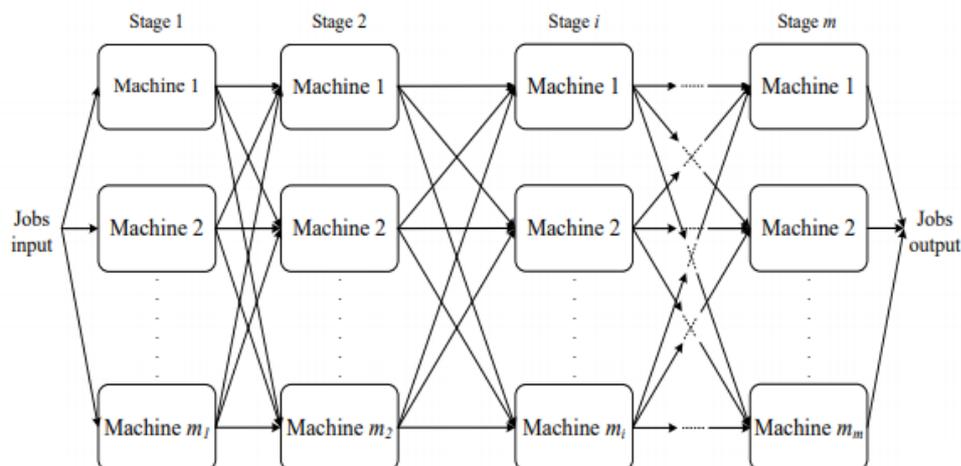


Figura 2.10 Hybrid layout (Framiñán, Leisten and Ruiz, 2014).

- 2) β : hace referencia a las restricciones o limitaciones a las que está sujeto el problema de programación. Las principales restricciones empleadas en los problemas son (Framiñán, Leisten and Ruiz, 2014):
- ❖ $\beta = r_j$: marca el instante temporal en el que el trabajo j puede empezar a ser procesado.
 - ❖ $\beta = d_j$: señala el instante de tiempo en el que el trabajo j debe ser entregado. Esto es propio de aquellos problemas en los que todos los trabajos presentan la misma fecha de entrega.
 - ❖ $\beta = s_{ij}$: marca la obligatoriedad de realizar una puesta a punto de la máquina i antes de procesar el trabajo j .
 - ❖ $\beta = s_{ijk}$: expresa la necesidad de realizar una puesta en marcha en la máquina i para procesar el trabajo j antes del k .
 - ❖ $\beta = batch$: indica que las máquinas pueden procesar lotes de b trabajos a lo sumo.
 - ❖ $\beta = prec$: marca que un trabajo no puede realizarse hasta que acabe el que le precede.
 - ❖ $\beta = prmu$: se trata de una restricción que se emplea en los entornos *Flow Shop* para indicar que el orden de los trabajos en cada una de las máquinas es el mismo.
 - ❖ $\beta = no - idle$: no están permitidos tiempos ociosos de las máquinas entre trabajos. Una vez arrancada la máquina, su funcionamiento no puede parar.
 - ❖ $\beta = no - wait$: los trabajos no pueden esperar entre máquinas.
 - ❖ $\beta = pmtn - non - resumable$: se pierde el trabajo hecho de la tarea que se interrumpe y se comienza desde cero cuando se reinicia.
 - ❖ $\beta = pmtn - semi - resumable$: se pierde parte del trabajo realizado.
 - ❖ $\beta = pmtn - resumable$: se reinicia el trabajo por donde se había dejado tras la interrupción.
- 3) γ : señala la función objetivo o criterio. El objetivo es minimizar siempre una función que va a depender de los tiempos de finalización de los trabajos (C_j) y que, de manera directa, depende del tiempo (Pinedo, 2016). A continuación, se van a describir las funciones objetivo existentes en la programación:
- ❖ C_j : instante de finalización del trabajo j .
 - ❖ F_j : tiempo de flujo del trabajo j (*flowtime*). Se trata del tiempo total que está ese trabajo en el entorno.
 - ❖ L_j : retraso del trabajo j (*lateness*). Mide en unidades temporales la demora de ese trabajo.
 - ❖ T_j : tardanza del trabajo j (*tardiness*). Indica el retraso de un trabajo.

- ❖ E_j : adelanto del trabajo j (*earliness*). Indica las unidades temporales que un trabajo se adelanta con respecto a su fecha de entrega.
- ❖ U_j : trabajo tardío (*tardy job*).
- ❖ V_j : trabajo temprano (*early job*).

Tabla 2.1 Cálculo de las principales funciones objetivo (Framiñán, Leisten and Ruiz, 2014).

Medida	Notación	Expresión
<i>Flowtime</i>	F_j	$C_j - r_j$
<i>Lateness</i>	L_j	$C_j - d_j$
<i>Tardiness</i>	T_j	$\text{máx} \{0, L_j\}$
<i>Earliness</i>	E_j	$\text{máx} \{0, -L_j\}$
<i>Tardy job</i>	U_j	$\left\{ \begin{array}{l} 1 \text{ si } L_j > 0 \\ 0 \text{ en otro caso} \end{array} \right.$
<i>Early job</i>	V_j	$\left\{ \begin{array}{l} 1 \text{ si } L_j < 0 \\ 0 \text{ en otro caso} \end{array} \right.$

Los objetivos que pueden aparecer en un modelo de programación de la producción se agrupan en cinco grandes grupos (Framiñán, Leisten and Ruiz, 2014):

- $\gamma = -$: se trata de encontrar un problema factible.
- Incluye todos aquellos objetivos relacionados con la reprogramación.
- Contiene todos aquellos criterios que no se encuentran relacionados con las fechas de entrega:
 - ❖ *Makespan*: $C_{\text{máx}} = \text{máx}_{1 \leq j \leq n} C_j$.
 - ❖ *Maximum Flowtime*: $\text{máx} F_j = \text{máx}_{1 \leq j \leq n} C_j$.
 - ❖ *Total Completion Time*: $\sum C_j = \sum_{j=1}^n C_j$.
 - ❖ *Total Flowtime*: $\sum F_j = \sum_{j=1}^n F_j$.
- Presenta criterios relacionados con las fechas de entrega:
 - ❖ *Maximum Lateness*: $\text{máx} L_j = \text{máx}_{1 \leq j \leq n} L_j$.
 - ❖ *Maximum Tardiness*: $\text{máx} T_j = \text{máx}_{1 \leq j \leq n} T_j$.
 - ❖ *Maximum Earliness*: $\text{máx} E_j = \text{máx}_{1 \leq j \leq n} E_j$.
 - ❖ *Total Lateness*: $\sum L_j = \sum_{j=1}^n L_j$.
 - ❖ *Total Tardiness*: $\sum T_j = \sum_{j=1}^n T_j$.
 - ❖ *Total Earliness*: $\sum E_j = \sum_{j=1}^n E_j$.
 - ❖ *Number of tardy jobs*: $\sum U_j = \sum_{j=1}^n U_j$.
- El último grupo contiene las expresiones vistas anteriormente multiplicadas por su peso o importancia dentro del criterio, w_j .

2.2.3 Complejidad computacional

Como ya se mencionó en la sección 2.4, Garey y Johnson introdujeron la Teoría de la Complejidad Computacional en 1979, la cual se basa en un enfoque no funcional, en donde interesa cómo se lleva a cabo el cálculo y su complejidad en función de los recursos que utiliza (ver Cortez, 2004). Es por ello, que los problemas existentes en la programación de la producción se clasifican en dos grandes grupos:

- Problemas polinomiales (*P*). Se dice que un problema es de este tipo si cumple las siguientes características (ver Flores, 2014):

1. Se trata de un problema de decisión.
2. Se tiene un tipo de problema resoluble en un tiempo polinomial.

El tiempo polinomial da la noción de que el problema en cuestión puede resolverse con un algoritmo que corra en un tiempo relativamente rápido. Formalmente, el que un algoritmo D que depende del tamaño de la entrada n corra en un tiempo polinomial significa que: $D = O(n^k)$. En este caso, se dice que el algoritmo D corre en un tiempo polinomial. Además, se dirá que un algoritmo que corre en un tiempo polinomial es eficiente, pero evidentemente, algún problema acotado por un polinomio de grado muy elevado (10000 por ejemplo) difícilmente se le puede catalogar como rápido o eficiente. Por ende, los problemas que se resuelven en un tiempo polinomial lo hacen con un grado relativamente pequeño.

- Problemas no polinomiales (*NP*). Se considera que un problema es de este tipo si cumple las siguientes características (ver Flores, 2014):

1. Se trata de un problema de decisión.
2. Se tiene un problema verificable en tiempo polinomial.

En cuanto a verificar, nótese que hace referencia a comprobar si el certificado emitido por el algoritmo cumple con los requerimientos del problema. Un problema pertenece a la clase *NP* si el algoritmo usado para verificar su certificado corre en tiempo polinomial.

Además, dentro de este campo se encuentran los problemas difíciles (*NP-hard*). La mayoría de los problemas existentes en programación de la producción son del tipo *NP-hard*, lo que significa que es poco probable encontrar una solución óptima para una instancia de gran tamaño en un tiempo razonable o tiempo polinomial. El tipo de problema que se describirá en la sección 3 en este presente Trabajo Fin de Grado pertenece a este subconjunto (*NP-hard*), pues es un tipo de problema para el cual no es posible encontrar una solución óptima para todas sus instancias en un tiempo polinomial.

A modo de esquema visual, se ilustra lo anteriormente expuesto en la *Figura 2.11* que aparece en la siguiente página.

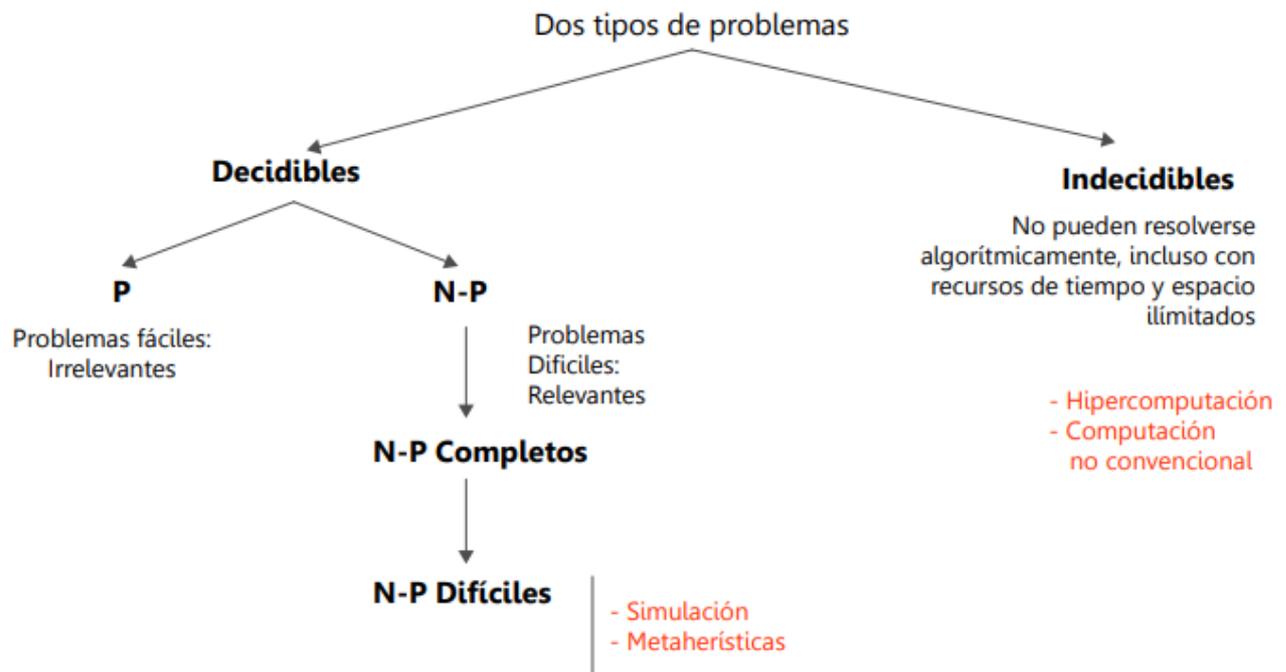


Figura 2.11 Tipología de problemas de programación de la producción (Maldonado, 2013).

2.2.4 Métodos de programación de la producción

El objetivo principal de los métodos existentes es resolver un problema de programación de la producción encontrado soluciones admisibles. Dichos métodos son aplicables a cualquier instancia de un modelo y la solución obtenida debe proporcionar unos buenos resultados teniendo en cuenta un objetivo (Framiñán, Leisten and Ruiz, 2014). Aquellos métodos que proporcionan una solución se denominan algoritmos y estos pueden ser de dos tipos:

- Exactos: proporcionan una solución óptima a un problema dado. A este conjunto pertenecen los algoritmos constructivos exactos (reglas de despacho y los algoritmos de *Johnson*, *Lawler* y *Moore*) y los algoritmos enumerativos, los cuales garantizan la evaluación de todas las soluciones del modelo. (programación matemática, *Branch and Bound* y programación dinámica). Estos últimos son no polinomiales así que solo son capaces de encontrar el óptimo para instancias pequeñas.
 1. Reglas básicas de despacho: cada regla resuelve de forma óptima un problema en concreto.
 2. Algoritmo de *Johnson*: resuelve de forma óptima el problema $F2 \parallel C_{\max}$.
 3. Algoritmo de *Lawler*: para resolver el problema $1 \mid \text{prec} \mid \max g(C_j)$.
 4. Algoritmo de *Moore*: resuelve $1 \mid \mid \sum U_j$.
 5. Programación matemática: la formulación se realiza mediante la programación lineal entera mixta (MILP) y se resuelve mediante un Solver (como puede ser *Gurobi*).
 6. *Branch and bound*: empleado para resolver problemas de optimización combinatoria. Po resto, es necesario definir los nodos, la ramificación, realizar el cálculo de la cota e ingeniar una estrategia de ramificación.

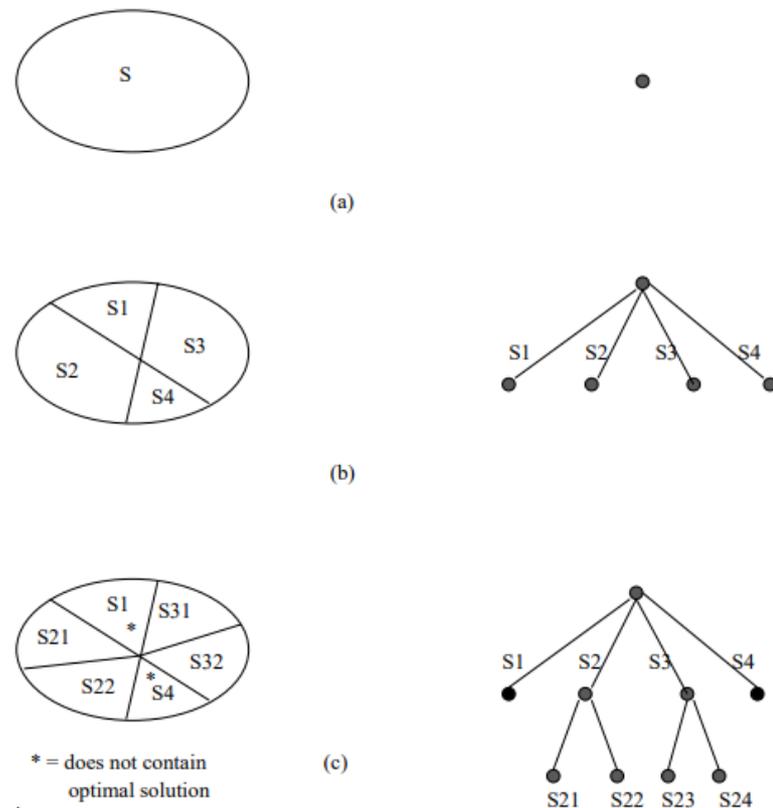


Figura 2.12 Espacio de búsqueda de un Branch and Bound (Clausen, 1999).

- Aproximados: generan una solución admisible pero no garantizan optimalidad. En esta tipología podemos encontrar las heurísticas constructivas, las heurísticas de mejora y las metaheurísticas. Algunas de las metaheurísticas más utilizadas son: el Método de Búsqueda Descendente Simple, *Simulated Annealing*, Algoritmo Genético e *Iterated Greedy*.

2.2.5 Programación multiobjetivo

Es muy común encontrarse en la literatura problemas de secuenciación en los que es necesario programar más de un entorno de trabajo. A este tipo de problemas se les conoce como problemas multiobjetivo. La idea es ordenar los trabajos de todos los conjuntos de tal forma que se satisfaga el objetivo individual de cada entorno, teniendo en cuenta que compiten por los mismos recursos, es decir, las distintas máquinas por las que deben ser secuenciados (Framiñán, Leisten and Ruiz, 2014). Debido a la gran dificultad que presenta este tipo de problemas, la mayoría se basan en dos conjuntos de trabajos (n_A , n_B) y una o dos máquinas.

Cuando se tienen dos o más conjuntos de trabajos, hay que definir una función objetivo para cada uno de ellos ya que, usualmente, tendrán objetivos distintos que optimizar (pese a que pueden ser los mismos para cada entorno). Para dos conjuntos de trabajos n_A , n_B , cada uno con un objetivo distinto O_A , O_B , se consideran los siguientes casos (Framiñán, Leisten and Ruiz, 2014):

- ❖ Método de Pareto: se enfrentan ambos objetivos en una curva denominada *Frente de Pareto*, en la que las mejoras de un objetivo suponen la disminución del objetivo del otro entorno. La idea principal es alcanzar el equilibrio entre ambos criterios. Se expresa como $\#(O_A, O_B)$. Véase la Figura 2.13 para los objetivos $\sum T_j$ y $C_{máx}$.

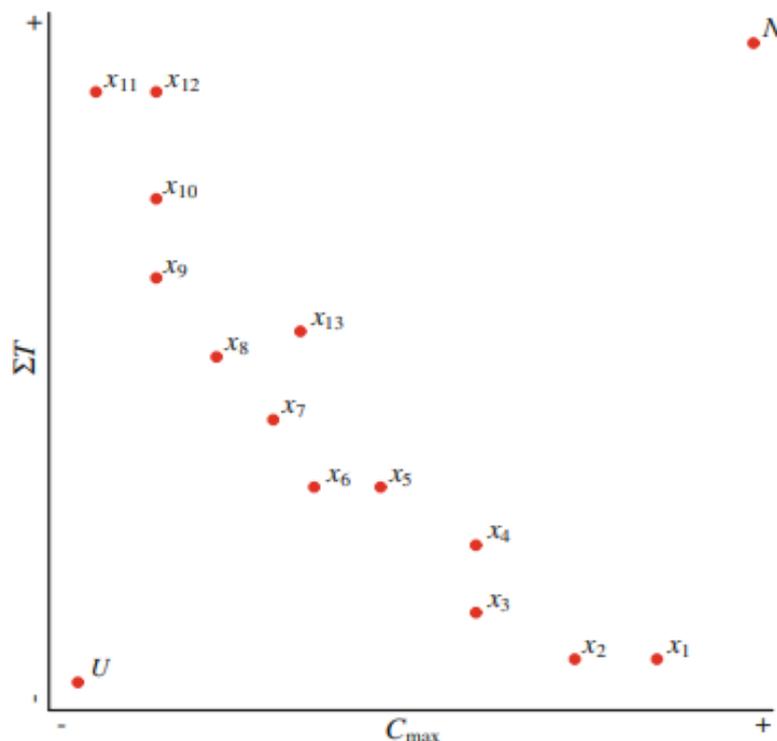


Figura 2.13 Frente de Pareto (Framiñán, Leisten and Ruiz, 2014).

- ❖ Método de Combinación Lineal Convexa: se considera un único objetivo principal compuesto por los dos criterios, ponderado con un parámetro $0 \leq \lambda \leq 1$. Se expresa como sigue: $Min \lambda O_A + (1 - \lambda) O_B$.
- ❖ Método de Restricción Épsilon: se busca el mínimo del objetivo O_A sujeto a que el objetivo O_B no puede exceder del valor de épsilon. Se denota como $\varepsilon(O_A, O_B)$ o también como $Min O_A: O_B \leq \varepsilon$. Es importante analizar cómo cambia el problema en función de los valores de ε . Considerar un valor muy elevado supone tener en cuenta exclusivamente O_A , y para un valor demasiado pequeño, supone la infactibilidad del problema.

En la secuenciación de problemas multicriterio, históricamente se han estudiado tanto problemas conjuntos como disjuntos. Estos últimos son más estudiados en la actualidad, así que este presente Trabajo Fin de Grado se centrará en aquellos conjuntos de trabajos que no presenten ningún elemento en común.

3 DESCRIPCIÓN DEL PROBLEMA

En este capítulo se describirá detenidamente el problema abordado en este documento, 3.1, se manifestará lo que significa un MILP, 3.2, se modelará matemáticamente el problema, 3.3, se explicará la generación de los parámetros del problema, 3.4, y se ilustrará un ejemplo que ayude a su comprensión, 3.5.

3.1 Descripción del problema

En este documento se va a desarrollar un problema de programación de la producción multicriterio en un entorno tipo taller de flujo o *Flow Shop*, con dos máquinas. Como ya se mencionó en la sección 2.2.4, el entorno de programación de tipo taller de flujo es aquel en el que las máquinas están dispuestas en serie y los trabajos se procesan en cada una de ellas con un vector de ruta constante. En este caso, al tratarse de dos máquinas, el vector de ruta es $R = [1\ 2]$, invariable, ya que todos los trabajos se procesan primero en la máquina 1 y, posteriormente, en máquina 2. De esta manera, el primer campo queda definido como $\alpha = F2$.

Además, se considera la restricción de permutación, la cual suele ser una restricción intrínseca a este tipo de entorno. Por ello, el siguiente campo se muestra como $\beta = \text{prmu}$. Esto quiere decir que el orden de los trabajos en cada una de las máquinas es el mismo. Esta restricción disminuye de $(N!)^m$ a $N!$ programas factibles a evaluar, reduciendo fuertemente el campo de soluciones. Pese a ello, el número de programas admisibles crece de manera factorial conforme aumenta el valor de n . En la *Figura 3.1* se muestra la evolución del número de programas admisibles conforme varía el número de trabajos.

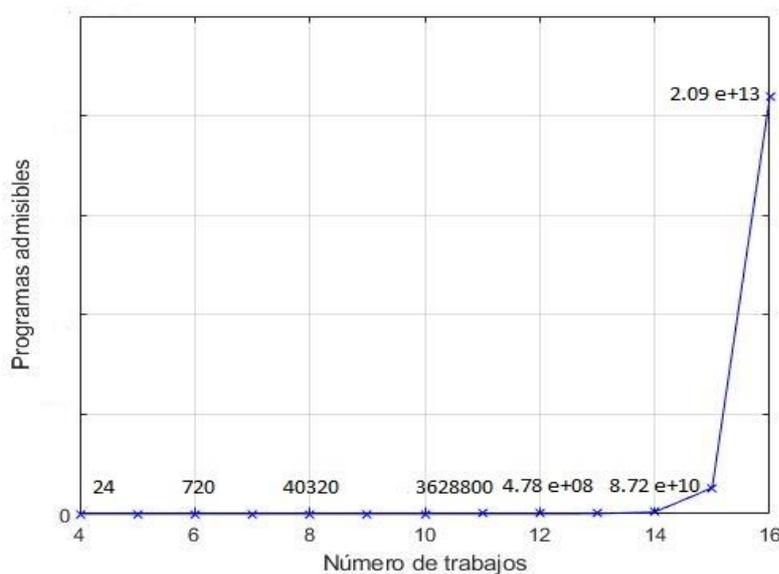


Figura 3.1 Evolución del número de programas admisibles.

Finalmente, dado que el problema que se analiza en este Proyecto es multiobjetivo, se definen dos conjuntos de trabajos n_A y n_B , los cuales hacen referencia al entorno de trabajo A y al entorno de trabajo B, respectivamente. Tal y como se comentó en la sección 3.1, se aborda un problema basado en un entorno de trabajos disjunto, lo que significa que todos los trabajos de un determinado conjunto pertenecen exclusivamente a ese conjunto, siendo la intersección de ambos entornos el vacío. Por consiguiente, se define un conjunto total de trabajos $N = \{1 \dots n\}$, el cual se divide en dos, de forma que se cumple que $N = n_A + n_B$. En base a lo anterior, se van a considerar dos casos:

- Para un N par: $n_A = \{1 \dots \frac{N}{2}\}$ y $n_B = \{\frac{N}{2} + 1 \dots N\}$.
- Para un N impar: se define $n_A = \{1 \dots \frac{N+1}{2}\}$ y $n_B = \{\frac{N+1}{2} + 1 \dots N\}$.

Bajo estos dos entornos, se va a trabajar con $\sum T_j^A$ como función objetivo de A y $C_{máx}^B$ como función objetivo de B. Esto implica que el problema a optimizar se trata de $Min \sum T_j^A : C_{máx}^B \leq Q$. Por ende, el ultimo campo queda como $\gamma = \sum T_j^A : C_{máx}^B \leq Q$. Nótese que Q hace referencia al valor ε mencionado en el subcapítulo (3.1), simplemente renombrado de forma distinta, lo que significa que el problema multicriterio se resolverá adoptando el Método de Restricción Épsilon, en el que Q se irá apretando y aflojando para ver cómo varía el *makespan*.

En base a todo lo anterior, el problema que se va a abordar en este documento queda totalmente definido como:

$$F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$$

El objetivo es encontrar un programa óptimo que permita obtener el menor valor del *total tardiness* del conjunto A, sujeto a que el *makespan* de B no puede exceder el valor de Q previamente establecido. Dado que se trata de un problema del tipo *NP-hard*, la idea es resolverlo con un modelo matemático de programación lineal, un *Branch and Bound* o una metaheurística. Cabe pensar que seleccionar un algoritmo *Branch and Bound* para resolver este tipo de problemas es una buena idea. Sin embargo, un algoritmo *Branch and Bound* solo está capacitado para resolver problemas de instancias de hasta 16 trabajos de tamaño (Ahmadi-Darani, Moslehi and Reisi-Nafchi, 2018). Ya que la eficiencia del *Branch and Bound* es limitada y excasa, se selecciona un modelo matemático de programación lineal entera mixta para resolver este tipo de problemas en un tiempo razonable.

3.2 Modelo matemático de programación lineal entera mixta

Un amplio abanico de problemas de programación de la producción presenta tanto variables continuas como variables enteras, en concreto suelen ser variables binarias (toman valores 0,1). Como resultado, gran parte de ellos se pueden representar como un modelo de programación lineal entera mixta, con una función objetivo a optimizar que se encuentra sujeta a una serie de restricciones. Estos modelos matemáticos son conocidos como problemas lineales enteros mixtos (MILP) o *Mixed Integer Linear Problems* en inglés (García, 2010).

Un MILP puede representarse en el caso más general como:

$$\begin{aligned} \min \quad & c^T x + d^T y \\ \text{s. a:} \quad & Ax + By \leq b \\ & x \geq 0 \\ & y \in \{0,1\} \end{aligned}$$

donde:

x es un vector de n variables continuas.

y es un vector de q variables binarias.

c, d son vectores ($n \times 1$) y ($q \times 1$) de parámetros.

A, B son matrices.

b es un vector de p componentes.

La formulación anterior tiene una función objetivo lineal, así como las restricciones en x e y . Remarcar que, si la matriz A y el vector c son ambos cero, entonces tenemos un problema de programación entera puro.

Las técnicas actuales más habituales para resolver este tipo de problemas son (García, 2010):

- 1) *Branch and Bound* (Ramificación y acotamiento)
- 2) Descomposición de Benders
- 3) Métodos de plano de corte
- 4) Métodos basados en la lógica.

En los métodos de ramificación y acotamiento, se emplea un árbol de combinaciones binario para representar las combinaciones 0-1 de las variables binarias. La región se divide de forma sistemática en subdominios y se derivan límites superior e inferior válidos en diferentes niveles del árbol. Por su parte, en los métodos de plano de corte la región factible no se divide en subdominios, pero en su lugar se añaden nuevas restricciones o “cortes” que reducen la región factible hasta que se obtiene la solución óptima. Posteriormente, en los métodos de descomposición se explota la estructura de los modelos a través de división de las variables, dualidad y métodos de relajación. Finalmente, en los métodos lógicos, se explotan restricciones disyuntivas o inferencia simbólica.

3.3 Modelado matemático del problema

El objetivo es generar un modelo matemático de programación lineal entera mixta (MILP) que recoja fielmente el problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ y permita resolverlo de manera exacta. Por este motivo, se presenta un MILP basado en la posición de los trabajos dentro de la secuencia capaz de alcanzar el óptimo del problema de manera eficiente (Kharbeche and Haouari, 2012). Entre muchos otros, el modelo matemático basado en los tiempos de proceso de los trabajos muestra la mejor puesta en escena para su resolución. De esta manera, el problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ también es conocido como modelo ACT (*Agent's jobs Completion Times*). Este modelo muestra la minimización de la suma de las tardanzas del entorno A en la función objetivo principal, mientras que el criterio que señala que el *makespan* del entorno B no puede exceder de un determinado valor se introduce como una restricción más dentro del problema. Gracias a esto, se contemplan los objetivos de ambos entornos. Como resultado, el modelo matemático requiere de las siguientes variables:

- T_k : tardanza del trabajo programado en la posición k de la secuencia.
- F_k : tiempo de terminación en la máquina 1 del trabajo programado en la posición k de la secuencia.
- C_k : tiempo de terminación en la máquina 2 del trabajo programado en la posición k de la secuencia.
- x_{kj} : variable binaria que toma el valor 1 si el trabajo j está asignado a la posición k de la secuencia y cero en otro caso.

Por su parte, los datos de entrada que necesita el modelo son:

- a_j : vector que contiene los tiempos de proceso de cada trabajo j en la máquina 1.
- b_j : vector que contiene los tiempos de proceso de cada trabajo j en la máquina 2.
- d_j : vector que presenta las fechas de entrega de cada trabajo j .
- M : constante de valor muy elevado. Se ha considerado $M = 1000000$.
- N : número de trabajos total que presenta la instancia.
- Q : parámetro que limita el valor del *makespan* de B.

En base a todo lo anterior, modelo matemático es el que se muestra a continuación (Ahmadi-Darani, Moslehi and Reisi-Nafchi, 2018):

$$\min \sum_{k=1}^N T_k \quad (1)$$

$$s. a: \quad \sum_{k=1}^N x_{kj} = 1 \quad j = 1 \dots N \quad (2)$$

$$\sum_{j=1}^N x_{kj} = 1 \quad k = 1 \dots N \quad (3)$$

$$F_1 = \sum_{j=1}^N a_j x_{1j} \quad (4)$$

$$F_k - F_{k-1} = \sum_{j=1}^N a_j x_{kj} \quad k = 2 \dots N \quad (5)$$

$$C_1 = \sum_{j=1}^N (a_j + b_j) x_{1j} \quad (6)$$

$$C_k \geq C_{k-1} + \sum_{j=1}^N b_j x_{kj} \quad k = 2 \dots N \quad (7)$$

$$C_k \geq F_{k-1} + \sum_{j=1}^N (a_j + b_j) x_{kj} \quad k = 2 \dots N \quad (8)$$

$$T_k \geq C_k - \sum_{j \in n_A} (d_j - M) x_{kj} - M \quad k = 1 \dots N \quad (9)$$

$$C_k \leq Q + M (1 - \sum_{j \in n_B} x_{kj}) \quad k = 1 \dots N \quad (10)$$

$$F_k, T_k, C_k \geq 0 \quad k = 1 \dots N \quad (11)$$

$$x_{kj} \in \{0,1\} \quad i, k = 1 \dots N \quad (12)$$

En el modelo, la función objetivo (1) minimiza la tardanza total o *total tardiness* de los trabajos que están asignados en las distintas posiciones k de la secuencia. Dado que la tardanza se calcula en este modelo únicamente para los trabajos del primer entorno o cliente, la ecuación es igual a minimizar el *total tardiness* de los trabajos del primer agente. La ecuación (2) indica que cada trabajo j solo se puede asignar a una posición k de la secuencia, mientras que la (3) señala que en cada posición k solo se puede asignar un trabajo j . De las restricciones (4) a la (8) determinan el tiempo de terminación de los trabajos en las diferentes posiciones de la secuencia en la primera máquina y en la segunda. La restricción (9) enuncia que la tardanza en la posición k se calcula si su trabajo programado pertenece al primer cliente. Muy similar a esto, la restricción (10) manifiesta que el tiempo de terminación de una determinada posición k de la secuencia está limitado por el valor Q si el trabajo del primer agente está asignado en esa posición. La restricción (11) indica que las variables F_k, T_k, C_k del problema deben ser positivas o nulas. Finalmente, la restricción (12) establece que las variables x_{kj} han de ser enteras, en concreto, binarias.

El número de trabajos de la instancia en cuestión, los vectores a_j y b_j y el número de máquinas, que es una constante de valor igual a 2, serán leídos de un fichero de texto, tal y como se verá en la sección 4.4. Por su parte, el valor de M siempre será el mismo al tratarse de una constante y se pasará como dato en el propio código del programa, que como se mostrará en el capítulo 4.2, se tratará de la herramienta Matlab. De esta manera, quedarían por conocer tanto el valor de Q como el valor del vector d_j , los cuales serán establecidos dentro del código del programa.

3.4 Generación de los parámetros del modelo

3.4.1 Generación del valor de Q

Dado que el *makespan* del cliente B está limitado al valor de Q , conviene resaltar la vital importancia que tiene la creación de un valor de Q correcto para la obtención de una solución óptima. El cuidado radica en lograr el valor de Q exacto que nos permita resolver adecuadamente el problema, pues se pueden dar los siguientes escenarios:

- Si el valor de Q es demasiado pequeño, no se satisface la restricción $C_{máx}^B \leq Q$, por lo que el problema resulta infactible.
- Si el valor de Q es excesivamente grande, el problema se limitaría a encontrar el mínimo valor de la suma de las tardanzas del cliente A, sin tener en consideración al cliente B.

De esta forma, es imprescindible encontrar un valor razonable de Q que se ajuste a las adecuaciones de este problema. Para encontrar este valor de Q se va a emplear la regla de Johnson. Por ello, es necesario conocer el funcionamiento de dicha regla.

El algoritmo de Johnson es un método exacto de programación de la producción que sirve para encontrar una solución óptima al problema $F2 \mid \mid C_{máx}$. La idea básica de resolución se recoge en el pseudocódigo mostrado en la Figura 3.2 que aparece a continuación.

Input: instance data

Output: Optimal sequence Π

begin

Let $\Pi = \emptyset, J = \{1, \dots, n\}$;

Let $J_1 = \{j \in J / p_{1,j} \leq p_{2,j}\}$;

Let $J_2 = \{j \in J / p_{1,j} > p_{2,j}\}$;

Construct Π_1 sorting jobs in J_1 by increasing values of $p_{1,j}$;

Construct Π_2 sorting jobs in J_2 by decreasing values of $p_{2,j}$;

$\Pi = (\Pi_1 \mid \Pi_2)$;

return Π

Figura 3.2 Pseudocódigo de la regla de Johnson (Framiñán, Leisten and Ruiz, 2014).

En base a ello, el método de Johnson consta de los siguientes pasos:

1. Se establece una secuencia π inicialmente vacía de tamaño n .
2. Se introducen en J_1 todos aquellos trabajos cuyo tiempo de proceso en la máquina 1 es menor o igual al de la máquina 2 para cada trabajo.
3. Se introducen en J_2 todos aquellos trabajos cuyo tiempo de proceso en la máquina 1 sea mayor al de la máquina 2 para cada trabajo.
4. Se ordenan los trabajos de J_1 de menor a mayor y se construye la secuencia π_1 .
5. Se ordenan los trabajos de J_2 de mayor a menor y se construye la secuencia π_2 .

6. Finalmente se crea la secuencia π que es la unión de π_1 y π_2 , en este orden. Esta es la secuencia óptima del problema.

Dado que el problema que se plantea también presenta dos máquinas en un taller de flujo y el objetivo del segundo cliente es minimizar el máximo tiempo de terminación, para obtener el valor de Q , se ordenan los trabajos que pertenecen al entorno A según dicta la regla de Johnson., obteniendo un vector J_A ordenado. Luego se ordenan los trabajos que pertenecen al entorno B según Johnson también, originando el vector J_B ordenado. Seguidamente, se obtiene una secuencia resultante de la unión de ambos vectores ordenados según Johnson denominada π , es decir, se alcanza la secuencia $\pi = [J_A \cup J_B]$. Seguidamente, se calcula el *makespan* para esa secuencia que determina el valor de Q , obteniendo el valor $C_{máx,\pi}^B$. Como el principal objetivo de este Proyecto es analizar cómo evolucionan el *total tardiness* del cliente A y el valor del *makespan* del entorno B, se va a alterar el valor de Q multiplicándolo por un parámetro que permita reflejar dichas variaciones. Por este motivo, se introduce el parámetro ϵ , que será un vector $\epsilon = [-0.25, 0, 0.25, 0.5, 0.75]$ que permite ajustar el valor de Q . En base a todo ello, el valor de Q se calcula como estipula la ecuación (13):

$$Q = C_{máx,\pi}^B (1 + \epsilon) \quad (13)$$

Como resultado, se obtienen siete valores de Q que serán un dato de entrada al modelo y que permitirán analizar el ajuste de $C_{máx}^B$ a Q .

3.4.2 Generación de los valores de d_j

Las fechas de entrega de los trabajos son otro dato que es necesario generar dentro del código del programa. Conviene destacar que generar valores muy pequeños dará lugar a que todos los trabajos programados en la secuencia vayan tarde, y que establecer valores muy elevados conllevará a que haya mucha diferencia con respecto a los tiempos de terminación en la máquina 2 y que, por tanto, se obtengan valores siempre cerca de cero. Por esto, es necesario fijar unos valores de las fechas de entrega que se asemejen a valores reales que es posible encontrar en las plantas de producción. En consecuencia, las fechas de entrega siguen una distribución discreta uniforme en el intervalo $[C_{máx,\pi}^B (1 - \tau - R/2), C_{máx,\pi}^B (1 - \tau + R/2)]$, donde R es una medida de dispersión, τ es un factor de tardanza y $C_{máx,\pi}^B$ es el tiempo de terminación en la máquina 2 de la secuencia (Ahmadi-Darani, Moslehi and Reisi-Nafchi, 2018). Se ha considerado $R = 0.5$ y $\tau = 0.75$ para generar unos d_j adecuados, ni excesivamente grandes, ni demasiado pequeños.

3.5 Ejemplo numérico

En este apartado se procede al análisis de un ejemplo concreto del problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$, con dos máquinas (que será constante en todo el Proyecto), dos conjuntos de trabajos (n_A y n_B), 3 trabajos ($N = 3$), $M = 1000000$ (tal y como se mencionó anteriormente) y los vectores a_j y b_j que se muestran en la *Tabla 3.1*. Además, el vector d_j se pasa también como dato en la *Tabla 3.1* y Q se calcula tal y como se enunció en la ecuación (13). Para este caso concreto, se va a analizar cómo cambia el *makespan* de B en función de un valor nulo y negativo de ϵ , $\epsilon = [0, -0.25]$ respectivamente. El objetivo es observar cómo se ordenan los trabajos y analizar qué ocurre si se aprieta o se ablanda el valor de Q en función de ϵ . Ello servirá de ayuda para entender los resultados que se obtendrán de la batería de problemas¹.

Tabla 3.1 Datos del ejemplo numérico del problema.

Trabajos	a_j (Máquina 1)	b_j (Máquina 2)	d_j
1	4	5	13
2	3	7	10
3	6	2	11

¹ Véase el capítulo 4.4 para conocer el significado de la batería de problemas y el 4.5 para la visualización de los resultados.

Antes de continuar, es necesario calcular la secuencia $\pi = [J_A \cup J_B]$ según dicta Johnson para estos datos concretos. Dado que los tiempos de proceso en máquina 1 son más pequeños que los tiempos de proceso en máquina 2 para los trabajos 1 y 2, ambos se deben incluir el vector J . Ahora, se ordena este vector en orden creciente teniendo en cuenta los tiempos de proceso de los trabajos 1 y 2 en máquina 1. Se obtiene que $\pi_1 = [2, 1]$. Por su parte, se hace el mismo proceso para los trabajos del entorno B, es decir, para el trabajo 3. Al tratarse de un único trabajo, se llega a $\pi_2 = [3]$. Por tanto, se obtiene $\pi = [2, 1, 3]$. De esta manera es posible calcular el tiempo de terminación en la máquina 2, resultando $C_{máx,\pi}^B = 17$.

Dado que se trata de un número pequeño de trabajos, se ha decidido resolver el ejemplo mediante la herramienta *solver* de Excel, la cual nos permite apreciar el valor de todas las variables, así como deducir la secuencia obtenida en cada caso para representarla en un Diagrama de Gantt. Puesto que se trata de un número de trabajos pequeño, la solución se obtiene en un tiempo reducido.

- $\varepsilon = 0$

En primer lugar, se procede a resolver el modelo para $Q = C_{máx,\pi}^B = 17$. El modelo matemático planteado se refleja en la *Figura 3.3*:

		VARIABLES																			SIGNO	T.I
		X11	X21	X31	X12	X22	X32	X13	X23	X33	F1	F2	F3	C1	C2	C3	T1	T2	T3			
R1		1	1	1																	=	1
					1	1	1														=	1
								1	1	1											=	1
R2		1			1			1													=	1
			1			1			1												=	1
R3		-4			-3			-6			1										=	0
			-4			-3			-6		-1	1									=	0
R4																					=	0
																					=	0
R5		-9			-10			-8						1							=	0
			-5			-7			-2					-1	1						>=	0
R6																					=	0
																					=	0
R7																					=	0
																					=	0
																					=	0
R8		999987			999990									1							<	1000000
			999987			999990									1						<=	1000000
				999987			999990									1					<=	1000000
								1000000							1						<=	1000017
R9									1000000							1					<=	1000017
										1000000							1				<=	1000017
											1										>=	0
R10												1			1						>=	0
													1			1					>=	0
																	1		1		>=	0
S		0	1	0	1	0	0	0	0	1	3	7	13	10	15	17	0	2	0		2	F.O

Figura 3.3 Resolución del ejemplo en Excel para $\varepsilon = 0$.

La secuencia obtenida es $S_1 = [2, 1, 3]$, el vector de tiempos de terminación en la máquina uno es $F = [3, 7, 13]$, el vector de tiempos de terminación en la máquina dos es $C = [10, 15, 17]$ y el valor de $\sum T_j^A = 2$. En la *Figura 3.4* se representa la secuencia S_1 en un Diagrama de Gantt.

Dado que $C_{máx}^B \leq Q$, el *makespan* del conjunto B no puede sobrepasar el valor de Q , pero sí se ajusta al límite, apurando al máximo la holgura que deja. Ello supone que una leve modificación en el valor de Q modifica el valor del *makespan* y, por ende, puede modificarse la secuencia, tal y como se verá en el siguiente caso de ε .

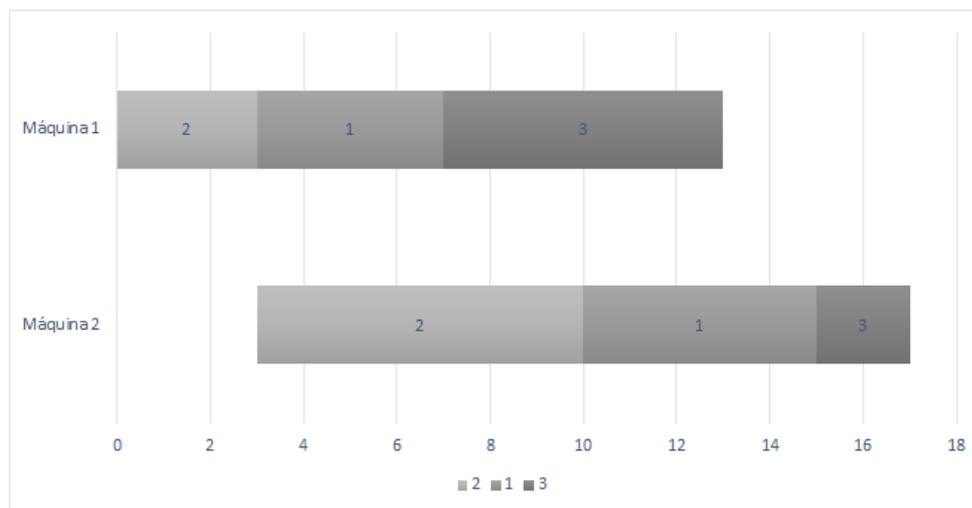


Figura 3.4 Diagrama de Gantt del ejemplo para $\varepsilon = 0$.

- $\varepsilon = -0.25$

En segundo lugar, se resuelve el modelo para $Q = 17 (1 - 0.25) = 12.75$, reduciendo el margen que tiene para crecer el valor $C_{máx}^B$. El MILP se expone en la Figura 3.5:

		VARIABLES																	SIGNO	T.I		
		X11	X21	X31	X12	X22	X32	X13	X23	X33	F1	F2	F3	C1	C2	C3	T1	T2	T3			
R1		1	1	1																	=	1
					1	1	1														=	1
								1	1	1											=	1
R2		1			1			1													=	1
				1			1			1											=	1
R3		-4			-3			-6			1										=	0
R4			-4			-3			-6		-1	1									=	0
				-4			-3			-6		-1	1								=	0
R5		-9			-10			-8						1							=	0
R6			-5			-7			-2					-1	1						>=	0
				-5			-7			-2					-1	1					>=	0
R7			-9			-10			-8		-1				1						>=	0
				-9			-10			-8		-1				1					>=	0
R8		999987			999990									1			-1				<	1000000
			999987			999990									1			-1			<=	1000000
				999987			999990									1			-1		<=	1000000
R9							1000000							1							<=	1000012.75
								1000000							1						<=	1000012.75
									1000000							1					<=	1000012.75
R10										1				1			1				>=	0
											1				1			1			>=	0
S		0	0	1	1	0	0	0	1	0	3	9	13	10	12	18	0	0	5		>=	5
																						F.O

Figura 3.5 Resolución del ejemplo en Excel para $\varepsilon = -0.25$.

La secuencia obtenida es $S_2 = [2, 3, 1]$, el vector de tiempos de terminación en la máquina uno es $F = [3, 9, 13]$, el vector de tiempos de terminación en la máquina dos es $C = [10, 12, 18]$ y el valor de $\sum T_j^A = 5$. En la Figura 3.6 se representa la secuencia S_2 en un Diagrama de Gantt, pudiéndose apreciar los valores obtenidos.

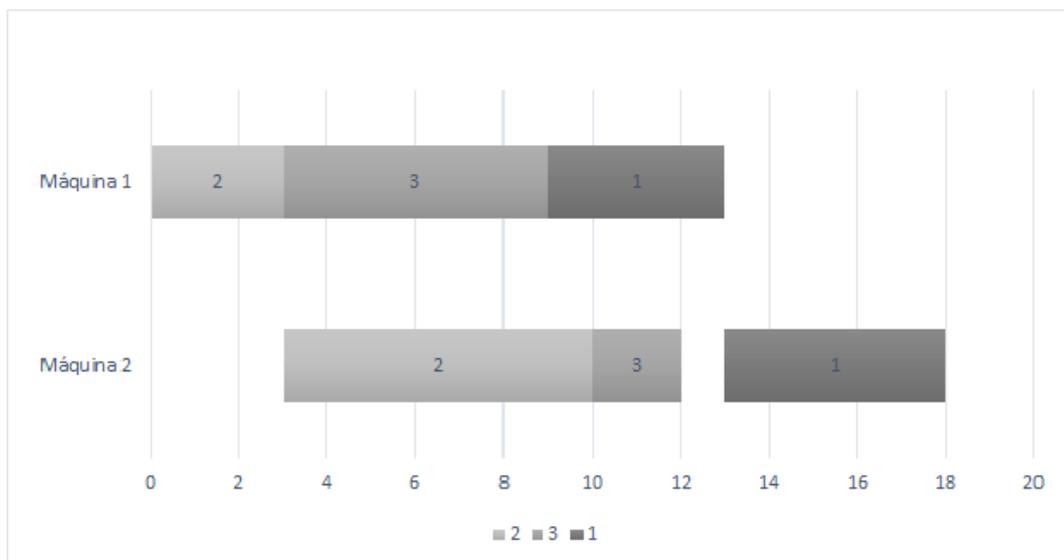


Figura 3.6 Diagrama de Gantt del ejemplo para $\varepsilon = -0.25$.

En este caso, se ha reducido considerablemente el valor de Q , limitando con ello también el valor de $C_{máx}^B$, el cual no puede exceder el valor de 12.75. La aparente mejora que se ha conseguido por un lado reduciendo el valor máximo del *makespan* del entorno B ha originado un peor valor en la función objetivo, pues aumenta el valor del *total tardiness* del cliente A de 2 a 5 ya que se ve obligada a secuenciar el trabajo 1 en último lugar.

La idea es simple: reducciones del valor de Q por debajo del tiempo de proceso máximo obtenido por la secuencia originan un peor valor en la función objetivo puesto que, a costa de disminuir el *makespan* del cliente B, se obliga a procesar los trabajos del entorno A más tarde, agravando el valor de la tardanza total de A, cuya minimización es el objetivo primordial de este proyecto. Esta situación se verá más adelante en el análisis de los resultados obtenidos tras la ejecución, en el capítulo 5 del documento.

4 RESOLUCIÓN EXACTA DEL PROBLEMA

En este capítulo se presentarán las herramientas que han servido para el estudio del problema: Matlab, 4.2, y el solver Gurobi, 4.3. Además, se explicarán detalladamente las baterías de problemas que se van a resolver, 4.4 y se presentarán los resultados obtenidos tras resolver el modelo, 4.5.

4.1 Introducción

Tras la presentación del modelo matemático de programación lineal entera mixta para resolver el problema y haber realizado algunos ejemplos de tamaño reducido para entender el alcance y comportamiento de este, se van a realizar una serie de pruebas con problemas de mayor tamaño para poder conocer el límite al que puede llegar el solver y, de esta forma, conocer el número máximo de trabajos que podemos resolver con el modelo sin que el tiempo de cómputo sea demasiado elevado.

Para la elaboración de este análisis se van a utilizar los programas Matlab y Gurobi. El primero es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio (lenguaje M). De acuerdo con sus desarrolladores, se trata de un lenguaje de alto nivel para el cálculo numérico, la visualización y la programación, con un entorno interactivo utilizado por millones de ingenieros y científicos en todo el mundo. Permite explorar y visualizar ideas, así como colaborar interdisciplinariamente en procesamiento de señales e imagen, comunicaciones, sistemas de control y finanzas computacionales. Gracias a Matlab, es posible analizar datos, desarrollar algoritmos y crear modelos matemáticos. Por todo ello, se ha optado por elegir este software de programación para la creación del modelo matemático.

El otro programa a utilizar es el optimizador de Gurobi. Este programa es un solver que resuelve eficientemente problemas relacionados con la programación lineal (LP), la programación cuadrática (QP), la programación cuadrática entera mixta (MIQP) y la programación cuadrática entera mixta restringida (MIQCP). Gurobi es capaz de soportar una gran variedad de lenguajes de programación y modelado tales como Matlab, Java, C o Python, entre otros.

4.2 Matlab

La herramienta Matlab es, al mismo tiempo, un lenguaje de programación y un entorno. Se pueden crear de manera sencilla programas útiles a gusto del usuario generando archivos de extensión propia, ya que Matlab presenta su propio lenguaje de programación (archivos de extensión .m). Además, una de las principales ventajas radica el empleo de códigos reutilizables, es decir, permite utilizar funciones o *scripts* previamente elaborados. Conviene resaltar que se pueden agrupar en *toolboxes*, es decir, una colección especializada de archivos de extensión .m para trabajar en 34 las clases particulares de problemas.

Como lenguaje de programación, presenta un alto nivel y se encuentra orientado a problemas de cálculo intensivo. Además de esto, permite el desarrollo de aplicaciones técnicas de maneras más eficiente que la programación en lenguajes tradicionales como C, C++ o Python ya que presenta funciones definidas en la librería de Matlab que estas herramientas no tienen.

Matlab no se trata de una simple calculadora equipada capaz de resolver ecuaciones en poco tiempo, pues ofrece un amplio abanico de posibilidades que generan de esta herramienta una visión mucho más atractiva. Matlab es una herramienta muy versátil y realmente útil para efectuar cálculos matemáticos de manera eficiente. En suma a ello, se trata de una plataforma de desarrollo de aplicaciones en la que las *toolboxes*, conjuntos de herramientas inteligentes para la resolución de problemas en áreas de aplicación específica, se pueden desarrollar con facilidad.

En cuanto a su utilidad, conviene destacar el cálculo matricial, álgebra lineal, los polinomios e interpolación, la regresión, el ajuste de funciones, las ecuaciones diferenciales ordinarias, la integración, las funciones y los gráficos en dos y tres dimensiones. Igualmente, se encuentran presentes los módulos de optimización, procesamiento de señales, ecuaciones en derivadas parciales y Simulink (simulación no lineal).

4.3 Gurobi

El optimizador de Gurobi se trata de un solver que sirve para encontrar la solución a un problema de programación matemática. Además, esta herramienta soporta una gran diversidad de lenguajes, entre los que destacan:

- C++, Java, Python, .NET como interfaces orientadas a objetos
- C, Matlab y R como interfaces orientadas a matrices.
- AIMMS, AMPL, GAMS y MPL como enlaces a lenguaje de modelado estándar.
- Premium Solver Platform y Risk Solver Platform como enlaces a Excel.

Como se ha comentado en la sección (4.1), resuelve modelos de programación lineal (LP), programación cuadrática (QP), programación cuadrática restringida (QCP), programación lineal entera mixta (MILP), programación cuadrática mixta (MIQP) y programación cuadrática entera mixta restringida (MIQCP)

El lenguaje principal con el que trabaja Gurobi es C, aunque se puede acceder a este mediante diversos lenguajes de programación. Tiene una interfaz de Python versátil e interactiva, un ejecutable simple de línea de comandos y una interface orientada a la matriz C, que proporcionan orientación a objetos de C++, Java y Python. En la *Figura 4.1* se muestra un boceto del modo de trabajo de Gurobi.

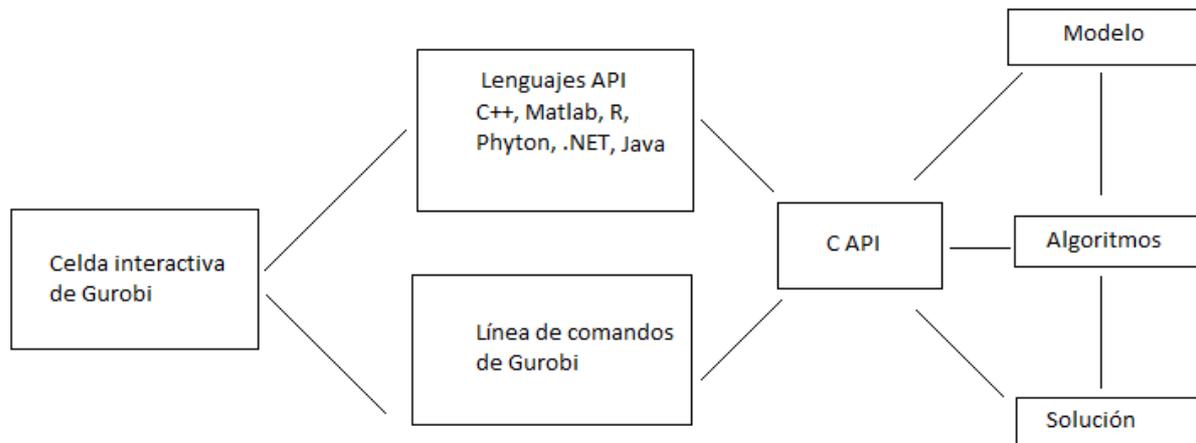


Figura 4.1 Esquema funcionamiento de Gurobi (<http://www.gurobi.com>).

Para poder trabajar con Gurobi y Matlab de manera conjunta, es necesario instalar Gurobi en la herramienta Matlab mediante la interfaz adecuada. Seguidamente, es necesario activar Gurobi dentro de Matlab por medio de la orden “gurobi_setup”. Tras ello, es necesario generar el modelo de programación matemática que se desea resolver, el cual depende de los parámetros de entrada ya mencionados en el subcapítulo (3.2.2). Para crear un modelo matemático en Matlab se deben seguir las instrucciones proporcionadas por Gurobi, el cual resuelve un modelo de la forma:

$$\min(c'x + x'Qx + \alpha)$$

s. a:

$$Ax = b$$

$$l \leq x \leq u$$

$$x'Q_c x + q'x \leq \beta$$

Además, Gurobi obliga a que los datos asociados al modelo se guarden en una estructura (Matlab *struct*) que contiene los siguientes campos:

Obligatorios

- `model.A`: matriz de restricciones del modelo.
- `model.obj`: vector objetivo lineal.
- `model.sense`: signo de las restricciones lineales. Los valores que se permiten son $<$, $>$ ó $=$.
- `model.rhs`: vector de término independiente de las restricciones lineales.

Opcionales

- `model.lb`: límites inferiores de las variables.
- `model.ub`: límites superiores de las variables.
- `model.vtype`: tipos de variables. Es un vector de caracteres. Los valores que se permiten son "C" (continua), "B" (binario), "I" (entero), "S" (semi-continuo), o "N" (semi-entero). Si se omite, cada variable se trata como una variable continua.
- `model.modelsense`: objetivo de optimización. Los valores permitidos son "min" (Minimizar) o "max" (Maximizar). Si se omite, el objetivo del modelo por defecto es de minimización.
- `model.modelname`: nombre para todo el modelo.
- `model.objcon`: desplazamiento de la función objetivo (α en el modelo anterior).
- `model.vbasis`: vector de estado de las variables. Se utiliza para proporcionar un avanzado punto de partida para el algoritmo Simplex.
- `model.cbasis`: vector de estado de las restricciones. Se utiliza para proporcionar un avanzado punto de partida para el algoritmo Simplex.
- `model.Q`: matriz objetivo cuadrática.
- `model.cones`: restricciones de segundo orden.
- `model.quadcon`: restricciones cuadráticas.
- `model.start`: vector de inicio MIP. El solucionador MIP intentará construir una solución inicial de este vector. Cuando está presente debe especificar un valor inicial para cada variable.
- `model.varnames`: nombres de las variables. Cuando está presente, cada elemento de la matriz define el nombre de una variable.
- `model.constrnames`: nombres de las restricciones. Cuando está presente, cada elemento de la matriz define el nombre de una restricción.

Gurobi devuelve una estructura de Matlab denominada *result* en la que almacena diversos resultados obtenidos en la resolución del problema. Los principales campos que guarda son:

- `result.status`: el estado de la optimización, lo devuelve como una cadena. Si el resultado es "óptimo", indica que se ha encontrado una solución óptima al modelo. También enuncia la no factibilidad del modelo, entre otros resultados posibles
- `result.objval`: el valor objetivo de la solución encontrada.
- `result.objbound`: mejor solución disponible para un límite inferior para la minimización y un límite superior para la maximización.
- `result.runtime`: tiempo en segundos de la resolución de la optimización.
- `result.x`: la solución calculada. Este vector contiene un valor para cada columna de la matriz A.
- `result.slack`: holgura de cada restricción de la solución. Este vector contiene un valor para cada fila de la matriz A.
- `result.intercount`: número de iteraciones que realiza.

- `result.nodecount`: número de nodos explorados.
- `result.qcslack`: holgura cuadrática de cada restricción de la solución actual.
- `result.rc`: coste reducido de la solución actual. Este campo solo está disponible para modelos continuos.
- `result.pi`: valores duales para la solución computarizada.
- `result.qcpi`: valores duales asociados a las restricciones cuadráticas.
- `result.vbasis`: valores de estado de las variables para el óptimo.
- `result.cbasis`: valores de estado de las restricciones para el óptimo.

4.4 Batería de problemas

Dado que la finalidad de este Trabajo Fin de Grado es analizar la capacidad del optimizador de Gurobi para resolver el problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ de manera exacta, se han generado diversos problemas en función del número de trabajos (n) en distintos archivos de texto, conocidos como instancias. Cada instancia presenta el número de máquinas (m), que siempre será constante de valor igual a dos, el número de trabajos que contiene la instancia en cuestión, así como los vectores a_j y b_j que presentan el tiempo de procesamiento del trabajo j en la máquina 1 y en la máquina 2, respectivamente. Estos tiempos de proceso se han generado mediante una distribución uniforme comprendida entre 1 y 99, es decir, $a_j = U(1,99)$ y $b_j = U(1,99)$. Esta colección de problemas proviene de una batería de hasta 30 instancias que es proporcionada como dato para la elaboración del análisis del problema. De esta manera, para cada número de trabajos existirán un total de 30 instancias enumeradas desde el cero hasta el veintinueve, generadas en archivos `.txt` con el formato "FS2PM_n_numeroinstancia". En la *Figura 4.2* se muestra un ejemplo para $n = 8$ y $\text{numeroinstancia} = 1$:

```

2
8
51 15
37 14
52 77
99 38
66 16
22 15
33 33
82 92
|

```

Figura 4.2 Ejemplo de instancia de 8 trabajos.

Asimismo, para estudiar el comportamiento del solver en función del número de trabajos, se lleva a cabo una clasificación de las distintas baterías de problemas en 3 grupos: batería pequeña, batería mediana y batería grande. Estos trabajos serán recogidos en un vector de enteros denominado *numerotrabajos*.

La batería pequeña es aquella en la que se utilizan $n \in \{8, 10, 12, 14, 16, 18, 20\}$. Como se ha comentado anteriormente, se van a resolver 30 instancias para cada valor de n , siendo un total de 210 instancias. Además, para cada instancia se va a analizar la alteración del parámetro $\varepsilon = [-0.25, 0, 0.25, 0.5, 0.75]$. Por ello, cada una de las 210 instancias será resultado un total de 5 veces, una para cada valor del parámetro ε , dando lugar a 1050 ejecuciones del problema para esta batería.

La batería mediana se trata de aquella que emplea $n \in \{22, 24, 26, 28, 30\}$. Para cada valor de n se van a resolver 30 instancias, generando un total de 150 instancias. Al igual que la batería pequeña, se resolverá 7 veces cada una de ellas, luego se tendrán 750 ejecuciones del programa para la batería de tamaño medio.

Finalmente, la batería grande alberga valores de $n \in \{35, 40, 45, 50\}$. Como en los casos anteriores, las 30 instancias serán resueltas para cada trabajo, dando lugar a 120 instancias. Estas, se ejecutarán 5 veces cada una de ellas, originando 600 ejecuciones en total. Computacionalmente, se trata de la batería más pesada. Gurobi estaría meses analizando las restricciones del modelo para alcanzar el óptimo dado el elevado número de trabajos. Por ello, se emplea un tiempo límite de ejecución de Gurobi de 900 segundos. Igualmente, este tiempo límite se ha empleado para las otras dos baterías pues, en algunos casos, el modelo empieza a complicarse para valores de n que pertenecen a la batería mediana. Esta orden se dicta mediante la orden “`params.timelimit = 900`” en el código de Matlab, entendiéndose Gurobi que el campo *params* ha sido alterado y debe tenerse en cuenta a la hora de la resolución del problema. Al igual que *result* y *model*, se trata de una estructura que Matlab le pasa a Gurobi cuando resuelve el MILP.

Una vez se tienen todas las instancias de las tres baterías en archivos de texto, se leen los datos a través de Matlab y se generan los valores de Q y vector d_j . Posteriormente, se crean las restricciones y la función objetivo del modelo matemático de programación lineal entera mixta y se llama a Gurobi para su resolución. Para concluir, se escriben los datos obtenidos en un fichero de extensión .txt. En la *Figura 4.3* se representa un esquema de bloques del funcionamiento de ejecución realizado.



Figura 4.3 Funcionamiento de ejecución.

Como se mencionó anteriormente, para la generación del código de programación en la herramienta Matlab se ha elaborado un *script*, que no es más que un archivo que contiene líneas secuenciales de comandos y llamadas a funciones de Matlab. Este archivo, de extensión propia de Matlab (extensión .m), se denomina “FS.m” y se ejecuta una vez para cada valor de n . La diferencia entre el *script* y la función de Matlab radica en la complejidad. Normalmente, las funciones son programas que realizan tareas más elaboradas que, a partir de ciertos argumentos de entrada, realizan una serie de cálculos programados en el editor, con una serie de líneas ejecutables y devuelven uno o varios argumentos de salida al llamar la función desde la ventana de comando u otra función.

El código de Matlab consta de distintas fases: desde la lectura de datos de los ficheros de texto hasta la escritura de datos en los ficheros de salida, pasando por la creación del modelo matemático. Para comprender bien su contenido, se ha elaborado un flujograma del código tal y como se muestra en la *Figura 4.4*.

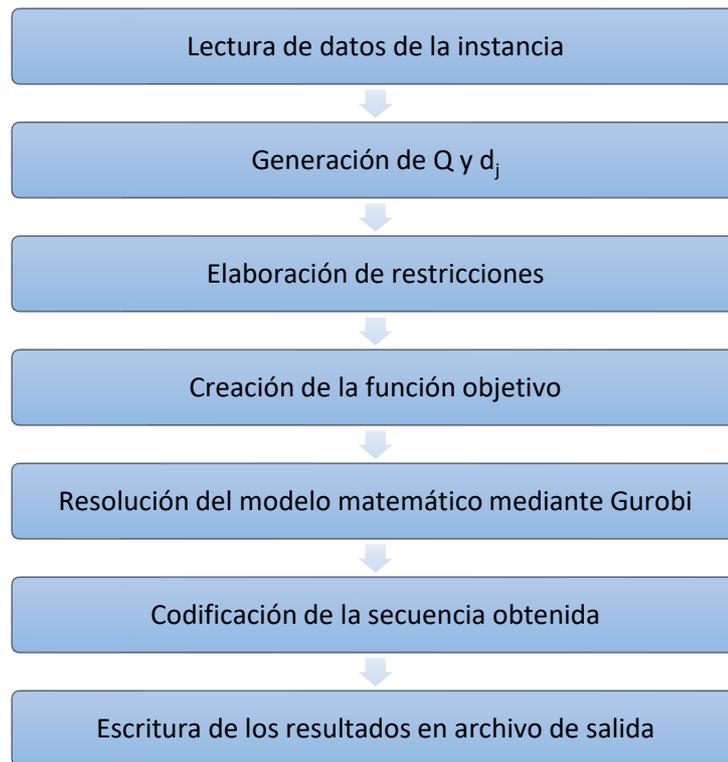


Figura 4.4 Flujograma del código en Matlab.

Por su parte, los datos de salida se muestran en un archivo de texto denominado “salida.txt”, en el que aparecen los resultados obtenidos de cada trabajo para cada valor de ϵ . En la *Figura 4.5* se muestra un ejemplo para $n = 8$ y $\epsilon = 0$:

```

salida: Bloc de notas
Archivo Edición Formato Ver Ayuda
FS2PM_8_0
Total tardiness de A: 270
Makespan de B: 396
El valor de Q es: 396
Tiempo de ejecución: 0.243616
Secuencia óptima: 1 3 4 2 8 7 5 6

FS2PM_8_1
Total tardiness de A: 101
Makespan de B: 477
El valor de Q es: 477
Tiempo de ejecución: 0.202713
Secuencia óptima: 4 2 1 3 7 8 6 5

FS2PM_8_2
Total tardiness de A: 242
Makespan de B: 505
El valor de Q es: 505
Tiempo de ejecución: 0.361694
Secuencia óptima: 3 4 1 2 7 6 5 8

FS2PM_8_3
Total tardiness de A: 354
Makespan de B: 526
El valor de Q es: 526
Tiempo de ejecución: 0.345692
Secuencia óptima: 4 1 2 5 3 6 7 8

FS2PM_8_4
Total tardiness de A: 67
Makespan de B: 512
El valor de Q es: 512
Tiempo de ejecución: 0.235977
Secuencia óptima: 4 2 3 7 1 5 8 6
  
```

Figura 4.5 Ejemplo de archivo de salida para 8 trabajos y $\epsilon = 0$.

Como se observa en el archivo, los datos a estudiar son el *total tardiness* del cliente A, el valor del *makespan* del cliente B, el tiempo que tarda Gurobi en resolver esa instancia y el valor de Q obtenido. Estos valores se

presentarán en el apartado 4.5 y se analizarán en el capítulo 5.

El código en Matlab de la recogida de datos de los archivos de texto, la creación de los valores de Q y vector d_j , la generación del modelo matemático de programación lineal entera mixta, la resolución mediante Gurobi, la codificación de la secuencia obtenida, la escritura de los resultados obtenidos en un archivo de salida se muestra en el *Anexo A*.

4.5 Resultados obtenidos

Tal y como se mencionó en la sección anterior 4.4, se han analizado 3 baterías de problemas distintas con el fin de estudiar el comportamiento del problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$. Para cada instancia, se toman los datos del archivo de texto en cuestión, cuyo formato es “FS2PM_n_numeroinstancia”, siendo n el número de trabajos y *numeroinstancia* el número de la instancia que se pasa como argumento en ese momento y que varía entre 0 y 29. Se genera el modelo matemático de programación lineal entera mixta mediante la herramienta Matlab y se resuelve de manera exacta. Una vez resuelto, se copian los valores obtenidos en un archivo de texto denominado “salida.txt” para cada instancia. En este archivo, se muestra el tiempo de cómputo de Gurobi en resolver el problema, el valor del *total tardiness* del cliente A, el valor del *makespan* del cliente B, los valores de Q obtenidos y la optimalidad, es decir, la capacidad que presenta Gurobi para resolver de manera óptima el problema sin llegar al tiempo límite establecido de 900 segundos. Para ello, se han contabilizado el número de instancias que han llegado al tiempo máximo de 900 segundos, se ha dividido entre 30 instancias en total y se ha restado a la totalidad de instancias tal y como figura en la ecuación (14):

$$\text{optimalidad (\%)} = 100 \left(1 - \frac{\text{instancias que exceden el tiempo límite}}{30} \right) \quad (14)$$

Estos cinco parámetros son el principal objeto de estudio de este Proyecto y, clasificados para cada batería de problemas, se muestran en las tablas que aparecen a continuación. Cada fila de la tabla corresponde el número de trabajos en cuestión, mientras en las columnas se indica el valor de ϵ correspondiente. En primer lugar, se mostrará el tiempo de ejecución, seguido de la tardanza total del cliente A, el *makespan* del cliente B, el valor de Q y la optimalidad.

Dado que se tienen un total de 30 instancias para cada número de trabajos, se ha realizado un promedio de todas ellas, obteniendo un único valor que representa el valor que puede obtener cada parámetro para cada valor de ϵ . Teniendo en cuenta la aleatoriedad que presentan los tiempos de proceso y las fechas de entrega, estos valores medios representan de manera simbólica lo que puede ocurrir en una planta productiva real. Para acceder a todos los resultados obtenidos de manera completa de este problema es necesario ver el *Anexo B*.

A continuación, se muestran los resultados medios de las 30 instancias obtenidos para cada batería:

1. Batería pequeña

Tabla 4.1 Tiempo de ejecución de Gurobi (s). Batería pequeña.

	Tiempo de ejecución de Gurobi (s)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=8	0,25	0,25	0,26	0,21	0,23
n=10	0,15	0,23	0,29	0,25	0,27
n=12	0,48	1,63	1,00	0,87	1,00
n=14	0,90	3,76	3,41	3,76	1,62
n=16	2,40	10,26	10,67	8,33	4,14
n=18	2,41	7,34	6,56	3,85	4,28
n=20	25,91	12,16	41,41	5,94	8,26

Tabla 4.2 Total Tardiness de A (u.t). Batería pequeña.

	Total Tardiness de A (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=8	593,13	185,46	144,80	124,3	167,93
n=10	751,70	202,86	171,86	187,3	178,36
n=12	1047,50	196,86	207,53	177,13	202,50
n=14	1346,36	276,46	212,56	241,20	260,43
n=16	1605,26	280,83	302,83	324,30	229,63
n=18	1697,23	267,66	214,00	223,93	211,63
n=20	2428,23	203,53	175,50	135,93	164,33

Tabla 4.3 Makespan de B (u.t). Batería pequeña.

	Makespan de B (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=8	347,13	471,70	552,73	587,06	682,73
n=10	424,36	583,20	682,40	748,43	829,10
n=12	505,63	690,46	784,16	914,83	1009,33
n=14	525,36	713,63	837,43	1008,93	1064,50
n=16	604,70	806,80	974,90	1144,70	1206,53
n=18	695,23	931,46	1112,66	1310,10	1528,03
n=20	774,93	1039,00	1220,66	1390,00	1651,10

Tabla 4.4 Valores de Q (u.t). Batería pequeña.

	Valores de Q (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=8	354,66	472,80	591,10	709,40	827,466
n=10	437,90	583,70	729,76	875,733	1021,60
n=12	518,03	690,46	863,23	1035,93	1208,50
n=14	535,63	713,73	892,50	1070,73	1249,36
n=16	608,06	810,43	1012,93	1216,06	1418,50
n=18	698,63	931,46	1164,36	1397,33	1630,10
n=20	779,30	1039,00	1298,83	1558,56	1818,30

Tabla 4.5 Optimalidad (%). Batería pequeña.

	Optimalidad (%)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=8	100,00	100,00	100,00	100,00	100,00
n=10	100,00	100,00	100,00	100,00	100,00
n=12	100,00	100,00	100,00	100,00	100,00
n=14	100,00	100,00	100,00	100,00	100,00
n=16	100,00	100,00	100,00	100,00	100,00
n=18	100,00	100,00	100,00	100,00	100,00
n=20	100,00	100,00	96,66	100,00	100,00

Los tiempos de ejecución son pequeños en esta batería dado que el número de trabajos no es muy elevado. Los valores de la tardanza total de A, el makespan de B y los valores de Q van aumentando conforme crece el número de trabajos. Se obtienen valores razonables en los parámetros de estudio del problema dada la optimalidad del mismo para estos valores de n .

Los tiempos de cómputo de Gurobi son relativamente bajos. Para los primeros valores de n ascienden a minutos, pero conforme crece este valor, llegan incluso a varias horas.

2. Batería mediana

Tabla 4.6 Tiempo de ejecución de Gurobi (s). Batería mediana.

	Tiempo de ejecución de Gurobi (s)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=22	24,95	11,06	6,36	22,71	45,97
n=24	29,47	111,73	12,96	26,03	51,79
n=26	87,54	93,04	14,21	31,20	17,65
n=28	152,85	157,00	145,27	225,74	174,69
n=30	189,19	130,27	108,21	95,29	104,25

Tabla 4.7 Total Tardiness de A (u.t). Batería mediana.

	Total Tardiness de A (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=22	2313,60	230,73	147,86	184,70	226,33
n=24	2208,20	310,23	152,33	203,93	220,63
n=26	2398,46	216,90	108,93	189,80	166,70
n=28	2406,63	225,86	225,93	287,00	235,96
n=30	2715,53	227,03	209,80	200,03	182,50

Tabla 4.8 Makespan de B (u.t). Batería mediana.

	Makespan de B (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=22	852,50	1138,06	1383,96	1558,73	1858,66
n=24	952,00	1277,30	1551,20	1839,20	2203,96
n=26	1003,63	1348,63	1655,43	1893,53	2273,90
n=28	1074,86	1446,50	1786,30	2100,66	2293,33
n=30	1142,76	1528,70	1891,43	2175,30	2427,86

Tabla 4.9 Valores de Q (u.t). Batería mediana.

	Valores de Q (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=22	854,26	1139,00	1423,76	1708,60	1993,26
n=24	958,03	1277,30	1596,66	1916,03	2235,33
n=26	1012,00	1349,06	1686,16	2024,00	2361,06
n=28	1085,06	1446,50	1808,06	2170,10	2531,56
n=30	1146,86	1528,90	1911,03	2293,73	2675,76

Tabla 4.10 Optimalidad (%). Batería mediana.

	Optimalidad (%)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=22	100,00	100,00	100,00	100,00	100,00
n=24	100,00	90,00	100,00	100,00	96,66
n=26	93,33	93,33	100,00	100,00	100,00
n=28	93,33	93,33	93,33	80,00	90,00
n=30	90,00	93,33	96,66	96,66	96,66

Dado que el número de trabajos aumenta, el tiempo de ejecución de Gurobi también crece, observándose un enorme aumento para ciertos valores de ϵ . Lógicamente, también crece la tardanza total de A, el *makespan* de B y los valores de Q al aumentar el número de trabajos ya que crece la cantidad de trabajos a secuenciar. Por su parte, la optimalidad va disminuyendo y, en algún caso, llega a estar por debajo del 90%, lo que quiere decir que Gurobi empieza a obstruirse ligeramente y no consigue alcanzar el óptimo para el tiempo máximo de 900 segundo establecido.

En esta batería, la resolución para todos los valores de ϵ asciende a un número considerable de horas puesto que, como se ha mencionado, la resolución de algunas instancias lleva a Gurobi al tiempo límite máximo. Esto difiere ligeramente con respecto a la batería anterior.

3. Batería grande**Tabla 4.11** Tiempo de ejecución de Gurobi (s). Batería grande.

	Tiempo de ejecución de Gurobi (s)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=35	468,15	291,73	272,86	356,11	218,63
n=40	391,56	246,84	409,94	304,44	324,42
n=45	560,28	370,59	307,46	235,44	438,97
n=50	692,27	400,18	295,67	351,12	389,96

Tabla 4.12 Total Tardiness de A (u.t). Batería grande.

	Total Tardiness de A (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=35	3179,40	282,70	357,80	358,96	256,46
n=40	3981,86	216,36	276,93	359,96	308,70
n=45	4609,20	272,20	376,70	239,50	308,10
n=50	5420,10	337,16	170,20	326,56	293,40

Tabla 4.13 Makespan de B (u.t). Batería grande.

	Makespan de B (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=35	1344,43	1803,06	2203,76	2646,73	2981,26
n=40	1552,40	2071,20	2578,63	3083,70	3492,63
n=45	1754,13	2346,10	2883,10	3380,63	3890,56
n=50	1895,20	2533,36	3166,43	3708,63	4316,06

Tabla 4.14 Valores de Q (u.t). Batería grande.

	Valores de Q (u.t)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=35	1352,30	1803,06	2253,86	2704,76	3155,36
n=40	1557,73	2076,60	2596,10	3115,06	3634,33
n=45	1759,73	2346,10	2932,50	3519,56	4105,83
n=50	1900,26	2533,36	3167,06	3800,20	4433,63

Tabla 4.15 Optimalidad (%). Batería grande.

	Optimalidad (%)				
	$\epsilon=-0,25$	$\epsilon=0$	$\epsilon=0,25$	$\epsilon=0,5$	$\epsilon=0,75$
n=35	53,33	80,00	76,66	70,00	86,66
n=40	70,00	83,33	63,33	83,33	70,00
n=45	50,00	76,66	70,00	83,33	66,66
n=50	36,66	60,00	83,33	70,00	63,33

Los tiempos de ejecución son los más elevados de las tres baterías. Aumentan mucho por término medio ya que, como se explicó en la sección 4.1, el número de soluciones crece de manera factorial, ampliando el abanico de soluciones admisibles en las que el optimizador de Gurobi tiene que buscar. Por ello, la resolución de todas las instancias para todos los valores de ϵ para cada número de trabajos requiere de muchas horas. La optimalidad decae abruptamente puesto que llega al tiempo máximo de 900 segundos en muchísimas ocasiones. Esto ayuda a entender la complejidad del problema y de la secuenciación de trabajos en un entorno *Flow Shop*.

En cuanto a los valores del *total tardiness* del cliente A, el *makespan* del cliente B y los valores de Q , aumentan considerablemente tal y como era de esperar dado que se eleva el número de trabajos a secuenciar.

Conviene resaltar que los valores de $C_{m\acute{a}x}^B$ se encuentran en todo momento por debajo de los valores obtenidos de Q , tal y como limitaba el entorno y que, posteriormente, se reflejó como una de las restricciones al problema $F2 \mid \mu \mid \sum T_j^A : C_{m\acute{a}x}^B \leq Q$ dentro del modelo matemático de programación. Esto se examinará detenidamente en el capítulo 5 en el que se analizarán los resultados obtenidos.

5 ANÁLISIS DE RESULTADOS

En este capítulo se procede el análisis experimental de los resultados obtenidos para cada una de las baterías del problema F2 $\mid \text{prmu} \mid \sum T_j^A : C_{m\acute{a}x}^B \leq Q$, resuelto de forma exacta mediante un modelo de programación lineal entera mixta (MILP).

Como ya se explicó con anterioridad en la sección 4.4, se parte de una batería de problemas de distintos tamaños. En primer lugar, la batería pequeña utiliza $n \in \{8, 10, 12, 14, 16, 18, 20\}$, seguida de la batería mediana, que emplea $n \in \{22, 24, 26, 28, 30\}$, y finalmente la batería grande, que maneja $n \in \{35, 40, 45, 50\}$. Todas ellas, contienen un total de treinta instancias para cada trabajo concreto. También cabe destacar que el número de máquinas es fijo a lo largo de todo el proyecto al no ser objeto de estudio. Sería viable investigar los cambios que ofrece aumentar el número de máquinas en este problema en investigaciones futuras, tal y como se detallará en el capítulo 6.

Debido a la complejidad que presentan algunas instancias para valores más elevados del número de trabajos, cabe esperar que el tiempo de cómputo sea demasiado grande. Para evitarlo, se ha decidido limitar el tiempo de ejecución para cada instancia a 15 minutos, es decir, 900 segundos. De esta forma, si un modelo no alcanza la solución óptima en ese tiempo, devolverá la mejor solución factible encontrada hasta el momento. Además, se considerará que esa instancia no se ha podido resolver de forma óptima ya que habrá encontrado una solución buena, pero seguramente, no la mejor, reduciéndose la optimalidad.

Por tanto, el balance de resultados de cada batería es el siguiente:

5.1 Batería pequeña

Los tiempos computacionales medios obtenidos por el optimizador de Gurobi han sido relativamente bajos ya que el número de trabajos es pequeño. Por término medio, se observa una tendencia lineal para toda la batería, salvo para $n = 20$, que presenta una gran diferencia con respecto a los demás trabajos, tal y como se aprecia en la *Figura 5.1*. Este hecho lleva a pensar que, en la siguiente batería, los tiempos empezarán a aumentar notablemente.

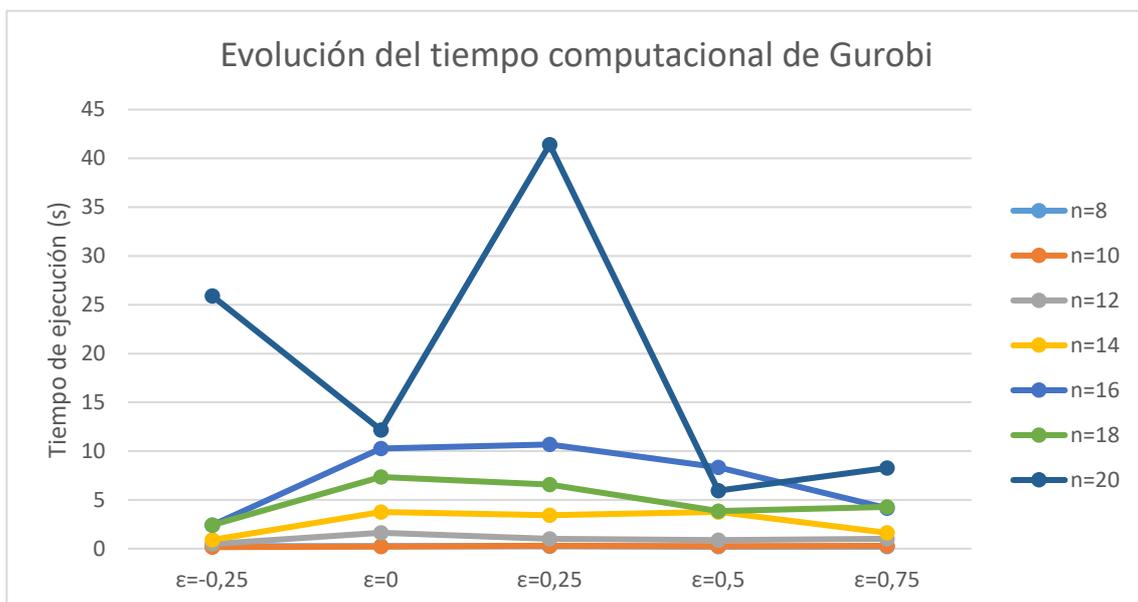


Figura 5.1 Tiempo de ejecución medio de Gurobi (s). Batería pequeña.

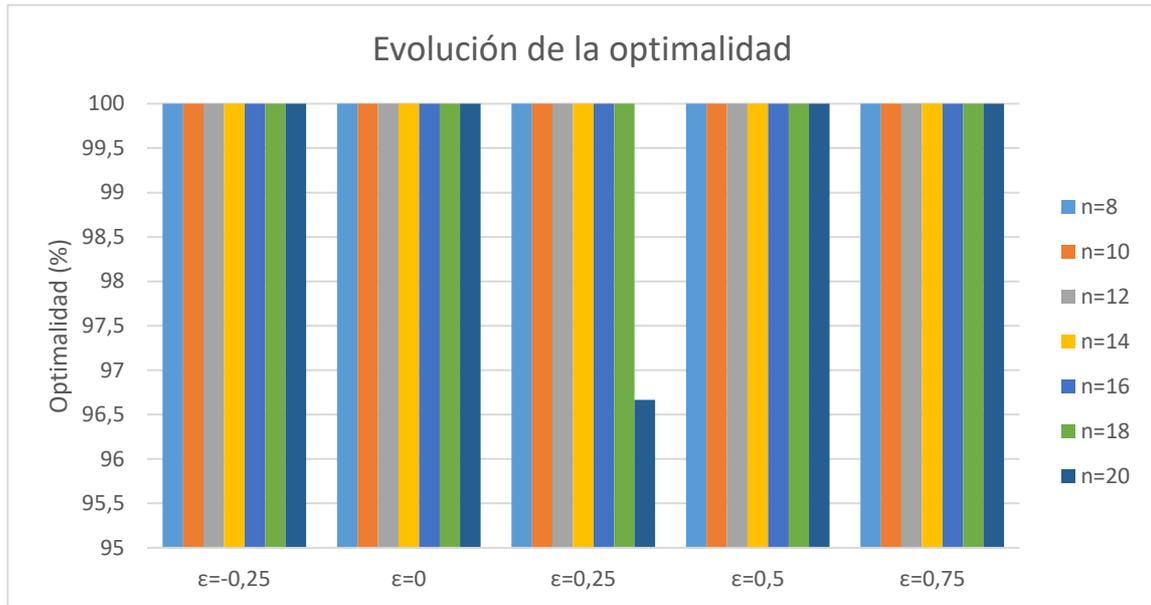


Figura 5.2 Comparativa de la optimalidad (%). Batería pequeña.

El motivo por el que la curva del tiempo de ejecución para $n = 20$ tenga un repunte en $\varepsilon = 0,25$ radica en la pérdida de la optimalidad, que para los anteriores trabajos era del 100% pero, ahora disminuye para este valor de ε , pasando a valer 96,67%. Esto se muestra en la *Figura 5.2*.

También es importante analizar qué sucede con la tardanza total del cliente A para cada valor de ε . En la *Figura 5.3* se muestra el análisis para la batería pequeña.

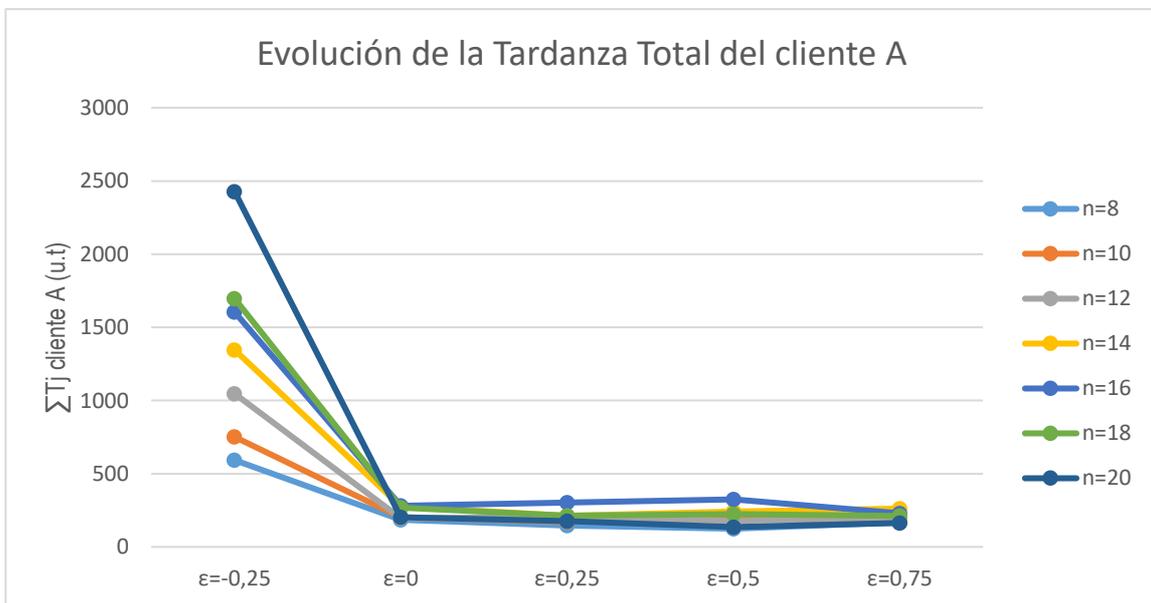


Figura 5.3 Comparativa de la tardanza total del cliente A (u.t). Batería pequeña.

Normalmente, los valores de la tardanza total del cliente A aumentan conforme crece el valor de n . Esto es así hasta $n = 16$, pero luego decae para $n = 18$ y $n = 20$. Esto se explica por la aleatoriedad de las fechas de entrega, las cuales siguen una distribución discreta uniforme. Además de ello, lo que más llama la atención de la *Figura 5.3* es el salto que presenta el primer valor de ε , $\varepsilon = -0,25$. Esto ocurre por el ajuste que se hace con $Q = C_{máx,\pi}^B (1 + \varepsilon)$. El hecho de reducir el valor de Q hace que el tiempo de finalización máximo del cliente B disminuya puesto que se debe satisfacer la restricción $C_{máx}^B \leq Q$. Este decrecimiento beneficia a la función objetivo del cliente B ya que se reduce, pero afecta negativamente a la función objetivo del cliente A. Al tener que cumplirse la ecuación, se secuencian antes los trabajos de B que los de A, por lo que gran parte de los trabajos del primer cliente van tarde. Esto explica la considerable subida de los valores del *total tardiness*.

Tras ello, es importante analizar la evolución del tiempo de finalización del cliente B, el cual se muestra en la *Figura 5.4*.

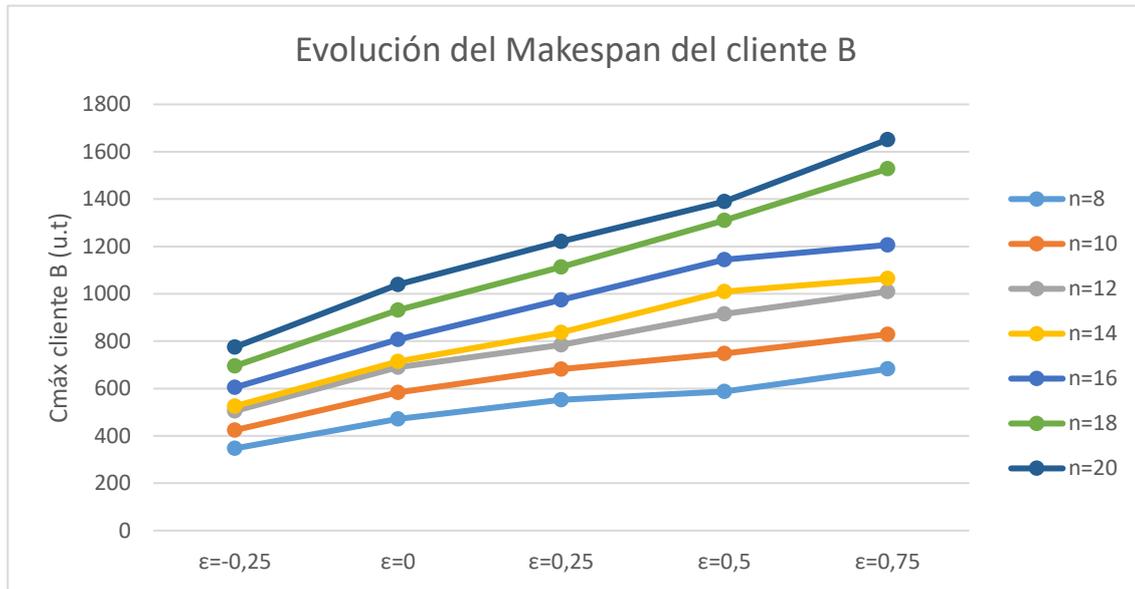


Figura 5.4 Comparativa del makespan del cliente B (u.t). Batería pequeña.

Dado que se tiene que cumplir la ecuación $C_{máx}^B \leq Q$, tiene coherencia que el valor del *makespan* crezca conforme aumenta el valor de Q , ya que depende de ϵ . Si se aumenta ϵ , crece Q y, por tanto, crece el tiempo de finalización máximo del cliente B. Por ende, el responsable de que el valor del *makespan* de B se vea afectado es Q y, en definitiva, ϵ . En la *Figura 5.5* se muestra la transformación de los valores de Q en función de ϵ .

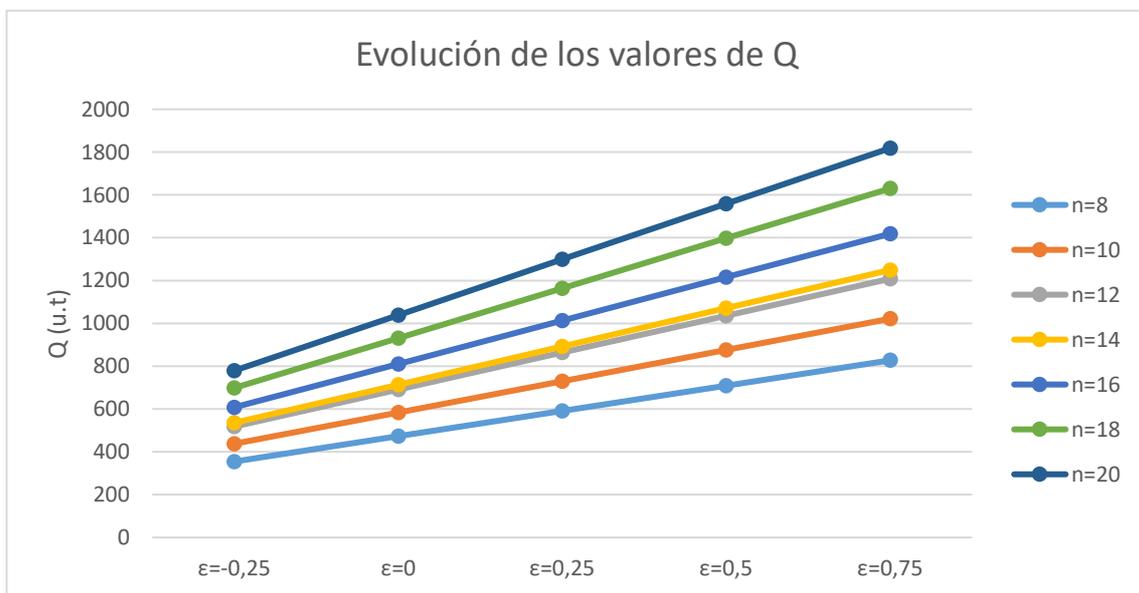


Figura 5.5 Comparativa de los valores de Q (u.t). Batería pequeña.

Se aprecia el cumplimiento de la restricción $C_{máx}^B \leq Q$ en la *Figura 5.6.*, en la que, para cada valor de n , se muestra cómo en todo valor de ϵ , el tiempo de finalización máximo del cliente B está siempre por debajo

del valor de Q .

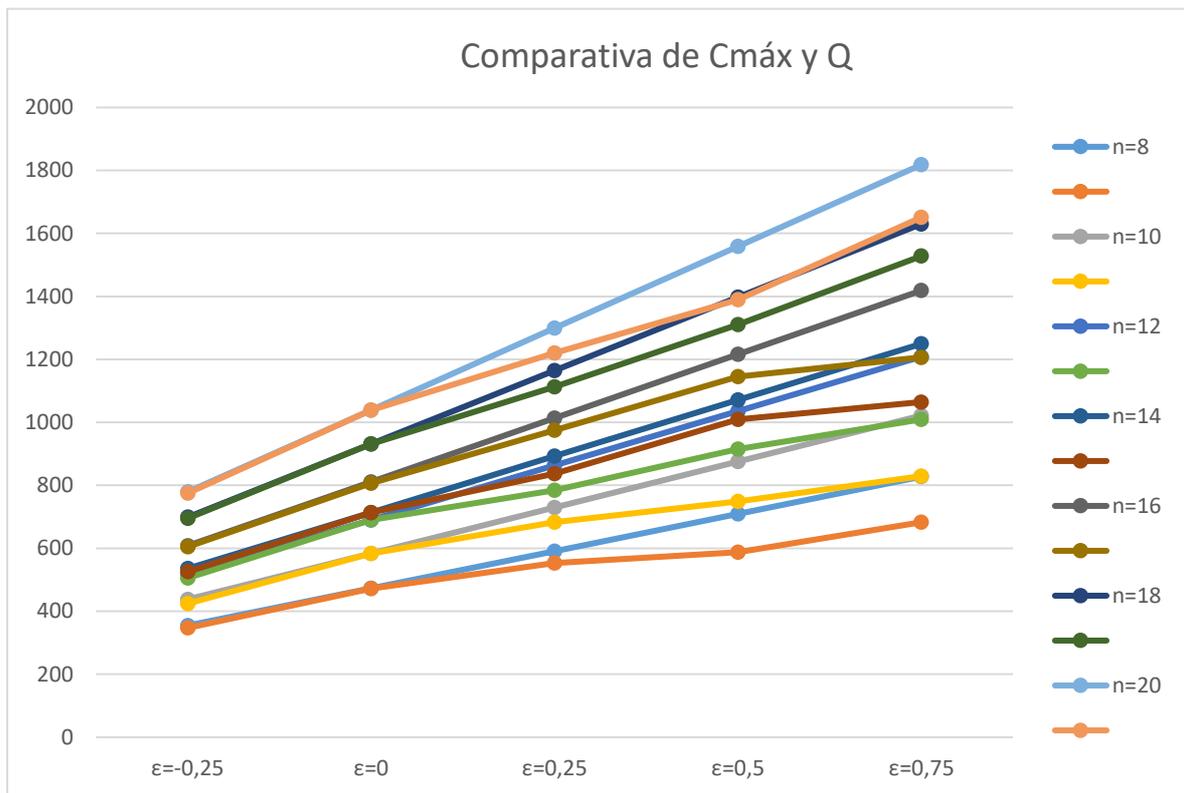


Figura 5.6 Comparativa entre Cmáx y Q (u.t). Batería pequeña.

5.2 Batería mediana

En la batería de tamaño medio, los tiempos medios computacionales de Gurobi aumentan considerablemente con respecto a la batería anterior. Para algunos valores de ϵ , superan incluso los doscientos segundos, lo que significa que muchas instancias llegan al tiempo límite establecido de novecientos segundos, pues se trata del tiempo de ejecución medio. Esto se observa en la *Figura 5.7*.

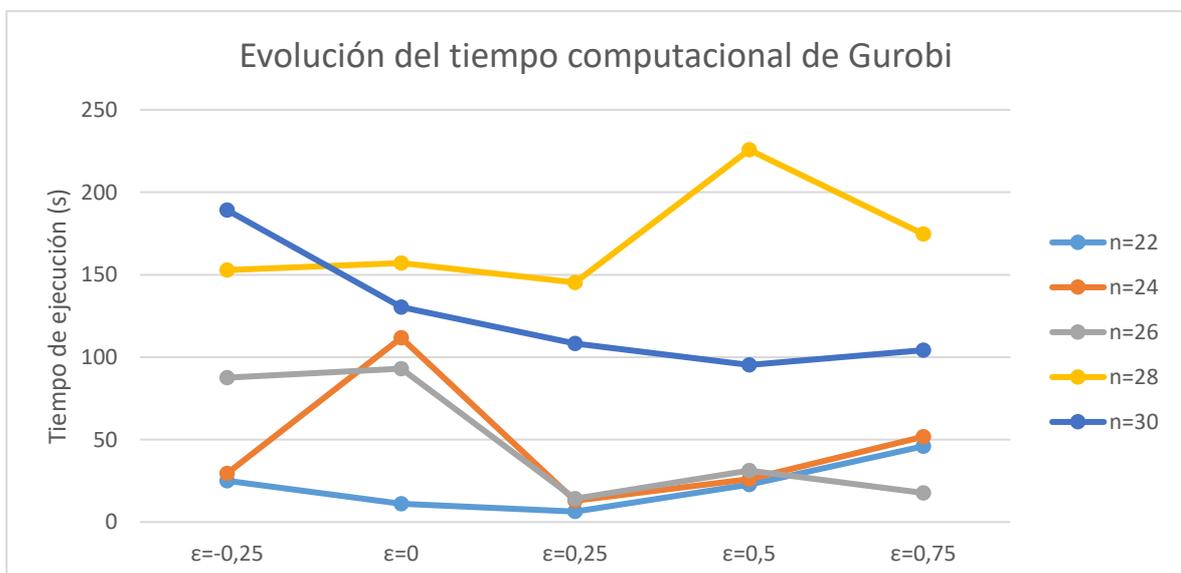


Figura 5.7 Tiempo de ejecución medio de Gurobi (s). Batería mediana.

Salvo en casos puntuales de ϵ , la tendencia de la curva para cada número de trabajo es similar. Se observa un mayor valor en el caso $\epsilon = -0,25$ puesto que se reduce el valor de Q , dificultando el problema.

Por su parte, la optimalidad recoge la eficiencia del optimizador de Gurobi a la hora de resolver el problema. En la *Figura 5.8* se recogen los resultados para esta batería de problemas.

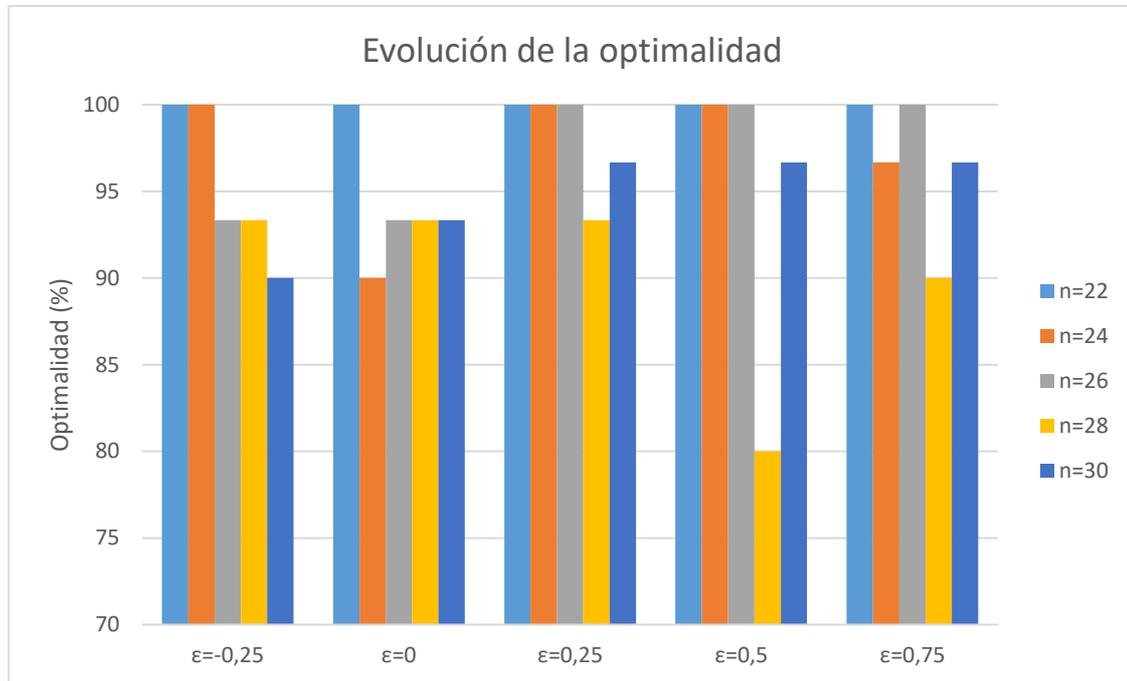


Figura 5.8 Comparativa de la optimalidad (%). Batería mediana.

Para los dos últimos valores del número de trabajos, es decir, $n = 28$ y $n = 30$, la optimalidad decae más que en otros casos. Como esta depende del número de instancias que llegan al tiempo máximo establecido para la ejecución del problema mediante Gurobi, esto implica que el número de instancias que exceden el tiempo límite asciende. En la *Figura 5.9* y en la *Figura 5.10* se aprecia la cantidad de instancias que alcanzan el tiempo límite, reduciendo la optimalidad de Gurobi para resolver el problema en ese tiempo fijado.

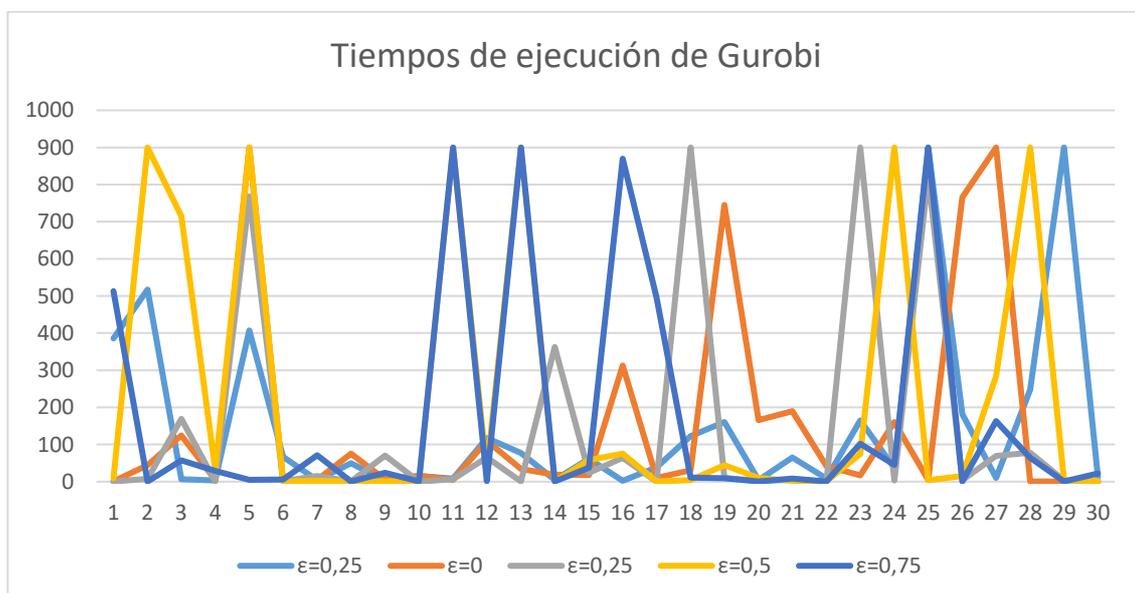


Figura 5.9 Tiempos de ejecución de Gurobi (s) para $n=28$.

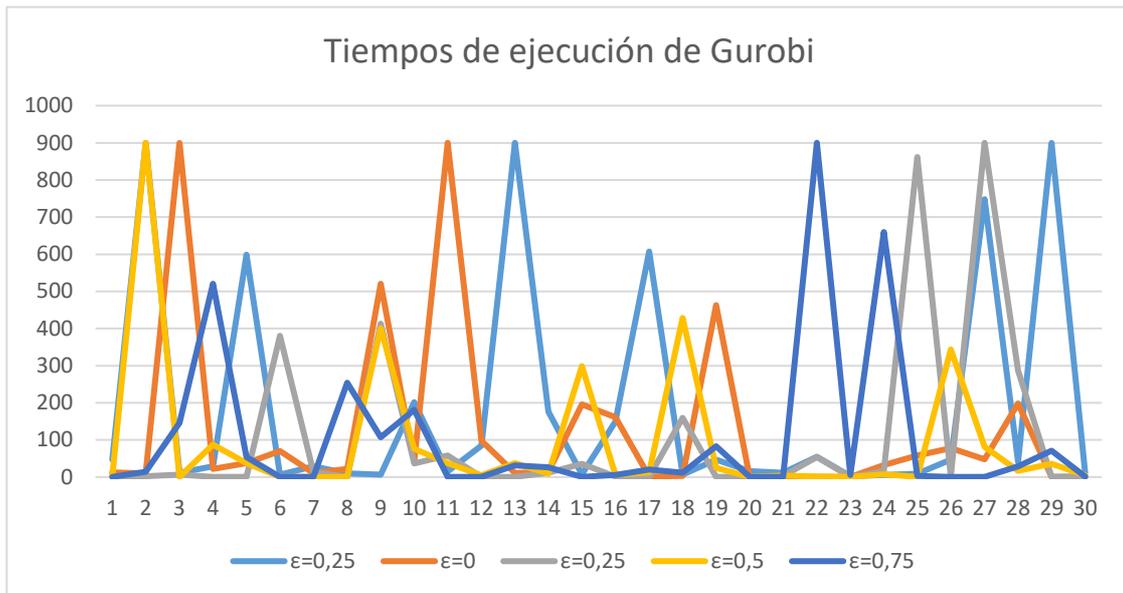


Figura 5.10 Tiempos de ejecución de Gurobi (s) para $n=30$.

Además de esto, es conveniente estimar la tardanza media para cada valor de n y compararla con la batería anterior para apreciar lo que sucede. En la *Figura 5.11* se evalúa la tardanza total para todos los trabajos de la batería.

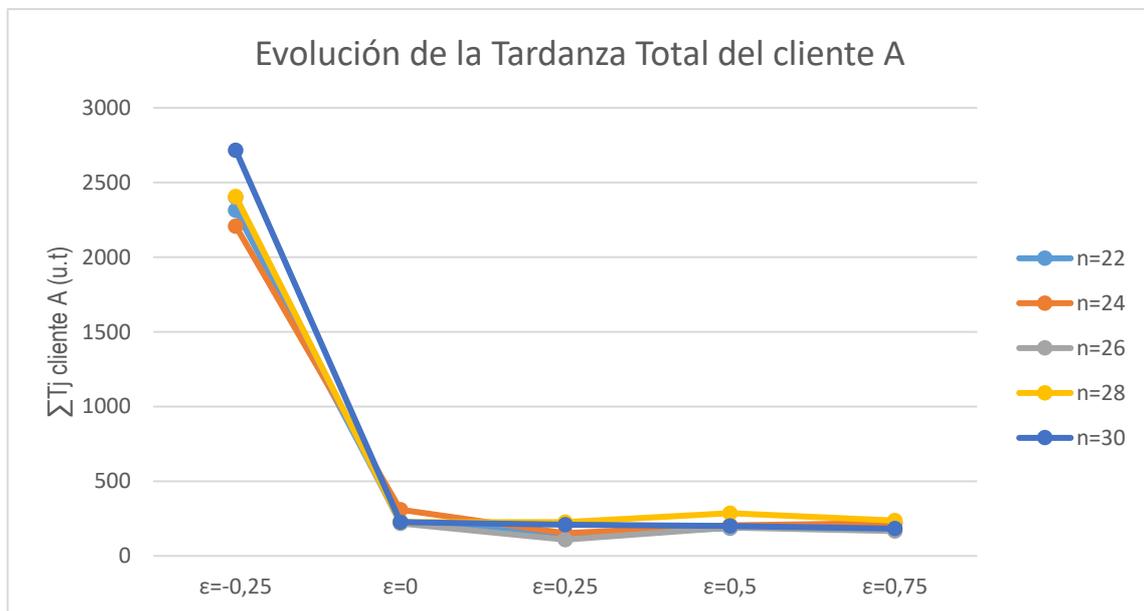


Figura 5.11 Comparativa de la tardanza total del cliente A (u.t). Batería mediana.

Al igual que en el caso anterior, se observa una tendencia nivelada de todos los trabajos para todos los valores de $\epsilon \geq 0$. De la misma manera que sucedía en la batería pequeña, los valores de la tardanza total del agente A se elevan de manera muy marcada para $\epsilon = -0,25$. Esto sucede porque los trabajos del primer cliente se deben ordenar más tarde, puesto que ahora la prioridad es reducir el tiempo de finalización máximo del agente B. Se disminuye el valor del *makespan* a costa de empeorar el valor de la tardanza total de A, principal objetivo del problema.

Por su parte, el valor del *makespan* del cliente B evoluciona de una manera muy similar a la batería anterior. Ahora se obtienen valores mayores, lo cual es coherente porque el número de trabajos a secuenciar es también más elevado, tal y como se detalla en la *Figura 5.12*.

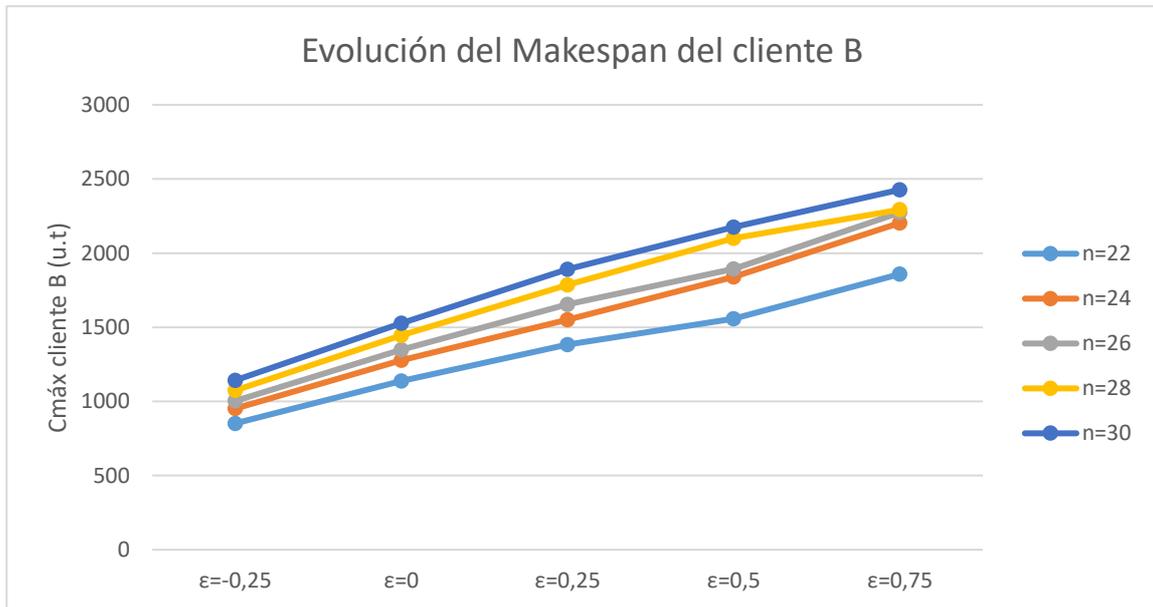


Figura 5.12 Comparativa del makespan del cliente B (u.t).

También ocurre lo mismo con Q , quien limita el crecimiento de $C_{máx}^B$. Ello se muestra en la *Figura 5.13*.

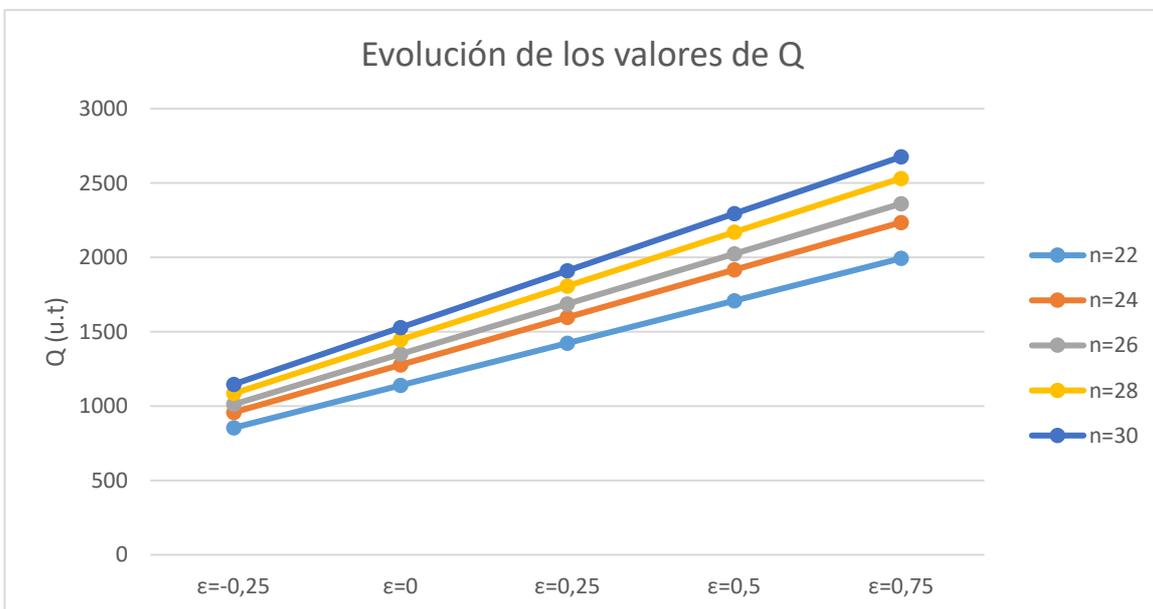


Figura 5.13 Comparativa de los valores de Q (u.t). Batería mediana.

Una ecuación que confirma de manera exitosa el cumplimiento del modelo es $C_{máx}^B \leq Q$, cuya curva para el tiempo máximo de finalización debe ajustarse o ser igual a la curva de los valores de Q para todos los valores de n . Véase *Figura 5.14*

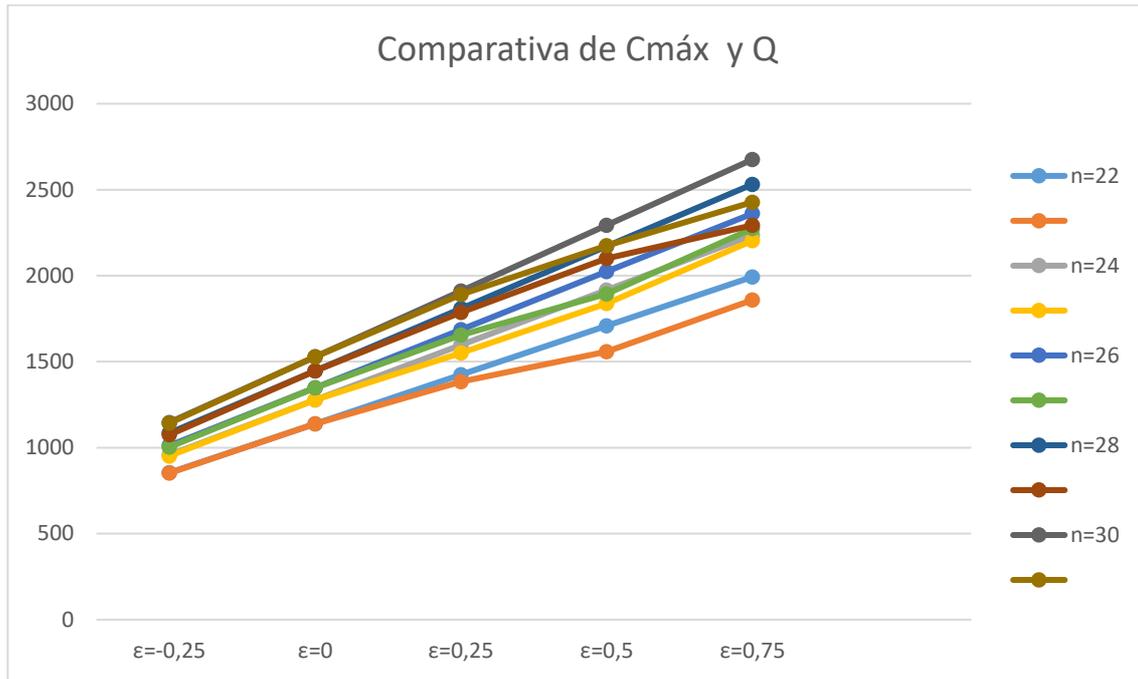


Figura 5.14 Comparativa entre Cmáx y Q (u.t). Batería mediana.

5.3 Batería grande

En este último caso, los tiempos computacionales se acentúan visiblemente en la *Figura 5.15*, apreciándose un notable aumento con respecto a la batería anterior y que, de manera directa, van a marcar el fin del estudio del problema de programación.

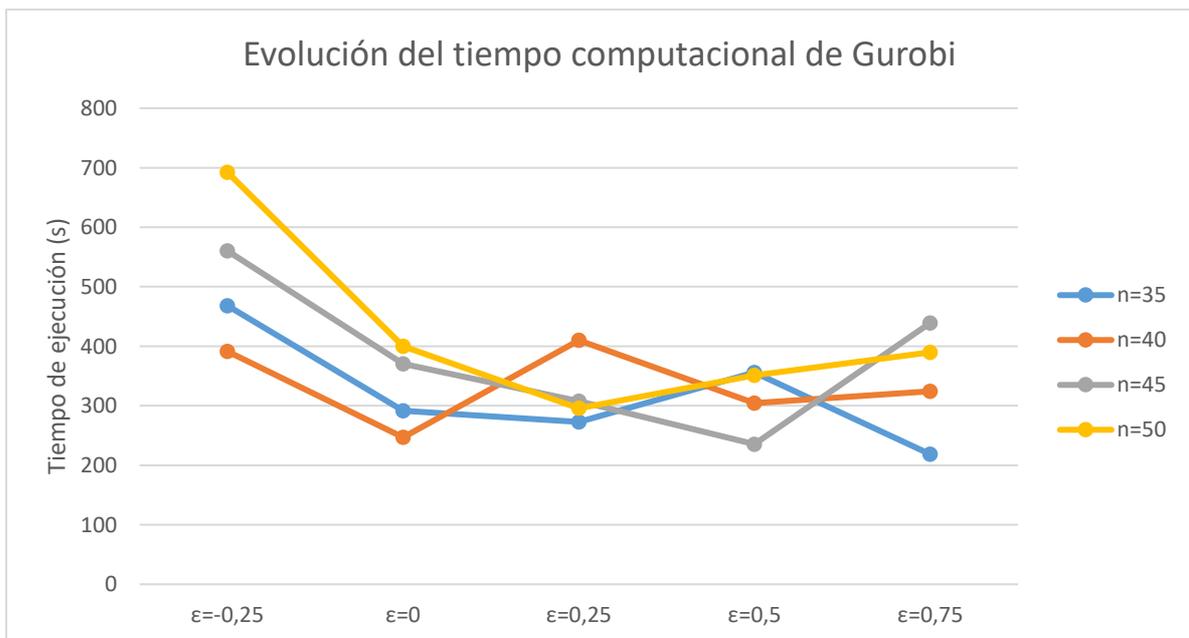


Figura 5.15 Tiempo de ejecución medio de Gurobi (s). Batería grande.

En esta última batería los tiempos medios llegan a alcanzar prácticamente los 700 segundos, lo que quiere decir

que existen muchos valores que exceden el tiempo límite establecido. Este hecho se aprecia en ya en el primer valor del número de trabajos, $n = 35$, en la Figura 5.16.

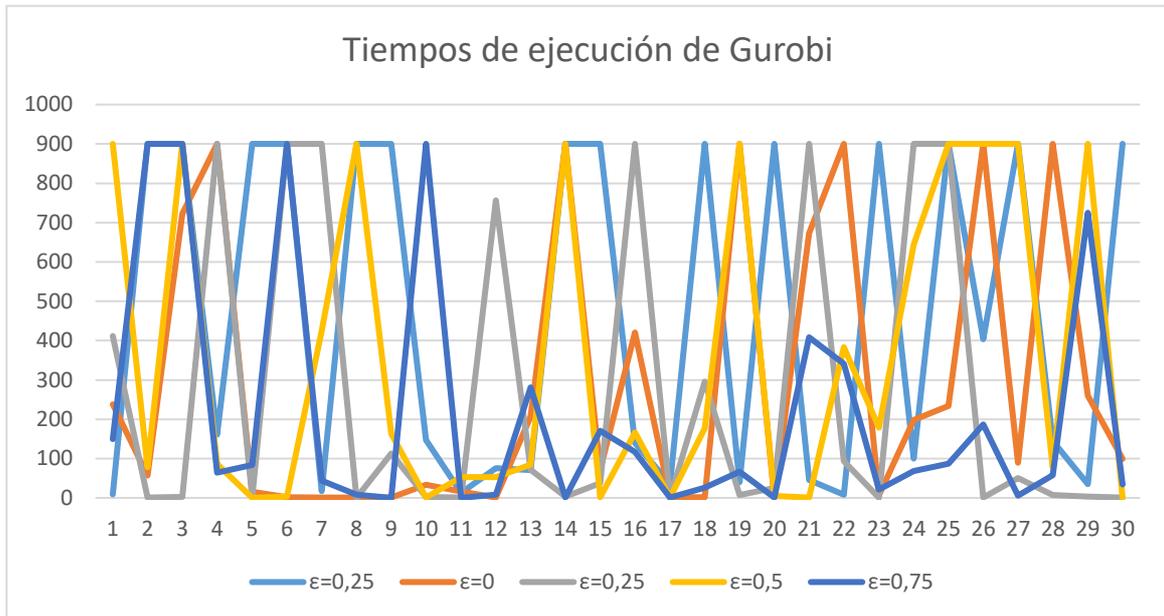


Figura 5.16 Tiempos de ejecución de Gurobi (s) para $n=35$.

Ello se acentúa de manera más evidente para $n = 40$ y $n = 45$, tal y como se aprecia en la Figura 5.17 y Figura 5.18 respectivamente.

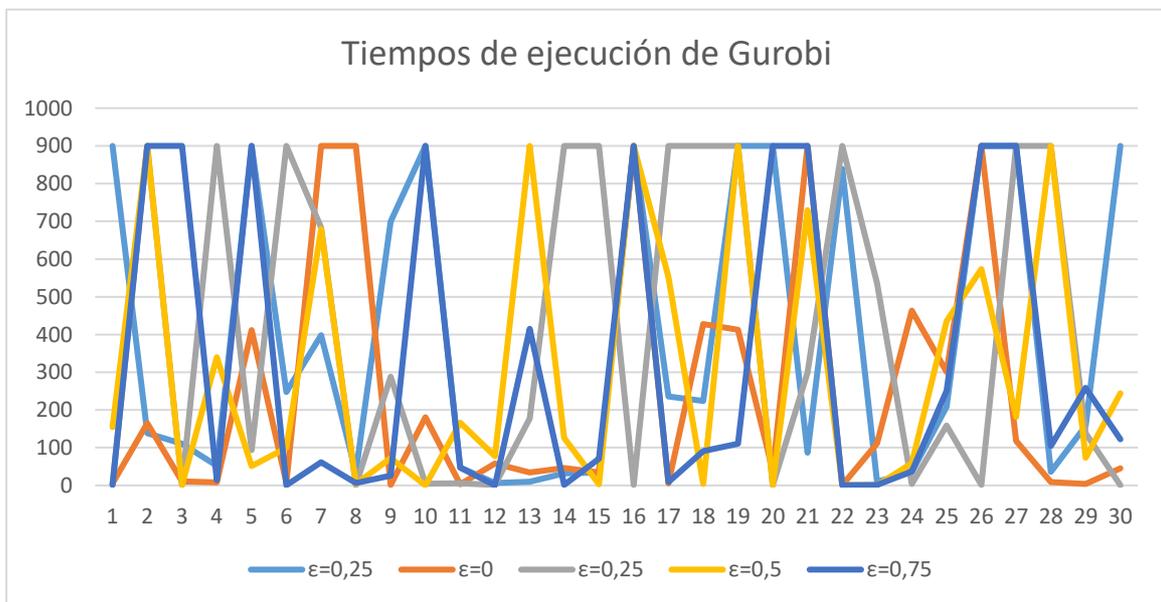


Figura 5.17 Tiempos de ejecución de Gurobi (s) para $n=40$.

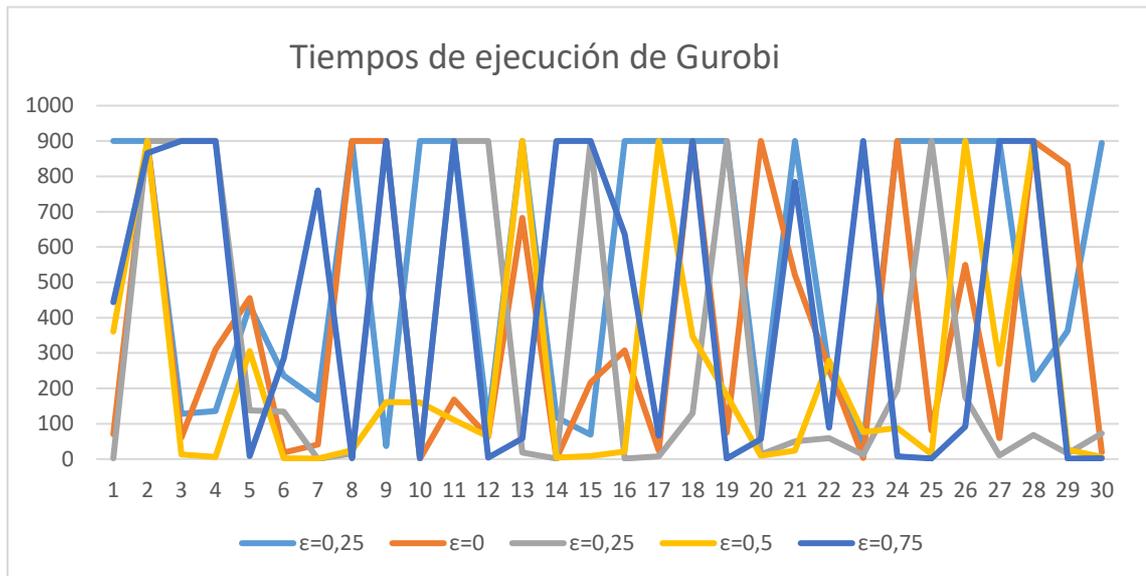


Figura 5.18 Tiempos de ejecución de Gurobi (s) para $n=45$.

Finalmente, los tiempos computacionales obtenidos para el último número de trabajos se observa en la *Figura 5.19*.

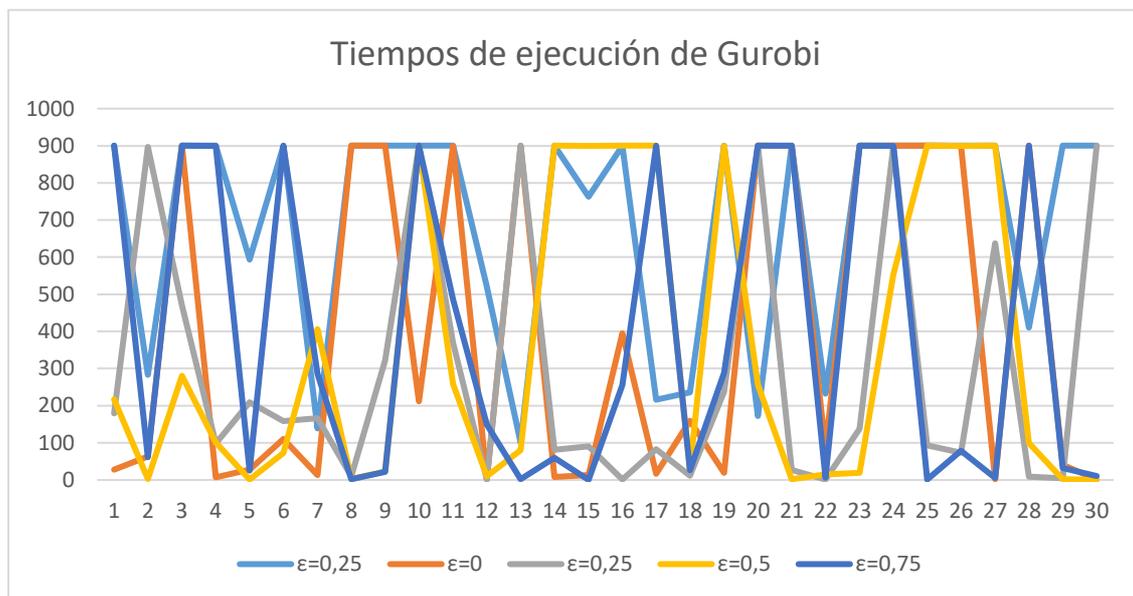


Figura 5.19 Tiempos de ejecución de Gurobi (s) para $n=50$.

Los tiempos de ejecución de Gurobi van creciendo para una cantidad mayor de instancias para cada valor de ϵ conforme se aumenta el número de trabajos. En esta última batería, los tiempos de ejecución empiezan a consolidarse en torno a un valor de 900 segundos, el cual se trata del límite máximo estipulado. Esto significa que Gurobi empieza a atascarse en cuanto a resolver problemas basados en modelos matemáticos de programación lineal entera mixta (MILP) para valores de $n \geq 50$, pues no consigue alcanzar el óptimo en ese tiempo de quince minutos. La decadencia de la optimalidad se refleja de manera exhaustiva en la *Figura 5.20*, la cual no excede prácticamente del 60% para muchos valores de ϵ en el caso de $n = 50$.

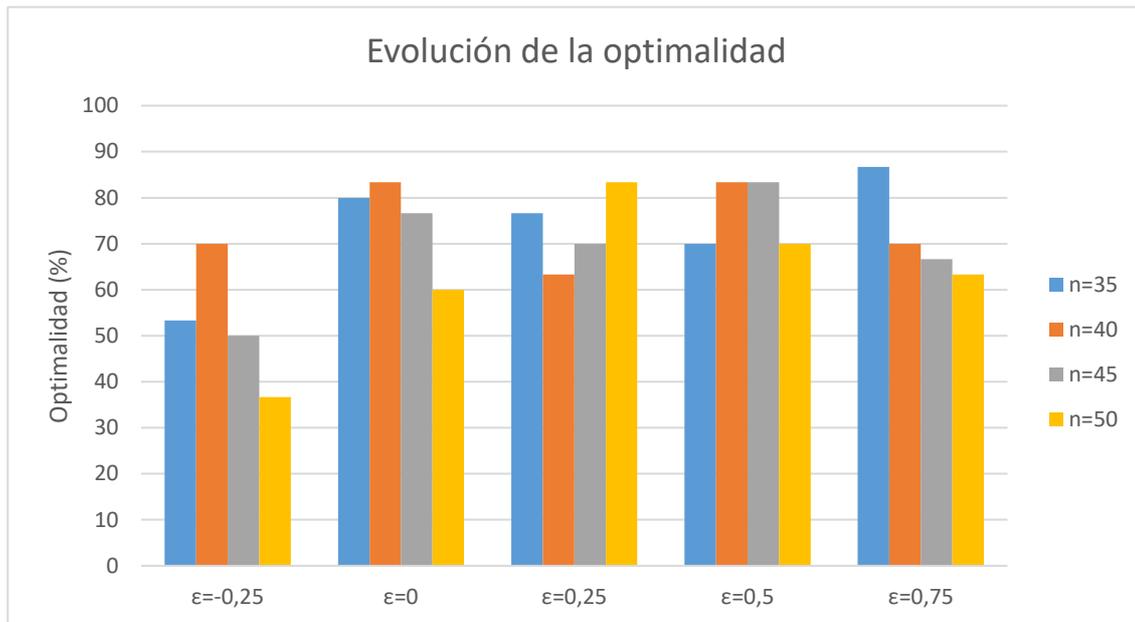


Figura 5.20 Comparativa de la optimalidad (%). Batería grande.

Como ya se ha mencionado con anterioridad, la optimalidad decae para valores de $\epsilon = -0.25$ puesto que el modelo se complica al reducir el valor de Q . Para el resto de los valores de ϵ , la optimalidad se reduce de una forma nivelada, al igual que sucedía en los dos casos anteriores; poniendo de manifiesto la complejidad del modelo con el crecimiento de n .

Por su parte, la tardanza total del agente A presenta un comportamiento muy similar a lo acontecido en las dos baterías precedentes. Ello se aprecia en la *Figura 5.21*. Se acentúa un crecimiento para el primer valor de ϵ y luego decae, con una nivelación para el resto de los valores de ϵ .

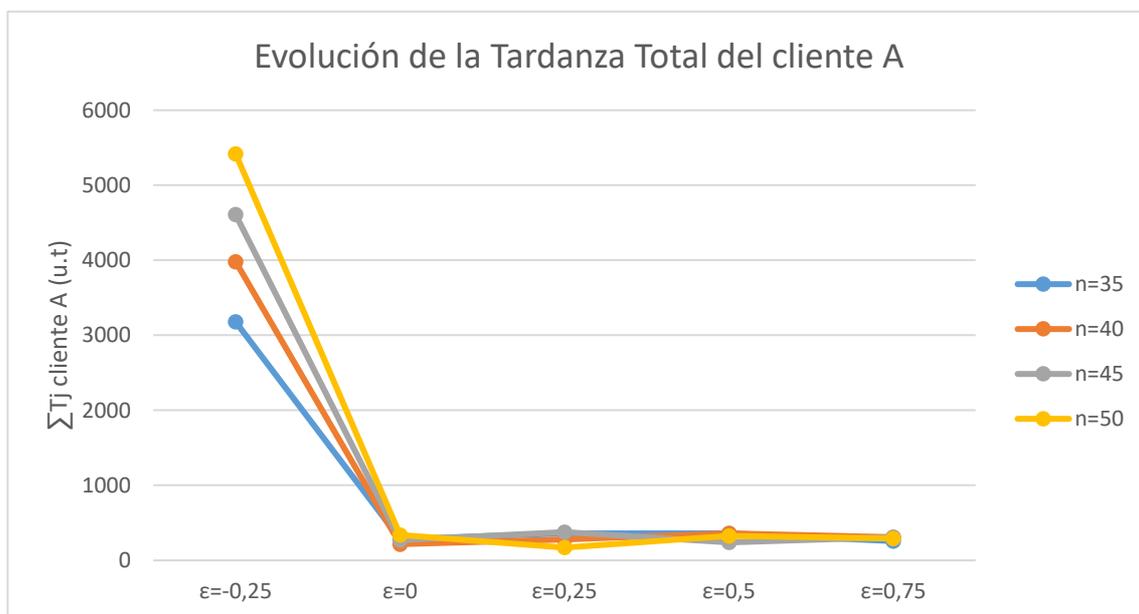


Figura 5.21 Comparativa de la tardanza total del cliente A (u.t.). Batería grande.

Por su parte, el *makespan* crece de manera prácticamente lineal para $n = 35, 40, 45, 50$, tal y como se percibe en la *Figura 5.22*.

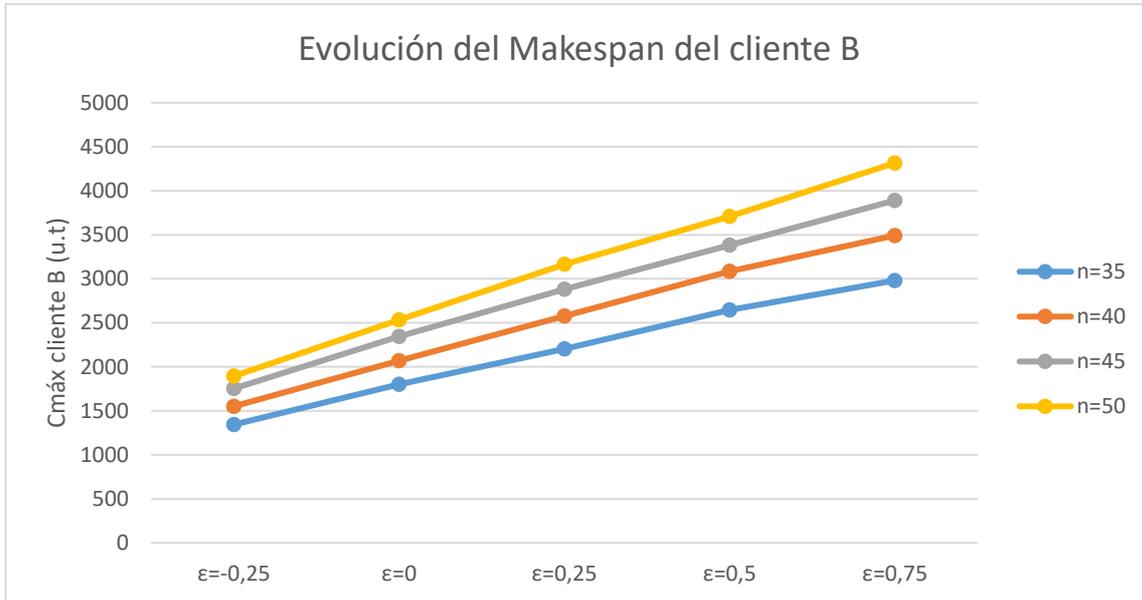


Figura 5.22 Comparativa del makespan del cliente B (u.t). Batería grande.

De la misma manera que en la batería anterior, los valores obtenidos de Q presentan el mismo comportamiento (véase *Figura 5.23*).

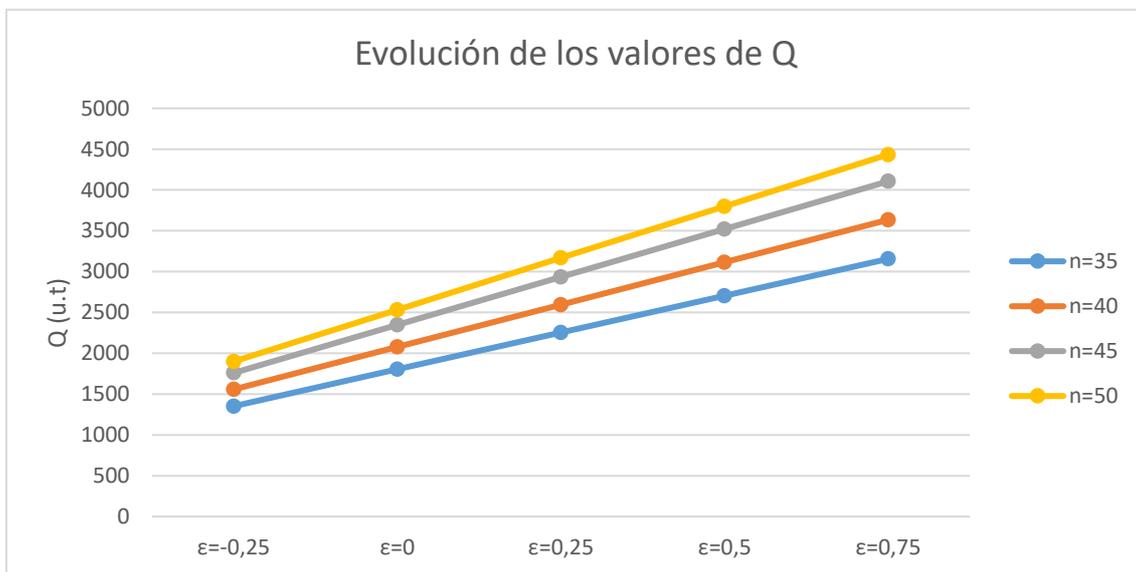


Figura 5.23 Comparativa de los valores de Q (u.t). Batería grande.

Dado que es obligatorio satisfacer la ecuación $C_{m\acute{a}x}^B \leq Q$ en todos los casos, la curva del tiempo de finalización máximo del agente B debe estar en todo momento por debajo o igual que la curva de los valores de Q . Este hecho se cumplía en las dos baterías anteriores y en esta también lo hace de manera exitosa. Esto se muestra en la *Figura 5.24*.

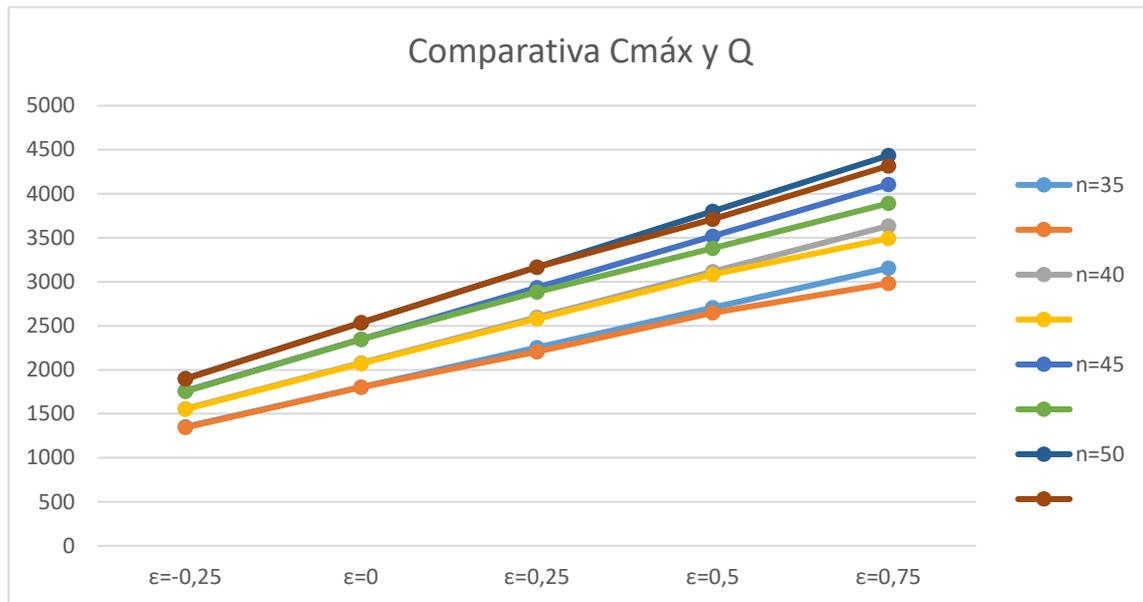


Figura 5.24 Comparativa entre $C_{máx}$ y Q (u.t). Batería grande.

5.4 Análisis de resultados entre las distintas baterías de problemas

Los análisis de los resultados obtenidos mediante gráficas permiten poner de manifiesto las consecuencias más directas que se pueden extraer de la resolución mediante Gurobi del problema $F2 \mid prmu \mid \sum T_j^A : C_{máx}^B \leq Q$. Un estudio general daría lugar a una serie de decisiones sobre la resolución del problema para los distintos valores de n . Por ello, se procede a comparar los efectos de cada una de las tres baterías.

En primer lugar, conviene resaltar la evolución que presenta el tiempo computacional de Gurobi conforme crece el número de trabajos. Como se mencionó en la sección 3.2, el espacio de soluciones crece de manera factorial al aumentar el número de trabajos. Esto supone que el rango en el que el optimizador de Gurobi ha de encontrar la secuencia óptima es cada vez mayor y más complejo. Por ello, los tiempos computacionales se hacen cada vez más grandes y Gurobi necesita de un mayor tiempo para la obtención de la mejor secuencia. En algunos casos, Gurobi no es capaz de encontrar el óptimo en un tiempo razonable, que para este presente Trabajo Fin de Grado, se ha considerado de quince minutos dado el número elevado de instancias para cada valor de n . En la Figura 5.25 se aprecia el desarrollo de los tiempos de ejecución.

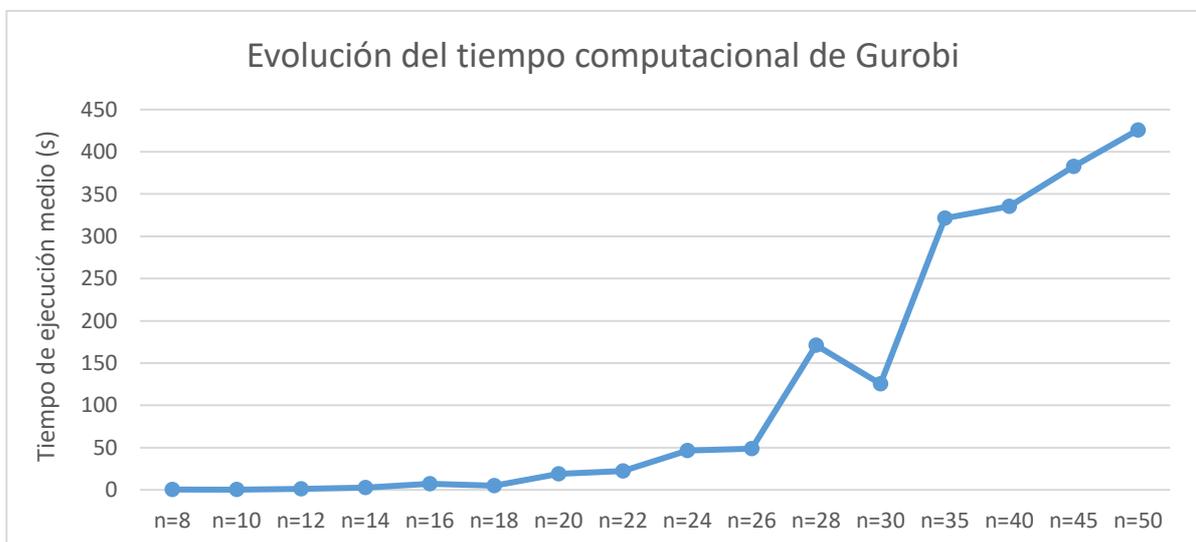


Figura 5.25 Evolución de los tiempos medios (s) de Gurobi.

Para $n = 50$, se observa que el tiempo computacional medio es prácticamente de 450 segundos, lo que significa que son muchas las instancias que requieren del tiempo máximo de Gurobi para su resolución, tal y como se observó en la sección (5.3). Es por ello, que se aprecia una considerable reducción de la optimalidad, lo que implica que la resolución mediante un modelo de programación lineal entera mixta (MILP) para el problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ comienza a ser poco eficiente (véase *Figura 5.26*) para valores de $n \geq 50$. Por este motivo, se podría afirmar que la resolución mediante un MILP del problema multiagente en un taller de flujo de dos máquinas es eficiente si $2 \leq n \leq 50$, siendo $N = \{1 \dots n\}$ y $N = n_A + n_B$.

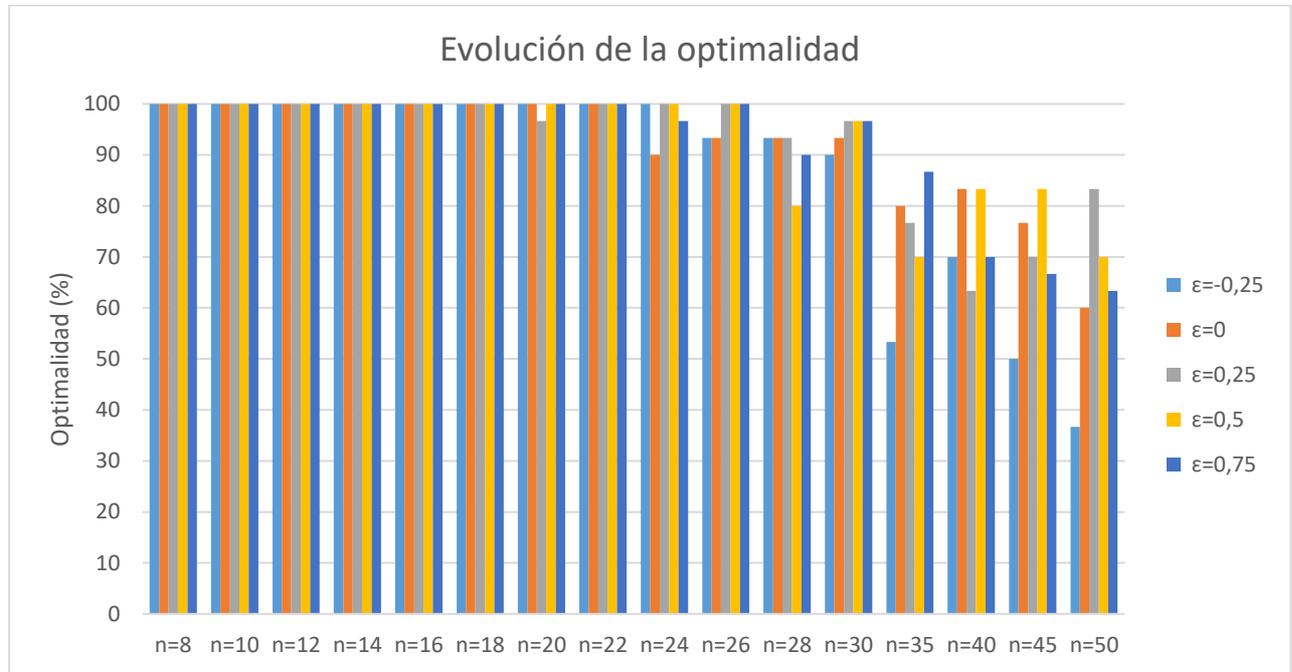


Figura 5.26 Análisis de la optimalidad (%) en función de ϵ .

Por su parte, la optimalidad media para cada valor de n , sin depender del valor de épsilon se refleja en la *Figura 5.27*. En ella, se pone de manifiesto la decadencia que presenta la resolución de manera óptima del problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ por parte del optimizador de Gurobi, la cual no supera el 65% para $n = 50$.

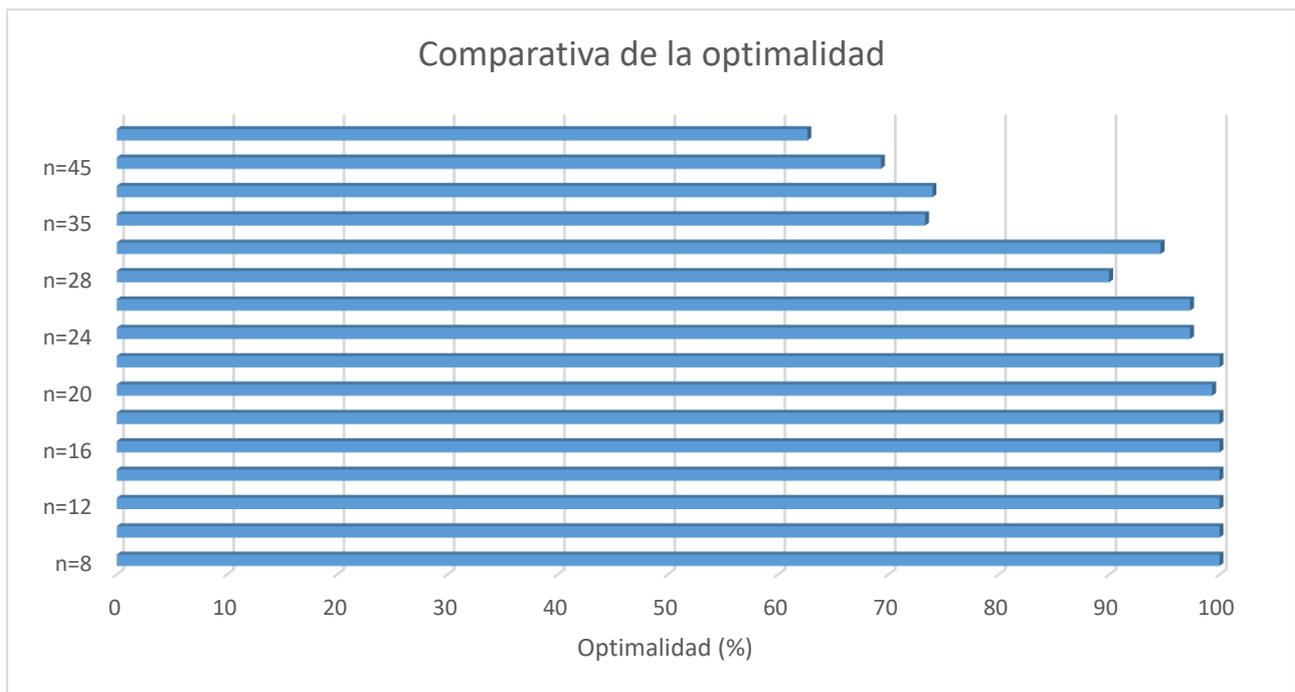


Figura 5.27 Análisis general de la optimalidad (%) de Gurobi.

En segundo lugar, la tendencia que sigue la evolución de la tardanza total del cliente A, cuya minimización era uno de los principales objetivos del proyecto, es creciente con el número de trabajos. Ello es consecuente con el crecimiento de n , pues al existir más trabajos que secuenciar también aumenta el valor total de la tardanza. Este hecho se aprecia en la *Figura 5.28*.

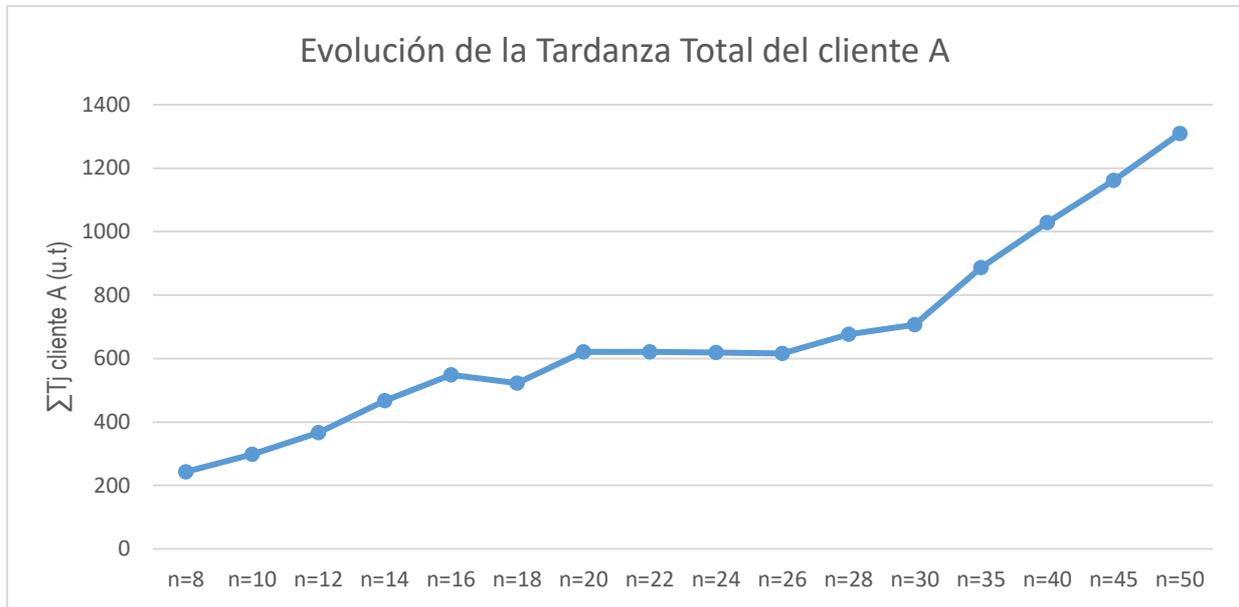


Figura 5.28 Análisis general de la tardanza total del cliente A (u.t).

En tercer y último lugar, la *Figura 5.29* representa la formalización de la restricción que impone el cliente B, introducida como una ecuación más dentro del modelo de programación lineal: $C_{máx}^B \leq Q$. El tiempo de finalización máximo de B debe ser en todo momento menor o igual al valor que tome Q . Esto pone de manifiesto que se satisface el modelo creado mediante la herramienta Matlab exitosamente para todos los valores de n .

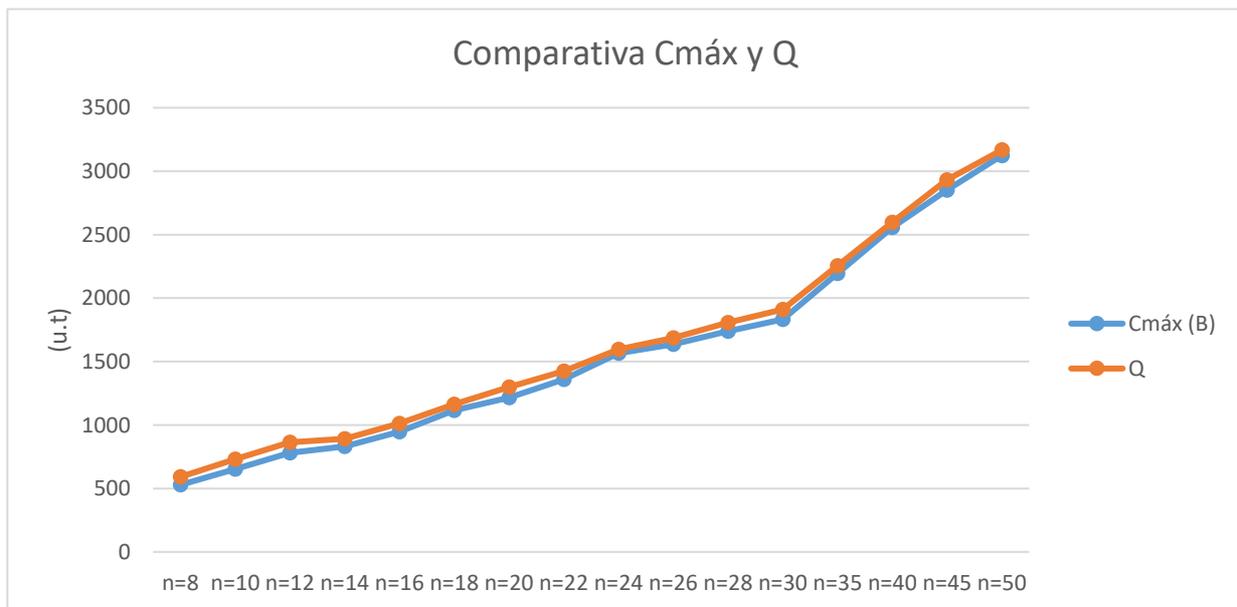


Figura 5.29 Análisis general de Cmáx y Q (u.t).

6 CONCLUSIONES

La secuenciación es un proceso basado en la toma de decisiones que desempeña un papel crucial en las empresas manufactureras. En la actualidad, existe un entorno muy competitivo que desencadena la lucha por un mercado común. Por ello, la necesidad de programar las operaciones de manera eficiente para aprovechar al máximo los recursos se ha convertido en el objetivo habitual para todas las empresas.

Por este requisito, se ha tratado un problema de secuenciación en un taller de flujo con dos máquinas y dos conjuntos de clientes distintos, en los que se pretendía minimizar la tardanza total del primer cliente condicionado a que el tiempo de finalización del segundo cliente debe ser el más pequeño posible. Este problema se ha plasmado como $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ y se ha modelado con un modelo de programación lineal entera mixta (MILP) a través de la herramienta de Matlab. Para su resolución, se han probado tres baterías de problemas y se han resuelto mediante el optimizador de Gurobi. Finalmente, se han obtenido una serie de resultados en la sección 4.5 y se han analizado posteriormente en el capítulo 5.

A partir de este estudio, se pueden extraer una serie de conclusiones basadas en los tiempos computacionales que ha tardado Gurobi en resolver las instancias, en el valor de la tardanza total del primer cliente (cliente A), el valor del tiempo de finalización del segundo cliente (cliente B), los valores obtenidos de Q en función de ϵ , y el ajuste de la curva de Q al *makespan* del cliente B.

En primer lugar, es conveniente resaltar los resultados obtenidos de los tiempos de ejecución de las baterías analizadas a través de Gurobi. Dado que un *Branch and Bound* es muy eficiente para resolver problemas de hasta 16 trabajos de tamaño en un tiempo coherente (Ahmadi-Darani, Moslehi and Reisi-Nafchi, 2018), el modelado del problema facilita su resolución pues se hace más eficiente, permitiendo resolver problemas de hasta 50 trabajos de tamaño. La complejidad radica en el modelado completo del problema, mientras que en el algoritmo *Branch and Bound* el principal obstáculo es encontrar un *Lower Bound* que permita minimizar el valor de la función objetivo. Dado que para $n \geq 50$ la resolución del problema mediante un MILP se ralentiza notablemente, quizá sería conveniente emplear un algoritmo *Branch and Bound* para ese tamaño y valores superiores ya que esa fórmula inteligente ayudaría a encontrar buenos valores entre un conjunto muy amplio. No ocurre así con el modelo matemático, pues ha de probar cada restricción del modelo, dando lugar a un tiempo de ejecución muy elevado.

Además, resulta interesante comentar un posible aumento del número de máquinas ante futuras investigaciones a este problema. Este hecho supondría alterar el modelo de programación matemática que se ha presentado en este documento, pues se ha estipulado para un número fijo de máquinas igual a dos. Ello complicaría mucho el modelado, incrementado de manera considerable el tiempo de ejecución de Gurobi, por lo que no ha sido objeto de estudio en este proyecto.

En segundo lugar, es imprescindible recalcar la satisfacción de la ecuación $C_{máx}^B \leq Q$, pues refleja el cumplimiento del modelo de programación matemática. Como cabía esperar, esta formalización se aprecia para cada uno de los valores del número de trabajos analizados en este proyecto, tal y como se evalúa en la *Figura 5.29* mostrada en la sección anterior 5.4.

En tercer lugar, es necesario explicar el aumento de la tardanza total del cliente A. Al existir más trabajos que secuenciar, aumenta la posibilidad de que el valor de la función objetivo crezca, incrementando su valor. Por su parte, el valor del *makespan* del cliente B y los valores de Q obtenidos crecen aproximadamente de forma lineal, aumentándose su valor ligeramente para cada aumento del número de trabajos. Esto es lógico puesto que ambos valores dependen del tiempo de finalización de la secuencia π , la cual aumenta al crecer n .

En definitiva, la tardanza total del cliente A, el tiempo de finalización del cliente B y los valores de Q aumentan con el número de trabajos. Por su parte, el modelo de programación lineal entera mixta (MILP) es útil para resolver el problema $F2 \mid \text{prmu} \mid \sum T_j^A : C_{máx}^B \leq Q$ para valores del número de trabajos inferiores o iguales a 50. Para $n \geq 50$, sería conveniente emplear un algoritmo *Branch and Bound* que ayude a eliminar una buena parte del amplio abanico de soluciones existentes, siendo ineficiente el empleo de un MILP.

BIBLIOGRAFÍA

- Ahmadi-Darani, M. H., Moslehi, G. and Reisi-Nafchi, M. (2018) 'A two-agent scheduling problem in a two-machine flowshop', *International Journal of Industrial Engineering Computations*.
- Cortez, A. (2004) 'Teoría de la complejidad computacional y teoría de la computabilidad', *Revista de investigación*.
- Flores, X. A. (2014) 'Problemas del Milenio: P vs NP'.
- Framiñán, José M; Carrasco, F. (2017) 'Autor: Fernando Carrasco Díaz Tutor: José Manuel Framiñán Torres'.
- Framiñán, J. M., Leisten, R. and Ruiz, R. (2014) *Manufacturing scheduling systems: An integrated view on models, methods and tools, Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*.
- García, J. (2010) 'Simulación y Optimización de los Procesos Químicos'.
- Garey, M. R. and Johnson, D. S. (1979) *A Guide to the Theory of NP-Completeness*.
- Pérez González, Paz; Framiñán Torres, José M; Fernández-Viagas, V. (2014) *Apuntes de la asignatura Programación y Control de la Producción. Universidad de Sevilla*.
- Pinedo, M. L. (2016) *Scheduling: Theory, algorithms, and systems, fifth edition, Scheduling: Theory, Algorithms, and Systems, Fifth Edition*.

ANEXO A

En este anexo se muestra el código del programa “FS.m” desarrollado en el lenguaje Matlab para el estudio y análisis del problema.

```
%Batería de problemas
salida= sprintf('salida.txt');
fichero_salida=fopen(salida,'w');
epsilon=[-0.75,-0.5,-0.25,0,0.25,0.5,0.75] ;
%epsilon=[0];
numepsilon=length(epsilon);
numeroprogramasfactibles=zeros(1,numepsilon);
eficiencia=zeros(1,numepsilon);
ef=zeros(1,numepsilon);
unos=100*ones(1,numepsilon);
for ep=1:numepsilon
clear model;
fprintf(fichero_salida,'EPSILON=%d \n',epsilon(ep));
fprintf(fichero_salida,\n');
fprintf(fichero_salida,\n');
numerotrabajos=[50];
cantidadtrabajos=length(numerotrabajos);
numeroinstancias=30;
numerototalinstancias=numeroinstancias*cantidadtrabajos;
contadorinstancias=0;
contadortrabajos=1;
for i=1:numerototalinstancias
%1)LECTURA DE INSTANCIA
entrada= sprintf('FS2PM_%d_%d.txt',numerotrabajos(contadortrabajos),contadorinstancias);
fichero_entrada=fopen(entrada,'r');
DATOS=fscanf(fichero_entrada,'%d');
Q=0;
M=1000000;
m=DATOS(1);
n=DATOS(2);
dim=length(DATOS);

if rem(n,2)==0
    CLIENTEA=[1:(n/2)];
    CLIENTEB=[(n/2+1):n];
else
    CLIENTEA=[1:(n/2+1)];
    CLIENTEB=[(n/2+1.5):n];
end
na=length(CLIENTEA);
nb=length(CLIENTEB);
a=zeros(1,n);
b=zeros(1,n);
d=zeros(1,n);
t=1;
l=1;
b1=1;
b2=0;

for i=3:(dim-1)
    if(b1==1 & b2==0);
```

```

a(t)=DATOS(i);
t=t+1;
b1=0;
b2=1;

elseif(b1==0 & b2==1 );
b(l)=DATOS(i);
l=l+1;
b1=1;
b2=0;
end

end
fclose(fichero_entrada);

%2)GENERACIÓN DE LOS Dj
R=0.5;
tau=0.75;
c=0;
elemento=0;
for j=1:n
    elemento=((a(j)+b(j))*n)/m;
    if (elemento>c)
        c=elemento;
    end
end
Cinf=c*(1-tau-R/2);
Csup=c*(1-tau+R/2);
for j=1:n
    d(j)=floor(random('unif',Cinf,Csup));
end

%3)GENERACIÓN DEL VALOR Q
pij=[a;b];
J1=[];
J2=[];
tiempos1=[];
tiempos2=[];

pi11=[];
pi21=[];
pi12=[];
pi22=[];

%%%%%%Ordenación Johnson CLIENTE A
cn1=1;
cn2=1;
for i=1:na
    if(pij(i,1)<=pij(i,2))
        J1(cn1)=i;
        tiempos1(cn1)=pij(i,1);
        cn1=cn1+1;
    else
        J2(cn2)=i;
        tiempos2(cn2)=pij(i,2);
        cn2=cn2+1;
    end
end

dimJ1=length(J1);
dimJ2=length(J2);

```

```

aux1=0;
aux2=0;

for i=1:(dimJ1-1)
    for j=1:(dimJ1-1)
        if tiempos1(j)>tiempos1(j+1)
            aux1=tiempos1(j+1);
            tiempos1(j+1)=tiempos1(j);
            tiempos1(j)=aux1;

            aux2=J1(j+1);
            J1(j+1)=J1(j);
            J1(j)=aux2;
        end

    end
end

for i=1:(dimJ2-1)
    for j=1:(dimJ2-1)
        if tiempos2(j)<tiempos2(j+1)
            aux1=tiempos2(j+1);
            tiempos2(j+1)=tiempos2(j);
            tiempos2(j)=aux1;

            aux2=J2(j+1);
            J2(j+1)=J2(j);
            J2(j)=aux2;
        end

    end
end

end
end
pi11=J1;
pi21=J2;
piA=[pi11 pi21];

%%% Ordenación Johnson CLIENTE B
cn1=1;
cn2=1;
J1=[];
J2=[];
for i=(na+1):(na+nb)
    if(pij(i,1)<=pij(i,2))
        J1(cn1)=i;
        tiempos1(cn1)=pij(i,1);
        cn1=cn1+1;
    else
        J2(cn2)=i;
        tiempos2(cn2)=pij(i,2);
        cn2=cn2+1;
    end
end

dimJ1=length(J1);
dimJ2=length(J2);
aux1=0;
aux2=0;

for i=1:(dimJ1-1)
    for j=1:(dimJ1-1)

```

```

    if tiempos1(j)>tiempos1(j+1)
        aux1=tiempos1(j+1);
        tiempos1(j+1)=tiempos1(j);
        tiempos1(j)=aux1;

        aux2=J1(j+1);
        J1(j+1)=J1(j);
        J1(j)=aux2;
    end

end

for i=1:(dimJ2-1)
    for j=1:(dimJ2-1)
        if tiempos2(j)<tiempos2(j+1)
            aux1=tiempos2(j+1);
            tiempos2(j+1)=tiempos2(j);
            tiempos2(j)=aux1;

            aux2=J2(j+1);
            J2(j+1)=J2(j);
            J2(j)=aux2;
        end

    end
end
pi12=J1;
pi22=J2;
piB=[pi12 pi22];
PI=[piA piB];

%%%%%% CÁLCULO DEL MAKESPAN
p1=zeros(1,n);
p2=zeros(1,n);
p1(1)=a(PI(1));
k=2;
for j=2:n
    p1(j)=p1(j-1)+a(PI(k));
    k=k+1;
end

p2(1)=p1(1)+b(PI(1));

for j=2:n
    if p2(j-1)> p1(j)
        p2(j)=p2(j-1)+b(PI(j));
    else
        p2(j)=p1(j)+b(PI(j));
    end
end
Q=(1+epsilon(ep))*p2(n);

```

%MODELO MATEMÁTICO

%%%%RESTRICCIONES

%%%%%%RESTRICCION 1

```

R1=zeros(n,n*n);
marcador=0;
for i=1:n
    cont=0;
    for j=(marcador+1):(n*n)
        if(cont<n)
            R1(i,j)=1;
            cont=cont+1;
            marcador=j;
        end
    end
end
AUX=zeros(n,3*n);
R1=[R1 AUX];

```

%%%%%%RESTRICCION 2

```

v=ones(n,1);
J=diag(v);
R2=[];
for i=1:n
    R2=[R2 , J];
end
AUX=zeros(n,3*n);
R2=[R2 AUX];

```

%%%%%%RESTRICCION 3

```

R3=zeros(1,n*n);
j=n*n+1;
R3(1,j)=1;
dim=n*n+3*n;
AUX=zeros(1,dim-j);
R3=[R3 AUX];

```

```

e=R2(1,:);
cont=1;
for i=1:(n*n)
    if e(i)==1
        e(i)=-a(cont);
        cont=cont+1;
    end
end
R3=R3+e;

```

%%%%%%RESTRICCION 4

```

AUX1=zeros(n-1,n*n);
v1=ones(1,n-1);
X1=-1*eye(n);
X2=diag(v1,1);
RE4=X1+X2;
RE4(n,:)=[];
AUX2=zeros(n-1,2*n);
R4=[AUX1 RE4 AUX2];
Y=R2;

```

```

Y(1,:)=[];
cont=1;
cuento=0;
for j=1:(n*n)
    for i=1:(n-1)
        if Y(i,j)==1
            Y(i,j)=-a(cont);
            cuento=cuento+1;
        end
        if(cuento==(n-1))
            cuento=0;
            cont=cont+1;
        end
    end
end
R4=R4+Y;

%%%%%%RESTRICCION 5
R5=zeros(1,n*n+n);
j=(n*n+n)+1;
R5(1,j)=1;
dim=n*n+3*n;
AUX=zeros(1,dim-j);
R5=[R5 AUX];
e=R2(1,:);
cont=1;
for i=1:(n*n)
    if e(i)==1
        e(i)=-a(cont)-b(cont);
        cont=cont+1;
    end
end
R5=R5+e;

%%%%%%RESTRICCION 6
AUX1=zeros(n-1,n*n+n);
v1=ones(1,n-1);
X1=-1*eye(n);
X2=diag(v1,1);
RE6=X1+X2;
RE6(n,:)=[];
AUX2=zeros(n-1,n);
R6=[AUX1 RE6 AUX2];
Y=R2;
Y(1,:)=[];
cont=1;
cuento=0;
for j=1:(n*n)
    for i=1:(n-1)
        if Y(i,j)==1
            Y(i,j)=-b(cont);
            cuento=cuento+1;
        end
        if(cuento==(n-1))
            cuento=0;
            cont=cont+1;
        end
    end
end
R6=R6+Y;

```

```

%%%%%%RESTRICCION 7
Y=R2;
Y(1,:)=[];
cont=1;
cuento=0;
for j=1:(n*n)
    for i=1:(n-1)
        if Y(i,j)==1
            Y(i,j)=-a(cont)-b(cont);
            cuento=cuento+1;
        end
        if(cuento==(n-1))
            cuento=0;
            cont=cont+1;
        end
    end
end
RE71=eye(n);
RE71(n,:)=[];
v1=ones(1,n-1);
RE72=diag(v1,1);
RE72(n,:)=[];
AUX1=zeros(n-1,n*n);
AUX2=zeros(n-1,n);
R7=[AUX1 -RE71 RE72 AUX2];
R7=R7+Y;

```

```

%%%%%%RESTRICCION 8
v=ones(n,1);
J=diag(v);
R8=[];
for i=1:na
    R8=[R8 , J];
end
AUX=zeros(n,n*nb+3*n);
R8=[R8 AUX];
cont=1;
cuento=0;
for j=1:(n*n)
    for i=1:n
        if R8(i,j)==1
            R8(i,j)=M-d(cont);
            cuento=cuento+1;
        end
        if(cuento==n)
            cuento=0;
            cont=cont+1;
        end
    end
end
v1=ones(1,n);
v2=-1*ones(1,n);
Y1=diag(v1);
Y2=diag(v2);
F=zeros(n,n*n+n);
RE8=[F Y1 Y2];
R8=R8+RE8;

```

```

%%%%%%RESTRICCION 9
M1=zeros(n,n*na);
M2=[];
J=M*eye(n,n);
for i=1:nb
    M2=[M2 , J];
end
M3=zeros(n,n);
M4=eye(n,n);
M5=zeros(n,n);
R9=[M1 M2 M3 M4 M5];

%%%%%%RESTRICCION 10
AUX=zeros(n,n*n);
R10=eye(n,n);
R10=[AUX R10 R10 R10];

RESTRICCIONES=[R1;R2;R3;R4;R5;R6;R7;R8;R9;R10];
dm=length(RESTRICCIONES);

model.A=sparse(RESTRICCIONES);

%%%%%%FUNCION OBJETIVO
FO=zeros(1,n*n+2*n);
AUX=ones(1,n);
FO=[FO AUX];

model.obj=FO;

%%%%%%OBJETIVO: MINIMIZAR LA SUMA DE LAS TARDANZAS
model.modelsense='Min';

%%%%%%TERMINO INDEPENDIENTE
B1=ones(1,n)';
B2=ones(1,n)';
B3=zeros(1,1);
B4=zeros(1,n-1)';
B5=zeros(1,1);
B6=zeros(1,n-1)';
B7=zeros(1,n-1)';
B8=M*ones(1,n)';
B9=(M+Q)*ones(1,n)';
B10=zeros(1,n)';
B=[B1;B2;B3;B4;B5;B6;B7;B8;B9;B10];

model.rhs=[B'];

%%%%%%SIGNO DE LAS RESTRICCIONES
for i=1:(2*n)
    model.sense(i)='=';
    marcador=i;
end
model.sense(marcador+1)='=';
for i=(marcador+2):(marcador+n)
    model.sense(i)='=';
    marcador=i;
end
model.sense(marcador+1)='=';

for i=(marcador+2):(marcador+n-1)

```

```

    model.sense(i)='>';
    marcador=i;

end
for i=(marcador+1):(marcador+n)
    model.sense(i)='>';
    marcador=i;
end
for i=(marcador+1):(marcador+n)
    model.sense(i)='<';
    marcador=i;
end
for i=(marcador+1):(marcador+n)
    model.sense(i)='<';
    marcador=i;
end
for i=(marcador+1):(marcador+n)
    model.sense(i)='>';
    marcador=i;
end

%%% %%% %%% %%% VARIABLES
for i=1:(n*n)
    model.vtype(i)='B';
    marcador=i;
end
for i=(marcador+1):(marcador+3*n)
    model.vtype(i)='C';
    marcador=i;
end

%4)RESOLUCIÓN DEL MODELO MATEMÁTICO
params.timelimit=900;
result=gurobi(model,params);
tipo=result.mipgap;
if tipo~=inf
    x=result.x';
    numeroprogramasfactibles(ep)=numeroprogramasfactibles(ep)+1;
%% CODIFICACION DE LA SECUENCIA OBTENIDA
MAT=[];
conteo=1;
for i=1:n
    for j=1:n
        MAT(i,j)=x(conteo);
        conteo=conteo+1;
    end
end
SECUENCIA=[];
col=0;
fil=0;
for i=1:n
    for j=1:n
        if MAT(i,j)==1
            col=j;
            fil=i;
            for k=1:n
                if(k==col)
                    SECUENCIA(col)=fil;
                end
            end
        end
    end
end
end
end

```

```

end

F=[];
ct=1;
for i=(n*n+1):(n*n+n)
    F(ct)=x(i);
    ct=ct+1;
    marcador=i;
end
C=[];
ct=1;
for i=(marcador+1):(marcador+n)
    C(ct)=x(i);
    ct=ct+1;
    marcador=i;
end
T=[];
ct=1;
for i=(marcador+1):(marcador+n)
    T(ct)=x(i);
    ct=ct+1;
end
disp(" ");
disp(" ");
disp(" ");
disp(" ");
disp("La secuencia de Johnson es: ");
disp(PI);
disp("La secuencia obtenida es: ");
disp(SECUENCIA);
disp("El tiempo de terminación de cada trabajo en M1 (el vector F) es: ");
disp(F);
disp("El tiempo de terminación de cada trabajo en M2 (el vector C) es: ");
disp(C);
trabajofinalB=0;
posiciontrabajofinalB=0;
for i=1:n
    for j=1:nb
        if SECUENCIA(i)==CLIENTEB(j)
            trabajofinalB=SECUENCIA(i);
            posiciontrabajofinalB=i;
        end
    end
end
CB=C(posiciontrabajofinalB);
disp("El vector Dj es: ");
disp(d);
disp("El vector de la tardanza de cada trabajo (el vector T) es: ");
disp(T);

disp("El valor de la función objetivo es: ");
disp(result.objval);
VALORFO=result.objval;

disp("El tiempo de ejecución ha sido: ");
disp(result.runtime);
TIEMPO=result.runtime;
if TIEMPO>=900
    ef(ep)=ef(ep)+1;
end

```

%5) ESCRITURA DE LOS VALORES EN ARCHIVO DE SALIDA

```

fprintf(fichero_salida,'FS2PM_%d_%d\n',numerotrabajos(contadortrabajos),contadorinstancias);
fprintf(fichero_salida,'Total tardiness de A: ');
fprintf(fichero_salida,'%d\n',round(VALORFO));
fprintf(fichero_salida,'Makespan de B: ');
fprintf(fichero_salida,'%d\n',round(CB));
fprintf(fichero_salida,'El valor de Q es: ');
fprintf(fichero_salida,'%d\n',round(Q));
fprintf(fichero_salida,'Tiempo de ejecución: ');
fprintf(fichero_salida,'%f\n',TIEMPO);
fprintf(fichero_salida,'Secuencia óptima: ');

```

```

SumTj(ep,contadorinstancias+1)=round(VALORFO);
makespan(ep,contadorinstancias+1)=round(CB);
tiemposgurobi(ep,contadorinstancias+1)=TIEMPO;
valoresQ(ep,contadorinstancias+1)=round(Q);

```

```

for i=1:n

```

```

    fprintf(fichero_salida,'%d\t',SECUENCIA(i));

```

```

end

```

```

fprintf(fichero_salida,'\n');

```

```

fprintf(fichero_salida,'\n');

```

```

contadorinstancias=contadorinstancias+1;

```

```

if (contadorinstancias==numeroinstancias)

```

```

    contadortrabajos=contadortrabajos+1;

```

```

    contadorinstancias=0;

```

```

end

```

```

else

```

```

fprintf(fichero_salida,'FS2PM_%d_%d\n',numerotrabajos(contadortrabajos),contadorinstancias);

```

```

    fprintf(fichero_salida,'Programa infactible\n');

```

```

    fprintf(fichero_salida,'\n');

```

```

    fprintf(fichero_salida,'\n');

```

```

    contadorinstancias=contadorinstancias+1;

```

```

if (contadorinstancias==numeroinstancias)

```

```

    contadortrabajos=contadortrabajos+1;

```

```

    contadorinstancias=0;

```

```

end

```

```

end

```

```

end

```

```

end

```

```

fclose(fichero_salida);

```

```

eficiencia=unos-(100/30)*ef;

```


ANEXO B

- Batería pequeña**

n=8	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	100
0	0,22	0,24	0,35	0,16	0,26	609	270	142	27	166	290	396	407	594	693	297	396	495	594	693	0	100
1	0,14	0,20	0,24	0,18	0,23	397	101	228	29	421	343	477	534	534	834	358	477	596	716	835	0,25	100
2	0,18	0,36	0,20	0,17	0,17	303	242	68	159	79	376	505	631	757	883	379	505	631	758	884	0,5	100
3	0,26	0,34	0,20	0,29	0,11	652	354	40	460	0	383	526	562	563	508	395	526	658	789	921	0,75	100
4	0,23	0,23	0,09	0,16	0,14	573	67	0	83	69	379	512	640	768	550	384	512	640	768	896		
5	0,40	0,33	0,18	0,29	0,25	756	385	118	427	65	284	390	487	585	682	293	390	488	585	683		
6	0,27	0,22	0,24	0,08	0,12	398	50	109	0	11	230	308	334	363	436	231	308	385	462	539		
7	0,32	0,32	0,25	0,37	0,36	408	96	136	108	131	392	510	678	814	605	407	543	679	815	950		
8	0,29	0,26	0,33	0,29	0,30	921	140	305	94	345	403	551	688	826	964	413	551	689	827	964		
9	0,19	0,36	0,14	0,41	0,29	677	140	0	127	468	357	476	595	477	489	357	476	595	714	833		
10	0,24	0,38	0,52	0,20	0,20	389	346	345	4	171	337	474	592	711	518	356	474	593	711	830		
11	0,23	0,14	0,17	0,11	0,33	295	11	176	0	148	361	488	610	508	854	366	488	610	732	854		
12	0,23	0,32	0,13	0,28	0,10	424	151	17	261	120	358	480	521	720	840	360	480	600	720	840		
13	0,30	0,14	0,22	0,11	0,24	668	33	155	24	30	411	548	593	620	648	411	548	685	822	959		
14	0,36	0,26	0,36	0,22	0,32	844	373	115	76	52	365	535	668	554	936	401	535	669	803	936		
15	0,15	0,18	0,19	0,15	0,23	649	214	71	111	228	447	596	609	611	1043	447	596	745	894	1043		
16	0,14	0,18	0,16	0,20	0,20	340	147	10	245	376	332	491	523	736	859	368	491	614	737	859		
17	0,21	0,20	0,19	0,10	0,16	648	166	103	0	59	325	453	566	492	792	340	453	566	680	793		
18	0,21	0,17	0,17	0,183	0,19	684	188	59	67	86	343	459	509	505	803	344	459	574	689	803		
19	0,25	0,35	0,39	0,29	0,25	627	100	224	339	145	313	424	530	501	742	318	424	530	636	742		
20	0,23	0,17	0,30	0,29	0,20	655	354	325	31	116	327	437	546	655	439	328	437	546	656	765		
21	0,16	0,19	0,31	0,27	0,19	1015	79	243	410	143	411	548	685	822	959	411	548	685	822	959		
22	0,16	0,30	0,33	0,13	0,28	591	238	318	0	69	354	472	495	528	826	354	472	590	708	826		
23	0,32	0,30	0,37	0,40	0,41	410	249	151	250	205	343	458	572	483	483	344	458	573	687	802		
24	0,25	0,20	0,31	0,11	0,36	909	189	224	0	298	371	495	548	522	496	371	495	619	743	866		
25	0,22	0,20	0,18	0,18	0,26	454	279	30	51	399	328	453	466	453	453	340	453	566	680	793		
26	0,39	0,37	0,38	0,21	0,29	800	387	111	99	288	277	370	462	555	647	278	370	463	555	648		
27	0,30	0,20	0,35	0,16	0,17	552	118	174	7	110	302	403	503	419	471	302	403	504	605	705		
28	0,37	0,18	0,183	0,10	0,11	496	71	112	0	0	284	396	495	403	496	297	396	495	594	693		
29	0,31	0,14	0,33	0,26	0,31	650	26	235	240	240	388	520	533	533	533	390	520	650	780	910		
Promedio	0,25	0,25	0,26	0,21	0,23	593	185	144	124	167	347	471	552	587	682	354	472	591	709	827		

n=10	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	100
0	0,09	0,17	0,39	0,07	0,11	859	67	175	0	9	426	569	628	665	632	427	569	711	854	996	0	100
1	0,15	0,07	0,14	0,14	0,2	954	0	100	82	83	373	530	662	607	927	398	530	663	795	928	0,25	100
2	0,15	0,12	0,18	0,28	0,2	1177	80	109	462	216	540	720	900	784	1260	540	720	900	1080	1260	0,5	100
3	0,15	0,20	0,18	0,07	0,09	174	65	128	0	93	342	451	541	488	483	350	466	583	699	816	0,75	100
4	0,35	0,20	0,12	0,45	0,26	727	109	5	383	225	388	592	587	888	598	444	592	740	888	1036		
5	0,12	0,18	0,812	0,18	0,09	864	194	276	143	0	471	628	785	942	710	471	628	785	942	1099		
6	0,09	0,20	0,17	0,14	0,18	978	148	101	49	120	429	602	752	636	1053	452	602	753	903	1054		
7	0,12	0,14	0,24	0,10	0,28	689	264	147	75	86	346	496	512	744	496	372	496	620	744	868		
8	0,15	0,18	0,51	0,28	0,06	468	149	133	202	50	297	399	445	419	698	299	399	499	599	698		
9	0,26	0,20	0,51	0,56	0,85	965	93	852	521	527	513	685	720	1027	729	514	685	856	1028	1199		
10	0,09	0,10	0,07	0,23	0,20	1064	51	0	211	222	446	627	683	940	683	470	627	784	941	1097		
11	0,10	0,26	0,28	0,29	0,32	780	243	397	446	301	472	632	677	648	1106	474	632	790	948	1106		
12	0,06	0,14	0,31	0,21	0,22	332	305	574	22	68	414	562	626	575	608	422	562	703	843	984		
13	0,21	0,20	0,24	0,17	0,20	815	189	355	61	327	403	538	547	807	941	404	538	673	807	942		
14	0,15	0,03	0,10	0,51	0,17	664	0	0	175	11	466	649	811	694	721	487	649	811	974	1136		
15	0,17	0,28	0,15	0,09	0,10	753	539	96	0	18	449	599	748	643	638	449	599	749	899	1048		
16	0,10	0,34	0,14	0,12	0,10	995	209	45	85	66	490	680	850	1020	1190	510	680	850	1020	1190		
17	0,15	0,43	1,01	0,59	0,62	571	106	287	221	438	371	553	691	829	967	415	553	691	830	968		
18	0,10	0,17	0,09	0,15	0,35	598	205	52	95	411	438	597	678	895	1044	448	597	746	896	1045		
19	0,06	0,32	0,12	0,12	0,26	757	183	113	47	60	388	532	665	543	608	399	532	665	798	931		
20	0,18	0,89	0,70	0,20	0,51	317	400	332	119	238	416	555	693	555	574	416	555	694	833	971		
21	0,09	0,26	0,20	0,26	0,21	1117	252	126	269	380	477	691	735	1036	780	518	691	864	1037	1209		
22	0,07	0,23	0,23	0,42	0,35	364	217	31	430	539	421	562	579	568	571	422	562	703	843	984		
23	0,31	0,29	0,17	0,21	0,23	963	513	123	538	186	533	732	915	1098	1281	549	732	915	1098	1281		
24	0,17	0,32	0,04	0,21	0,21	801	190	0	65	132	432	576	620	576	1008	432	576	720	864	1008		
25	0,15	0,23	0,42	0,55	0,04	663	120	87	100	0	353	496	620	744	643	372	496	620	744	868		
26	0,24	0,28	0,28	0,15	1,21	540	85	15	5	242	311	422	527	633	738	317	422	528	633	739		
27	0,12	0,14	0,20	0,21	0,10	825	232	299	185	38	421	594	742	891	1039	446	594	743	891	1040		
28	0,18	0,17	0,67	0,23	0,14	939	233	158	99	114	435	596	745	894	1043	447	596	745	894	1043		
29	0,17	0,24	0,15	0,31	0,29	838	645	40	529	151	470	631	788	664	1104	473	631	789	947	1104		
Promedio	0,15	0,23	0,29	0,25	0,27	751	202	171	187	178	424	583	682	748	829	437	583	729	875	1021		

n=12	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	100
0	0,26	1,70	1,39	0,28	2,03	427	154	228	44	420	461	615	768	668	1076	461	615	769	923	1076	0	100
1	0,32	0,50	0,25	1,26	0,34	877	328	36	372	440	532	748	765	1122	1309	561	748	935	1122	1309	0,25	100
2	0,45	1,82	0,77	0,40	0,32	894	300	86	62	27	396	529	581	793	601	397	529	661	794	926	0,5	100
3	0,40	0,30	0,39	1,24	1,31	1051	96	175	323	218	504	672	694	1008	1176	504	672	840	1008	1176	0,75	100
4	0,53	0,30	0,74	1,23	0,33	1606	79	349	169	69	526	722	751	1083	752	542	722	903	1083	1264		
5	0,35	1,08	1,00	1,24	0,15	812	225	86	164	0	439	608	760	912	608	456	608	760	912	1064		
6	0,24	0,59	1,49	2,26	0,70	495	14	283	235	219	435	589	589	883	1030	442	589	736	884	1031		
7	0,48	0,33	1,83	1,07	0,30	1487	184	157	144	129	536	779	973	1168	832	584	779	974	1169	1363		
8	0,38	0,51	0,34	0,60	0,26	1201	201	233	15	0	497	663	755	994	745	497	663	829	995	1160		
9	0,64	7,17	0,26	1,02	1,28	1095	200	0	474	142	510	690	730	694	749	518	690	863	1035	1208		
10	0,77	1,43	0,44	1,21	3,48	741	175	32	310	754	490	654	817	981	1144	491	654	818	981	1145		
11	0,20	3,11	0,37	0,15	0,18	1191	98	495	0	0	484	742	927	771	1186	557	742	928	1113	1299		
12	0,69	4,47	1,93	1,25	6,73	1092	481	378	130	793	563	754	942	1131	1319	566	754	943	1131	1320		
13	0,97	2,28	5,31	1,35	0,95	1440	710	442	275	193	624	834	855	1251	1459	626	834	1043	1251	1460		
14	0,32	2,97	1,52	0,70	1,12	1120	349	214	114	446	531	724	905	724	1267	543	724	905	1086	1267		
15	0,36	0,53	0,13	0,37	0,21	1204	183	11	126	0	532	732	809	1098	732	549	732	915	1098	1281		
16	0,49	0,21	0,09	0,23	0,33	1059	0	0	28	187	442	595	621	658	1041	446	595	744	893	1041		
17	0,55	0,64	1,30	0,29	2,68	613	163	311	0	675	462	618	660	646	654	464	618	773	927	1082		
18	0,44	1,09	1,09	0,92	1,20	985	355	369	480	231	516	688	860	794	731	516	688	860	1032	1204		
19	0,36	0,10	0,10	0,13	0,16	920	0	11	0	2	476	635	654	660	635	476	635	794	953	1111		
20	0,32	0,13	1,19	1,78	0,96	1373	0	110	37	201	568	761	951	803	1331	571	761	951	1142	1332		
21	0,37	0,30	0,23	1,15	0,84	1455	47	51	383	456	591	823	838	1234	1440	617	823	1029	1235	1440		
22	0,38	0,78	2,41	0,73	0,09	1233	231	250	160	0	463	622	777	682	707	467	622	778	933	1089		
23	0,69	2,60	0,07	1,28	0,41	712	63	0	90	36	495	661	666	991	704	496	661	826	992	1157		
24	0,66	1,06	0,14	0,27	0,13	822	172	0	14	0	388	533	598	799	540	400	533	666	800	933		
25	0,38	0,82	0,45	1,96	0,26	793	92	177	251	0	490	661	692	991	1156	496	661	826	992	1157		
26	0,78	1,74	2,44	0,55	0,80	1793	417	807	289	85	561	749	849	1123	1310	562	749	936	1124	1311		
27	0,42	0,97	1,19	0,31	2,14	1053	267	223	63	342	488	657	821	657	1149	493	657	821	986	1150		
28	0,81	8,88	0,21	0,41	0,08	653	239	0	255	0	637	874	946	1311	1529	656	874	1093	1311	1530		
29	0,41	0,49	0,97	0,48	0,48	1228	83	712	307	10	532	782	971	815	1368	587	782	978	1173	1369		
Promedio	0,48	1,63	1,00	0,87	1,00	1047	196	207	177	202	505	690	784	914	1009	518	690	863	1035	1208		

n=14	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
	Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25
0	0,84	12,43	3,59	5,61	1,20	586	405	443	475	222	637	867	993	1300	867	650	867	1084	1301	1517	0	100
1	0,91	2,60	5,75	4,96	2,03	1723	221	38	477	664	523	698	735	1047	774	524	698	873	1047	1222	0,25	100
2	1,68	3,28	6,57	3,14	1,30	1523	246	368	87	138	511	698	872	766	1221	524	698	873	1047	1222	0,5	100
3	0,86	0,99	10,82	9,07	5,82	1685	145	359	610	803	523	698	766	767	737	524	698	873	1047	1222	0,75	100
4	0,36	3,96	0,68	0,23	2,97	1160	648	353	7	459	523	698	872	1047	1221	524	698	873	1047	1222		
5	0,52	3,64	0,35	11,53	1,90	972	464	0	458	313	523	698	872	1047	1221	524	698	873	1047	1222		
6	1,30	2,79	3,40	0,68	2,44	2072	384	248	0	74	523	698	872	1047	1221	524	698	873	1047	1222		
7	0,77	0,87	3,48	5,41	1,52	1738	277	26	174	617	523	698	746	780	708	524	698	873	1047	1222		
8	0,74	1,80	2,99	6,28	0,21	1474	88	37	351	0	519	698	718	750	1221	524	698	873	1047	1222		
9	0,45	4,02	4,07	4,12	1,85	1182	606	232	615	481	497	698	872	1047	1221	524	698	873	1047	1222		
10	0,82	3,23	4,75	2,27	0,74	2047	128	9	369	231	497	698	872	1047	727	524	698	873	1047	1222		
11	2,33	0,65	3,81	17,55	1,17	1294	18	305	481	303	585	805	1006	1208	1408	604	805	1006	1208	1409		
12	0,61	4,35	0,28	1,92	0,12	839	89	8	149	0	636	859	964	1288	1140	644	859	1074	1289	1503		
13	0,63	1,98	0,34	0,18	1,05	1018	33	38	0	34	636	859	944	1288	879	644	859	1074	1289	1503		
14	1,01	5,19	2,19	3,35	1,10	1788	448	486	489	331	523	698	775	1047	1221	524	698	873	1047	1222		
15	0,67	2,98	3,07	2,17	1,62	1980	205	611	114	233	523	698	872	1047	1221	524	698	873	1047	1222		
16	0,60	0,14	7,39	2,92	1,05	784	54	292	153	361	498	665	831	997	1163	499	665	831	998	1164		
17	0,89	2,62	9,70	1,37	4,19	1674	351	550	95	351	499	698	872	1047	1221	524	698	873	1047	1222		
18	0,50	5,08	0,43	0,24	1,70	1429	548	71	5	150	523	698	872	1047	1221	524	698	873	1047	1222		
19	1,65	6,29	3,60	0,26	1,29	1199	351	354	0	404	515	714	892	881	753	536	714	893	1071	1250		
20	0,66	6,27	1,82	2,41	0,79	681	140	252	261	177	567	772	965	1158	1351	579	772	965	1158	1351		
21	0,68	12,28	3,70	6,21	2,48	698	680	143	171	218	396	528	660	792	924	396	528	660	792	924		
22	1,30	4,12	0,07	6,75	0,54	911	23	0	626	108	488	702	734	886	783	529	705	881	1058	1234		
23	1,02	3,97	1,62	2,66	1,47	1901	586	291	270	431	518	698	774	1047	1221	524	698	873	1047	1222		
24	0,95	0,12	9,03	0,97	2,38	1472	0	148	190	15	510	698	872	1047	718	524	698	873	1047	1222		
25	0,90	0,24	0,80	5,35	2,23	1592	0	42	313	81	523	698	774	782	1221	524	698	873	1047	1222		
26	1,13	3,16	1,21	1,83	0,95	1665	356	222	176	253	515	698	766	1047	1221	524	698	873	1047	1222		
27	0,80	3,46	3,05	2,07	0,17	1746	377	19	53	5	513	698	872	951	723	524	698	873	1047	1222		
28	0,76	4,05	0,34	0,35	0,65	1273	160	0	35	108	567	781	873	1171	1366	586	781	976	1172	1367		
29	0,77	6,43	3,65	1,22	1,75	285	263	432	32	248	427	595	615	892	1041	446	595	744	893	1041		
Promedio	0,90	3,77	3,42	3,77	1,62	1346	276	212	241	260	525	713	837	1008	1064	535	713	892	1070	1249		

n=16	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	100
0	4,60	5,53	7,51	4,55	0,33	791	74	290	242	37	597	760	1002	1203	792	602	802	1003	1203	1404	0	100
1	1,68	4,36	7,24	37,64	5,03	1599	129	320	734	578	594	793	793	1190	1387	595	793	991	1190	1388	0,25	100
2	2,36	1,55	23,85	3,94	6,91	1730	124	641	670	149	594	793	991	1189	1387	595	793	991	1190	1388	0,5	100
3	1,78	10,64	60,47	5,87	4,50	2376	655	986	227	270	594	793	991	1189	1387	595	793	991	1190	1388	0,75	100
4	8,52	3,86	66,38	22,82	6,03	2266	135	1101	845	267	590	793	991	1190	1387	595	793	991	1190	1388		
5	2,01	1,92	9,64	2,73	8,14	1438	168	11	142	198	594	793	991	1189	861	595	793	991	1190	1388		
6	3,36	18,23	3,11	49,65	7,05	1760	404	75	555	368	594	793	991	1190	1387	595	793	991	1190	1388		
7	1,98	11,78	0,88	2,15	9,67	1885	399	0	118	267	594	793	856	1189	1387	595	793	991	1190	1388		
8	3,42	0,70	2,18	16,98	1,34	2079	75	208	432	161	594	793	856	1189	807	595	793	991	1190	1388		
9	1,57	1,18	29,26	15,70	3,78	2079	230	494	671	76	585	793	991	876	856	595	793	991	1190	1388		
10	0,98	21,91	14,00	1,94	3,14	1530	483	534	321	123	594	793	991	1189	830	595	793	991	1190	1388		
11	1,69	0,21	6,08	1,42	4,88	1303	9	484	117	314	608	811	876	1216	813	608	811	1014	1217	1419		
12	0,88	17,15	4,72	0,53	2,85	909	495	208	78	215	607	858	1072	902	929	644	858	1073	1287	1502		
13	1,58	28,78	0,25	3,98	5,28	1751	570	0	167	724	634	791	1072	1287	1501	644	858	1073	1287	1502		
14	0,52	4,73	4,43	1,39	2,56	1406	438	78	306	149	594	793	991	861	1387	595	793	991	1190	1388		
15	1,05	5,49	18,86	1,53	4,28	2336	161	367	270	7	594	793	825	1189	1387	595	793	991	1190	1388		
16	8,03	30,75	0,55	1,33	14,45	1448	747	227	3	536	696	929	1161	969	1625	697	929	1161	1394	1626		
17	0,86	8,72	3,69	6,03	2,24	1620	147	485	461	237	593	793	991	1189	1387	595	793	991	1190	1388		
18	0,39	18,98	3,70	5,72	1,54	1257	748	233	412	126	594	793	991	1189	856	595	793	991	1190	1388		
19	3,05	2,65	6,59	4,87	0,20	2045	180	243	412	0	690	921	1151	1381	965	691	921	1151	1382	1612		
20	0,47	1,81	2,72	5,28	0,15	785	67	114	202	0	586	793	991	1189	1387	595	793	991	1190	1388		
21	1,06	0,33	3,73	0,51	0,06	1612	72	124	92	0	681	908	1135	981	1010	681	908	1135	1362	1589		
22	1,04	0,35	0,83	2,20	3,24	1503	42	7	137	83	665	887	1108	1330	1552	665	887	1109	1331	1552		
23	3,11	5,20	4,77	6,44	1,11	2207	54	386	279	189	594	793	991	1189	1387	595	793	991	1190	1388		
24	0,82	3,71	14,41	1,44	2,73	1594	90	482	159	426	594	793	991	1189	1387	595	793	991	1190	1388		
25	3,51	69,13	2,01	16,21	9,62	1836	834	195	763	587	594	793	991	1189	1387	595	793	991	1190	1388		
26	0,65	17,17	5,68	11,98	0,78	1378	248	228	458	167	594	793	991	1189	1387	595	793	991	1190	1388		
27	1,27	5,19	10,95	6,53	0,30	2241	143	503	189	0	591	793	849	1189	793	595	793	991	1190	1388		
28	4,29	0,22	1,67	0,18	12,00	917	132	61	22	635	625	834	943	1251	1460	626	834	1043	1251	1460		
29	5,75	5,63	0,19	8,37	0,09	477	372	0	245	0	483	645	683	679	1128	484	645	806	968	1129		
Promedio	2,41	10,26	10,68	8,33	4,14	1605	281	303	324	230	605	807	975	1145	1207	608	810	1013	1216	1419		

n=18	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
	Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25
0	1,13	0,39	1,69	32,69	2,44	1646	5	39	331	353	758	1011	1263	1517	1769	758	1011	1264	1517	1769	0	100
1	1,53	2,10	1,15	1,33	1,21	1648	15	227	142	376	688	920	1150	1380	1610	690	920	1150	1380	1610	0,25	100
2	2,56	3,59	1,71	1,70	0,40	1845	510	271	254	46	690	920	989	1380	1610	690	920	1150	1380	1610	0,5	100
3	3,43	1,27	3,74	5,43	23,15	2359	223	337	237	342	690	920	1030	1380	1610	690	920	1150	1380	1610	0,75	100
4	1,69	1,70	0,28	0,06	0,22	1374	98	37	0	61	690	920	1016	1045	1610	690	920	1150	1380	1610		
5	1,05	2,76	0,08	3,23	22,28	1981	152	0	480	979	685	920	1017	1380	1610	690	920	1150	1380	1610		
6	1,65	9,62	8,97	0,69	0,31	1920	373	686	95	133	690	920	989	1380	1610	690	920	1150	1380	1610		
7	0,60	14,33	13,54	0,28	2,42	1217	759	246	28	609	690	920	1150	1380	1610	690	920	1150	1380	1610		
8	2,62	16,67	35,87	2,44	10,93	1685	728	587	410	579	690	920	1150	1380	1610	690	920	1150	1380	1610		
9	2,67	0,92	1,82	0,95	2,25	2422	239	274	213	129	690	920	972	1380	1610	690	920	1150	1380	1610		
10	3,47	1,03	9,66	0,45	0,69	2571	32	388	89	65	690	920	1150	1380	1610	690	920	1150	1380	1610		
11	1,16	1,20	0,07	2,32	0,78	2044	24	55	208	14	728	971	1213	1067	1699	728	971	1214	1457	1699		
12	0,86	7,03	34,91	2,77	2,00	1380	604	640	174	267	828	1105	1381	1657	1933	829	1105	1381	1658	1934		
13	1,29	53,47	0,17	0,68	4,85	1270	1068	0	49	482	823	1105	1105	1657	1933	829	1105	1381	1658	1934		
14	2,71	2,21	2,49	10,00	0,19	1860	317	193	275	0	690	920	1150	1380	1016	690	920	1150	1380	1610		
15	0,75	2,29	0,78	0,10	0,28	1974	289	145	0	23	685	920	1150	1039	1610	690	920	1150	1380	1610		
16	1,55	42,30	2,72	0,38	0,20	595	737	280	23	20	609	813	1016	1219	1422	610	813	1016	1220	1423		
17	1,12	0,93	1,78	1,97	0,97	1263	35	260	302	240	684	920	1150	1380	1610	690	920	1150	1380	1610		
18	2,43	2,77	49,76	0,56	0,66	1635	364	796	53	52	690	920	1150	1380	1016	690	920	1150	1380	1610		
19	0,79	3,80	1,07	3,98	3,39	2739	276	166	219	270	786	1050	1312	1575	1837	788	1050	1313	1575	1838		
20	0,89	1,25	1,18	0,22	1,43	1302	109	175	50	10	722	979	1223	1059	1713	734	979	1224	1469	1713		
21	2,03	21,54	0,20	0,27	30,82	1758	416	0	11	183	620	827	942	845	1447	620	827	1034	1241	1447		
22	13,92	0,07	2,18	0,22	0,11	1402	0	23	0	0	650	908	1135	964	1589	681	908	1135	1362	1589		
23	1,00	19,92	2,14	22,24	3,05	1846	220	24	856	274	690	920	1150	1380	1610	690	920	1150	1380	1610		
24	3,77	3,76	0,27	2,50	7,91	2067	151	0	195	165	689	920	1006	1380	1610	690	920	1150	1380	1610		
25	1,18	0,11	16,85	0,88	1,49	1506	0	385	55	310	678	920	1150	1380	1610	690	920	1150	1380	1610		
26	3,76	0,67	0,45	1,99	2,17	2267	265	40	135	114	690	920	1003	1380	961	690	920	1150	1380	1610		
27	2,00	1,44	1,08	9,72	0,41	1511	21	93	842	97	684	920	1150	1380	1610	690	920	1150	1380	1610		
28	5,38	1,20	0,21	1,44	0,18	761	0	53	291	0	671	901	1126	1351	898	676	901	1126	1352	1577		
29	3,59	0,12	0,10	4,18	1,43	1069	0	0	701	156	589	794	992	848	848	596	794	993	1191	1390		
Promedio	2,42	7,35	6,56	3,86	4,29	1697	268	214	224	212	695	931	1113	1310	1528	699	931	1164	1397	1630		

n=20	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
	Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25
0	2,75	0,94	5,19	0,65	4,31	1153	94	225	29	87	642	892	897	1338	1561	669	892	1115	1338	1561	0	100
1	3,13	1,37	17,58	1,13	0,05	3437	125	182	3	0	786	1052	1067	1097	1072	789	1052	1315	1578	1841	0,25	96,6667
2	1,56	1,64	0,33	8,82	0,88	2638	246	0	292	103	780	1052	1315	1578	1841	789	1052	1315	1578	1841	0,5	100
3	4,06	3,05	60,46	0,74	1,20	3244	467	508	142	106	789	1052	1315	1578	1841	789	1052	1315	1578	1841	0,75	100
4	5,77	0,15	0,37	10,13	1,68	2306	0	35	265	263	786	1052	1315	1578	1841	789	1052	1315	1578	1841		
5	3,31	4,08	0,23	0,27	1,34	1990	94	0	23	203	788	1052	1148	1578	1108	789	1052	1315	1578	1841		
6	7,69	0,46	0,72	0,16	3,10	2506	0	96	0	336	789	1052	1315	1578	1841	789	1052	1315	1578	1841		
7	3,98	3,14	0,61	0,33	44,16	2367	177	155	0	616	789	1052	1315	1164	1841	789	1052	1315	1578	1841		
8	5,06	0,59	0,35	3,62	4,86	2556	0	0	111	146	789	1052	1121	1578	1841	789	1052	1315	1578	1841		
9	5,32	111,07	33,51	0,20	4,12	2972	733	250	0	95	788	1052	1315	1121	1104	789	1052	1315	1578	1841		
10	4,72	38,88	4,51	6,58	6,10	3296	458	238	398	69	789	1052	1315	1578	1841	789	1052	1315	1578	1841		
11	4,45	3,64	0,16	2,55	4,12	1362	161	0	102	88	850	1136	1234	1704	1253	852	1136	1420	1704	1988		
12	6,60	3,48	1,46	6,08	4,67	2494	31	235	270	109	832	1110	1387	1665	1942	833	1110	1388	1665	1943		
13	3,21	10,21	1,81	3,24	6,51	2518	79	187	74	133	832	1110	1387	1205	1942	833	1110	1388	1665	1943		
14	7,77	7,95	0,09	0,91	6,63	3081	477	0	50	183	789	1052	1315	1148	1841	789	1052	1315	1578	1841		
15	1,57	5,97	2,36	1,53	2,97	2577	171	50	99	110	789	1052	1315	1578	1148	789	1052	1315	1578	1841		
16	3,01	2,72	4,08	6,18	2,25	2139	175	202	364	119	722	975	1218	1462	1706	731	975	1219	1463	1706		
17	6,27	0,19	17,99	0,32	1,32	3871	0	436	0	75	789	1052	1315	1578	1841	789	1052	1315	1578	1841		
18	2,03	5,78	0,22	1,71	2,61	2931	540	0	81	315	789	1052	1138	1143	1841	789	1052	1315	1578	1841		
19	6,15	18,78	0,13	3,08	0,16	1585	412	109	235	0	680	919	947	1095	1004	689	919	1149	1379	1608		
20	1,05	1,39	0,10	0,18	0,21	1994	11	0	0	58	844	1144	1430	1149	2002	858	1144	1430	1716	2002		
21	26,64	23,56	900,04	3,49	22,82	1791	253	1065	0	140	772	1030	1287	1446	1802	773	1030	1288	1545	1803		
22	3,38	6,02	0,57	3,70	4,94	807	450	0	47	144	716	998	1247	1497	1746	749	998	1248	1497	1747		
23	5,77	0,13	0,16	34,25	0,56	2642	0	0	677	92	785	1052	1121	1578	1841	789	1052	1315	1578	1841		
24	3,04	1,51	3,56	0,92	2,16	3234	132	256	108	24	789	1052	1315	1578	1100	789	1052	1315	1578	1841		
25	1,62	0,44	1,18	4,75	0,39	2297	105	152	35	0	789	1052	1052	1106	1841	789	1052	1315	1578	1841		
26	4,16	8,28	0,27	0,17	0,09	2838	129	0	0	0	785	1052	1238	1148	1841	789	1052	1315	1578	1841		
27	2,60	3,45	18,95	1,57	3,94	2312	157	277	31	349	785	1052	1066	1052	1841	789	1052	1315	1578	1841		
28	637,78	92,19	165,23	3,94	104,66	2220	384	607	143	678	740	993	1241	1489	1738	745	993	1241	1490	1738		
29	2,91	3,89	0,13	67,26	5,12	1689	45	0	499	289	656	875	929	1313	1531	656	875	1094	1313	1531		
Promedio	25,91	12,16	41,41	5,95	8,27	2428	204	176	136	164	775	1039	1221	1390	1651	779	1039	1299	1559	1818		

• **Batería mediana**

n=22	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	0,25	100
0	3,23	0,94	15,11	0,31	0,33	2408	163	371	0	84	960	1280	1600	1920	2240	960	1280	1600	1920	2240	0	100
1	16,29	11,86	1,16	0,30	2,98	2733	67	265	0	6	849	1132	1415	1698	1185	849	1132	1415	1698	1981	0,25	100
2	5,75	7,98	25,63	4,93	18,49	2475	268	262	195	676	849	1132	1415	1191	1981	849	1132	1415	1698	1981	0,5	100
3	48,63	9,67	7,63	9,47	39,68	3783	73	361	251	607	849	1132	1415	1698	1981	849	1132	1415	1698	1981	0,75	100
4	20,54	5,48	0,25	0,14	3,36	3111	567	91	23	377	849	1132	1415	1253	1981	849	1132	1415	1698	1981		
5	13,31	7,98	0,30	0,76	11,01	3558	53	7	0	62	849	1132	1415	1698	1981	849	1132	1415	1698	1981		
6	17,27	12,85	2,06	120,12	0,31	2387	145	357	622	0	844	1132	1415	1698	1214	849	1132	1415	1698	1981		
7	7,69	5,31	3,94	0,16	0,19	2436	152	128	0	0	849	1132	1415	1166	1182	849	1132	1415	1698	1981		
8	10,67	94,21	1,11	16,08	0,56	3107	705	203	146	33	849	1104	1415	1698	1981	849	1132	1415	1698	1981		
9	201,63	3,08	0,23	0,55	3,12	3241	76	0	19	188	849	1132	1415	1698	1981	849	1132	1415	1698	1981		
10	45,50	0,31	0,41	0,33	773,99	2358	0	65	0	916	849	1132	1415	1698	1981	849	1132	1415	1698	1981		
11	4,22	1,69	2,66	1,00	0,31	904	260	53	17	0	849	1132	1415	1165	1981	849	1132	1415	1698	1981		
12	23,68	52,19	2,06	0,64	2,06	1182	326	159	67	74	814	1087	1092	1630	1902	815	1087	1359	1631	1902		
13	150,74	6,83	0,39	79,58	33,80	1469	395	0	371	352	793	1087	1358	1631	1902	815	1087	1359	1631	1902		
14	2,12	2,64	0,06	0,50	16,04	2304	307	0	11	539	849	1132	1200	1698	1981	849	1132	1415	1698	1981		
15	11,87	16,11	5,16	0,33	2,08	1843	272	153	92	13	849	1132	1415	1199	1772	849	1132	1415	1698	1981		
16	4,83	3,70	0,44	2,45	0,64	3042	623	17	58	33	1003	1338	1427	2007	2341	1004	1338	1673	2007	2342		
17	11,95	9,29	0,84	0,23	7,79	2315	245	188	71	128	849	1132	1177	1183	1981	849	1132	1415	1698	1981		
18	57,15	1,59	0,52	2,00	0,14	3438	279	0	246	0	849	1132	1415	1698	1981	849	1132	1415	1698	1981		
19	7,64	1,86	0,20	68,65	87,37	2412	9	0	371	611	876	1168	1460	1752	2044	876	1168	1460	1752	2044		
20	10,92	2,32	3,11	0,23	0,64	3302	22	75	0	86	888	1185	1481	1281	2073	889	1185	1481	1778	2074		
21	6,24	1,11	2,59	0,16	0,86	1583	43	223	8	40	790	1055	1318	1582	1204	791	1055	1319	1583	1846		
22	0,47	1,41	4,00	313,89	0,25	1770	41	28	1387	0	909	1212	1515	1818	2121	909	1212	1515	1818	2121		
23	7,35	0,69	2,41	6,30	1,91	1875	59	314	79	224	848	1132	1415	1698	1159	849	1132	1415	1698	1981		
24	12,01	31,02	5,19	9,64	2,81	2114	674	28	136	101	849	1132	1415	1698	1981	849	1132	1415	1698	1981		
25	2,33	26,96	3,98	4,44	2,53	1408	535	202	27	132	841	1132	1415	1698	1981	849	1132	1415	1698	1981		
26	6,45	1,12	4,12	28,17	0,52	1975	90	351	510	15	849	1132	1415	1698	1981	849	1132	1415	1698	1981		
27	30,97	9,59	0,27	1,78	3,23	2860	343	1	140	538	843	1132	1189	1698	1981	849	1132	1415	1698	1981		
28	2,31	1,12	2,64	7,81	342,63	1271	0	49	694	788	832	1117	1396	1144	1955	838	1117	1396	1676	1955		
29	4,83	1,11	92,51	0,41	19,56	744	130	485	0	167	750	1001	1251	1068	1751	751	1001	1251	1502	1752		
Promedio	24,95	11,07	6,37	22,71	45,97	2314	231	148	185	226	853	1138	1384	1559	1859	854	1139	1424	1709	1993		

n=24	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-	100
0	41,31	6,80	2,09	1,65	174,07	2619	88	219	13	758	933	1258	1572	1887	2202	944	1258	1573	1887	2202	0,25	100
1	2,55	2,92	14,59	0,51	0,16	1850	409	285	22	0	966	1292	1615	1938	2261	969	1292	1615	1938	2261	0,25	100
2	8,84	62,45	0,17	1,52	0,91	3033	627	0	14	23	969	1292	1615	1938	2261	969	1292	1615	1938	2261	0,5	100
3	85,86	0,38	3,78	22,91	900,02	3744	0	30	323	570	969	1292	1615	1938	2261	969	1292	1615	1938	2261	0,75	96,6667
4	7,33	45,99	12,04	0,48	9,23	1754	383	328	0	290	969	1292	1370	1324	2261	969	1292	1615	1938	2261		
5	54,19	10,78	2,24	1,53	1,91	2896	666	110	33	62	969	1292	1615	1938	2261	969	1292	1615	1938	2261		
6	2,65	12,34	47,91	0,30	0,48	2064	287	282	0	0	969	1292	1615	1938	2261	969	1292	1615	1938	2261		
7	6,92	3,11	13,82	30,95	8,56	1956	63	380	162	230	966	1292	1325	1938	2261	969	1292	1615	1938	2261		
8	45,23	155,34	102,01	0,75	1,66	2946	512	451	51	75	969	1292	1615	1342	2261	969	1292	1615	1938	2261		
9	13,49	0,64	0,39	7,88	6,16	2574	0	0	6	403	967	1292	1615	1938	2261	969	1292	1615	1938	2261		
10	7,68	0,37	0,37	94,33	1,27	1797	81	56	745	0	960	1292	1615	1938	2261	969	1292	1615	1938	2261		
11	3,33	19,74	4,03	1,70	41,66	3409	333	314	137	711	1076	1435	1793	2152	2511	1076	1435	1794	2153	2511		
12	11,76	22,76	16,79	350,81	95,68	1568	327	134	781	434	984	1313	1641	1970	2298	985	1313	1641	1970	2298		
13	50,55	173,72	27,29	15,61	1,40	2184	980	519	191	69	947	1313	1478	1440	2297	985	1313	1641	1970	2298		
14	6,32	0,61	10,65	75,27	0,70	2081	0	58	609	54	961	1292	1615	1938	2261	969	1292	1615	1938	2261		
15	42,07	1,16	5,44	0,71	0,14	2862	0	123	89	0	961	1292	1350	1376	1324	969	1292	1615	1938	2261		
16	22,41	3,66	4,12	60,52	0,22	2286	104	121	233	58	678	904	1130	1356	1582	678	904	1130	1356	1582		
17	4,75	900,03	0,34	2,00	82,54	2625	1049	72	150	272	968	1292	1615	1938	2261	969	1292	1615	1938	2261		
18	35,04	16,21	0,17	32,56	8,41	2680	220	66	481	67	954	1292	1412	1938	2261	969	1292	1615	1938	2261		
19	3,37	85,32	0,27	0,57	54,92	711	402	0	61	308	844	1128	1217	1692	1974	846	1128	1410	1692	1974		
20	59,77	900,03	33,61	0,21	76,58	2258	902	116	16	386	900	1200	1500	1800	2100	900	1200	1500	1800	2100		
21	1,24	0,65	0,48	2,11	5,57	2658	42	14	95	149	1161	1551	1938	2326	2714	1163	1551	1939	2327	2714		
22	212,78	1,67	5,92	2,34	17,92	1906	57	77	69	148	935	1261	1576	1891	2206	946	1261	1576	1892	2207		
23	103,74	7,23	6,23	2,46	2,17	2378	336	197	281	150	969	1292	1615	1938	2261	969	1292	1615	1938	2261		
24	17,00	1,49	0,68	12,79	3,33	2015	90	60	263	262	945	1292	1615	1938	2261	969	1292	1615	1938	2261		
25	3,78	900,03	10,39	38,68	39,97	2269	959	95	353	698	935	1292	1615	1938	2261	969	1292	1615	1938	2261		
26	5,36	2,24	0,35	16,12	17,79	1459	129	88	686	342	968	1292	1615	1938	2261	969	1292	1615	1938	2261		
27	2,21	7,01	60,58	2,87	0,15	1041	84	212	128	0	969	1292	1615	1938	2261	969	1292	1615	1938	2261		
28	4,57	4,37	0,57	0,36	0,18	1614	127	29	74	0	991	1322	1652	1983	2313	992	1322	1653	1983	2314		
29	18,07	3,07	1,61	0,46	0,24	1009	50	134	52	100	808	1086	1357	1629	1900	815	1086	1358	1629	1901		
Promedio	29,47	111,74	12,96	26,03	51,80	2208	310	152	204	221	952	1277	1551	1839	2204	958	1277	1597	1916	2235		

n=26	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	93,33
0	17,355	2,656	12,325	10,021	0,094	3564	39	223	286	0	1017	1362	1567	2043	2383	1022	1362	1703	2043	2384	0	93,33
1	70,244	122,287	0,719	2,562	39,179	2779	678	43	4	188	1017	1357	1696	2035	2375	1018	1357	1696	2036	2375	0,25	100
2	30,864	11,654	1,578	0,375	12,469	2117	78	81	0	17	1017	1357	1696	1417	2374	1018	1357	1696	2036	2375	0,5	100
3	337,462	900,047	0,141	629,427	5,927	3537	362	0	1152	175	998	1357	1696	2036	2374	1018	1357	1696	2036	2375	0,75	100
4	4,565	0,797	19,425	2,109	9,935	2263	0	46	234	266	1011	1357	1696	2035	2374	1018	1357	1696	2036	2375		
5	7,663	16,090	39,364	47,827	0,219	1383	343	224	526	0	1012	1357	1696	2036	2374	1018	1357	1696	2036	2375		
6	107,168	0,656	0,437	0,219	0,406	3050	0	0	0	3	1002	1357	1696	2035	2374	1018	1357	1696	2036	2375		
7	7,608	6,530	0,109	0,344	66,690	2182	24	20	0	575	1017	1357	1696	1435	2375	1018	1357	1696	2036	2375		
8	3,327	5,777	0,297	32,057	7,289	1779	312	54	311	40	1007	1357	1696	2035	1491	1018	1357	1696	2036	2375		
9	10,105	56,470	1,281	9,670	69,906	3562	491	85	472	432	1013	1357	1696	2035	2375	1018	1357	1696	2036	2375		
10	24,865	3,249	2,649	3,430	9,051	2429	246	46	43	109	1006	1357	1696	2035	2374	1018	1357	1696	2036	2375		
11	30,537	8,517	0,219	8,661	0,281	2185	60	0	234	0	975	1300	1625	1950	2275	975	1300	1625	1950	2275		
12	14,031	4,280	54,454	7,568	14,892	1558	11	109	128	147	901	1224	1530	1836	2142	918	1224	1530	1836	2142		
13	20,381	0,609	50,479	0,562	13,373	1898	0	525	0	121	904	1224	1530	1836	2142	918	1224	1530	1836	2142		
14	11,763	17,042	0,765	0,406	1,842	3429	214	0	17	34	1006	1357	1696	2035	2374	1018	1357	1696	2036	2375		
15	2,921	1,297	167,386	39,388	0,250	2259	85	302	408	0	1016	1357	1696	2035	2374	1018	1357	1696	2036	2375		
16	2,421	356,171	0,186	16,371	4,546	2021	808	0	188	156	963	1284	1315	1430	2247	963	1284	1605	1926	2247		
17	900,032	7,155	0,203	0,500	3,015	2989	98	0	57	7	1001	1357	1508	1466	1443	1018	1357	1696	2036	2375		
18	3,247	9,266	0,344	0,297	37,596	1725	76	57	2	321	1003	1357	1696	2035	2375	1018	1357	1696	2036	2375		
19	1,687	0,406	0,156	1,218	4,343	532	66	0	116	206	900	1201	1501	1266	1317	901	1201	1501	1802	2102		
20	3,515	900,037	25,963	5,782	55,218	3398	1056	659	112	433	1185	1581	1976	2371	2767	1186	1581	1976	2372	2767		
21	8,170	1,640	0,805	0,141	5,608	2020	180	13	0	54	1151	1534	1918	1700	2686	1151	1535	1919	2303	2686		
22	8,623	0,812	10,791	16,293	44,278	3912	0	91	424	814	1056	1409	1761	2113	2466	1057	1409	1761	2114	2466		
23	8,311	10,435	2,390	0,156	0,312	1926	118	20	0	0	1017	1357	1696	2035	2374	1018	1357	1696	2036	2375		
24	24,493	61,206	0,250	5,561	97,019	2690	300	0	226	474	1012	1357	1696	2035	2375	1018	1357	1696	2036	2375		
25	26,301	6,468	12,107	0,328	2,718	1784	91	168	0	74	998	1357	1696	2035	2374	1018	1357	1696	2036	2375		
26	17,913	4,277	0,219	70,853	3,374	1785	16	0	399	116	998	1357	1696	2035	2374	1018	1357	1696	2036	2375		
27	16,400	267,097	0,531	6,858	2,906	2950	504	0	252	5	1000	1357	1389	2035	2374	1018	1357	1696	2036	2375		
28	900,020	6,252	8,214	15,731	0,487	3416	101	296	29	117	1011	1373	1716	2059	2402	1030	1373	1716	2060	2403		
29	4,364	2,078	12,575	1,297	16,418	832	150	206	74	117	895	1184	1495	1322	2093	897	1196	1495	1794	2093		
Promedio	87,55	93,04	14,21	31,20	17,65	2398	217	109	190	167	1004	1349	1655	1894	2274	1012	1349	1686	2024	2361		

n=28	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	93,3333
0	385,44	0,86	0,56	8,91	512,85	1381	0	0	121	264	1004	1348	1685	2022	2359	1011	1348	1685	2022	2359	0	93,3333
1	517,13	44,41	7,31	900,00	0,33	2674	237	67	554	0	1050	1401	1751	2101	1462	1051	1401	1751	2102	2452	0,25	93,3333
2	6,56	124,81	168,91	715,18	57,32	1629	443	379	754	410	1049	1401	1751	2102	2451	1051	1401	1751	2102	2452	0,5	80
3	3,45	4,81	0,79	32,88	28,61	1246	42	77	300	64	1037	1401	1751	2101	2451	1051	1401	1751	2102	2452	0,75	90
4	407,46	900,06	768,59	900,02	5,17	2577	689	326	928	0	1046	1401	1751	2101	2451	1051	1401	1751	2102	2452		
5	66,86	2,39	1,41	0,41	5,57	1972	62	56	0	346	1044	1401	1751	2101	2451	1051	1401	1751	2102	2452		
6	3,91	4,99	15,59	2,11	71,02	2361	135	2	35	86	1064	1450	1812	2175	2538	1088	1450	1813	2175	2538		
7	49,89	75,40	0,56	0,60	0,45	3503	139	34	92	51	1154	1539	1923	2308	2693	1154	1539	1924	2309	2693		
8	2,47	5,76	70,46	1,33	23,95	552	70	106	0	106	995	1348	1685	2022	1973	1011	1348	1685	2022	2359		
9	15,52	15,94	0,23	0,57	0,83	3356	255	0	0	4	1066	1421	1776	2131	1861	1066	1421	1776	2132	2487		
10	8,73	5,02	7,03	900,02	900,02	2403	0	431	938	746	1024	1378	1722	2067	2411	1034	1378	1723	2067	2412		
11	117,71	109,97	64,72	28,55	1,08	1658	313	230	268	14	995	1334	1668	1477	1460	1001	1334	1668	2001	2335		
12	76,74	34,58	0,36	900,03	900,03	4113	40	0	1126	1194	1307	1743	1771	2614	3050	1307	1743	2179	2615	3050		
13	5,05	18,51	362,76	0,53	0,56	3388	182	279	52	51	1282	1743	2179	2614	3050	1307	1743	2179	2615	3050		
14	61,65	18,03	23,70	58,47	37,44	2075	3	257	360	328	1050	1401	1751	2101	2451	1051	1401	1751	2102	2452		
15	2,56	312,96	64,52	75,08	869,52	916	283	363	181	531	1016	1401	1751	2102	2452	1051	1401	1751	2102	2452		
16	38,69	10,07	0,23	0,65	496,33	2554	183	0	0	252	1016	1381	1726	2071	2417	1036	1381	1726	2072	2417		
17	122,76	30,65	900,02	4,24	11,08	3185	272	825	123	23	1046	1401	1751	1991	2451	1051	1401	1751	2102	2452		
18	160,77	745,48	10,29	44,83	9,43	2286	514	231	154	214	1050	1401	1751	2102	2451	1051	1401	1751	2102	2452		
19	4,40	165,32	2,08	9,93	0,81	3529	480	246	18	0	1319	1776	2220	1840	3108	1332	1776	2220	2664	3108		
20	65,43	189,92	4,95	2,73	8,34	2160	306	17	50	81	1144	1529	1911	2293	1698	1147	1529	1911	2294	2676		
21	8,79	41,54	0,60	1,50	0,95	2264	390	76	79	35	1088	1460	1825	2190	1484	1095	1460	1825	2190	2555		
22	164,67	16,78	900,03	74,30	101,83	3856	72	1005	809	859	1224	1658	2072	2487	2902	1244	1658	2073	2487	2902		
23	36,86	160,22	3,53	900,03	45,73	1908	183	225	350	171	1029	1401	1510	1493	2451	1051	1401	1751	2102	2452		
24	900,03	4,52	823,65	4,47	900,02	2924	150	718	134	660	1009	1401	1751	2101	1448	1051	1401	1751	2102	2452		
25	181,61	765,45	4,49	15,10	0,94	2488	276	133	76	11	1051	1401	1751	2101	2210	1051	1401	1751	2102	2452		
26	10,27	900,02	69,08	284,29	163,25	1967	1057	394	298	175	1044	1401	1751	2102	2452	1051	1401	1751	2102	2452		
27	248,02	0,68	78,35	900,03	64,28	2615	0	213	621	268	1039	1401	1751	2101	1487	1051	1401	1751	2102	2452		
28	900,03	0,34	2,98	3,28	0,70	2813	0	59	89	0	1044	1393	1741	2089	2437	1045	1393	1741	2090	2438		
29	12,04	0,52	0,58	2,27	22,30	1846	0	29	100	135	960	1280	1600	1920	2240	960	1280	1600	1920	2240		
Promedio	152,85	157,00	145,28	225,74	174,69	2407	226	226	287	236	1075	1447	1786	2101	2293	1085	1447	1808	2170	2532		

n=30	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	90
0	47,06	12,58	0,70	2,51	0,47	2932	109	75	64	0	969	1309	1636	1963	2290	982	1309	1636	1964	2291	0	93,33
1	900,02	9,56	2,19	900,03	13,78	4536	191	35	905	239	1134	1517	1896	2275	2654	1138	1517	1896	2276	2655	0,25	96,67
2	11,42	900,03	6,38	0,61	145,87	2370	704	79	0	455	1127	1517	1591	2275	2655	1138	1517	1896	2276	2655	0,5	96,67
3	28,43	22,06	0,37	87,10	520,74	1702	24	0	603	243	1136	1517	1896	2276	2655	1138	1517	1896	2276	2655	0,75	96,67
4	598,42	37,47	0,56	37,53	53,84	2619	237	0	138	206	1137	1517	1896	2276	2654	1138	1517	1896	2276	2655		
5	6,00	70,27	380,60	0,55	0,44	2030	103	231	0	39	1130	1517	1896	2275	2654	1138	1517	1896	2276	2655		
6	28,56	10,92	13,42	0,12	0,31	2963	297	82	0	0	1207	1613	2016	2419	1641	1210	1613	2016	2420	2823		
7	10,07	21,71	0,23	0,61	254,42	2652	42	0	111	439	1110	1480	1850	2220	2590	1110	1480	1850	2220	2590		
8	6,33	520,63	412,83	400,61	106,87	1866	525	252	159	241	981	1309	1636	1964	2291	982	1309	1636	1964	2291		
9	201,23	41,27	36,29	76,01	181,34	2956	376	28	146	211	1204	1608	2010	2412	2814	1206	1608	2010	2412	2814		
10	15,44	900,04	57,78	37,31	0,83	2141	1334	727	255	112	1231	1642	2053	2463	1778	1232	1642	2053	2463	2874		
11	84,53	97,22	0,72	4,33	0,39	4451	681	0	25	0	1210	1613	2016	2419	2822	1210	1613	2016	2420	2823		
12	900,03	10,93	0,44	37,19	31,33	3882	32	0	415	45	1225	1637	2053	2464	2875	1232	1643	2054	2465	2875		
13	175,67	13,95	11,50	8,66	26,04	2823	12	137	59	411	1231	1643	2053	2464	2875	1232	1643	2054	2465	2875		
14	7,31	195,35	35,59	299,01	0,45	1628	416	169	799	3	1127	1517	1896	2276	2654	1138	1517	1896	2276	2655		
15	148,80	160,90	1,11	6,27	5,57	2053	420	50	213	173	1136	1517	1896	2275	2654	1138	1517	1896	2276	2655		
16	607,40	1,38	0,72	12,29	20,57	4969	84	0	301	65	1243	1657	2071	2485	2899	1243	1657	2071	2486	2900		
17	6,72	2,76	159,60	428,77	11,52	2072	0	313	406	166	1132	1517	1896	2276	2654	1138	1517	1896	2276	2655		
18	46,94	462,77	0,48	24,66	82,75	1740	179	62	43	277	1137	1517	1896	1667	2655	1138	1517	1896	2276	2655		
19	15,75	0,64	0,37	0,17	0,78	1530	0	33	0	3	925	1239	1270	1858	1351	929	1239	1549	1859	2168		
20	11,29	0,37	2,80	4,13	0,41	4253	12	223	164	49	1351	1802	2252	2703	1830	1352	1802	2253	2703	3154		
21	54,27	0,31	53,34	1,50	900,04	3370	0	281	0	830	1176	1576	1970	1656	2758	1182	1576	1970	2364	2758		
22	2,22	0,22	0,39	0,27	5,72	2597	0	11	38	44	1123	1498	1872	2000	2621	1124	1498	1873	2247	2622		
23	5,17	32,22	16,16	7,20	659,38	2072	106	316	85	387	1136	1517	1896	1707	2655	1138	1517	1896	2276	2655		
24	8,85	57,66	862,01	0,47	2,84	2095	161	559	0	45	1116	1517	1896	2275	1588	1138	1517	1896	2276	2655		
25	47,14	78,06	1,72	343,40	0,28	2012	384	166	396	0	1134	1517	1896	2276	2654	1138	1517	1896	2276	2655		
26	748,61	47,26	900,02	81,32	0,33	2489	79	1836	322	53	1134	1517	1896	2276	1586	1138	1517	1896	2276	2655		
27	40,91	197,90	286,38	16,60	28,97	1904	191	604	297	349	1137	1517	1896	2275	2654	1138	1517	1896	2276	2655		
28	900,02	0,34	0,55	35,31	70,54	4692	0	0	56	312	1173	1569	1961	1619	2746	1177	1569	1961	2354	2746		
29	11,33	1,60	1,30	4,33	0,95	2067	112	25	1	78	1071	1428	1785	1470	1629	1071	1428	1785	2142	2499		
Promedio	189,20	130,28	108,22	95,30	104,26	2716	227	210	200	183	1143	1529	1891	2175	2428	1147	1529	1911	2294	2676		

• **Batería grande**

n=35	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
	Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25
0	8,68	237,53	411,41	900,03	149,54	371	634	342	576	561	1337	1823	2279	2734	3190	1367	1823	2279	2735	3190	0	80,00
1	900,01	57,05	1,11	77,77	900,00	3600	224	0	286	899	1346	1796	2245	2694	3143	1347	1796	2245	2694	3143	0,25	76,67
2	900,02	722,68	2,28	900,02	900,03	3042	76	0	846	260	1347	1796	2245	2694	3143	1347	1796	2245	2694	3143	0,5	70,00
3	160,84	900,06	900,00	86,15	64,16	2484	833	1643	231	396	1327	1796	2245	2694	3143	1347	1796	2245	2694	3143	0,75	86,67
4	900,01	15,34	0,94	0,34	84,02	3309	89	0	0	381	1347	1796	1847	2694	3143	1347	1796	2245	2694	3143		
5	900,02	1,43	900,04	3,61	900,00	4377	0	1821	29	1056	1340	1796	2245	2694	3143	1347	1796	2245	2694	3143		
6	16,73	1,12	900,02	424,45	44,26	1574	73	549	268	216	1330	1789	2236	2684	3130	1342	1789	2236	2684	3131		
7	900,03	0,84	0,34	900,02	8,22	1814	0	0	1020	0	1433	1911	2289	2866	3344	1433	1911	2389	2867	3344		
8	900,05	0,95	113,23	163,44	0,39	2013	0	497	838	0	1367	1823	2279	2735	3190	1367	1823	2279	2735	3190		
9	146,99	33,20	2,80	0,97	900,02	6621	167	101	0	412	1496	2004	2505	2152	3507	1503	2004	2505	3006	3507		
10	12,69	16,12	0,48	52,57	0,12	3123	264	0	276	0	1256	1677	1806	2515	2134	1258	1677	2096	2516	2935		
11	75,74	1,00	756,94	52,85	8,17	2897	0	671	1	117	1288	1717	2146	2576	1810	1288	1717	2146	2576	3005		
12	70,86	207,45	71,21	84,55	280,98	3960	194	105	151	469	1387	1856	2320	2784	3248	1392	1856	2320	2784	3248		
13	900,02	900,03	3,26	900,02	0,84	4652	421	89	1009	0	1383	1856	2320	2784	3248	1392	1856	2320	2784	3248		
14	900,03	71,77	37,33	1,75	170,84	2898	393	220	49	203	1327	1796	2245	2694	3143	1347	1796	2245	2694	3143		
15	139,32	420,35	900,04	166,34	116,66	3137	38	207	386	224	1340	1796	2245	2694	3143	1347	1796	2245	2694	3143		
16	35,87	1,65	0,30	0,92	0,55	2772	0	0	0	0	1389	1875	2343	1930	3281	1406	1875	2344	2813	3281		
17	900,02	1,35	295,21	177,24	26,44	4451	0	536	371	47	1347	1796	2245	2694	3143	1347	1796	2245	2694	3143		
18	39,72	900,05	7,75	900,02	66,70	3130	408	15	728	429	1342	1796	2245	2694	3143	1347	1796	2245	2694	3143		
19	900,05	0,97	25,07	5,01	0,62	3049	0	211	153	0	1300	1734	2167	2601	1984	1301	1734	2168	2601	3035		
20	45,55	673,36	900,02	0,92	408,15	3888	579	1400	0	589	1389	1852	2315	2778	3241	1389	1852	2315	2778	3241		
21	7,98	900,07	93,09	382,86	341,11	2283	1680	203	400	133	1313	1751	2189	2627	3064	1313	1751	2189	2627	3064		
22	900,04	6,59	0,27	179,06	21,46	3508	0	0	166	178	1291	1727	2158	2591	1878	1295	1727	2159	2591	3022		
23	100,29	198,41	900,01	643,47	68,66	2733	111	995	358	87	1346	1796	2245	2694	3143	1347	1796	2245	2694	3143		
24	900,02	234,02	900,01	900,02	86,65	4172	466	624	542	73	1336	1796	2245	2694	3143	1347	1796	2245	2694	3143		
25	403,78	900,03	1,05	900,02	187,17	2216	495	0	471	23	1315	1796	2245	2694	3143	1347	1796	2245	2694	3143		
26	900,03	89,67	50,33	900,03	6,08	2945	359	316	397	93	1336	1796	2245	2694	3143	1347	1796	2245	2694	3143		
27	144,55	900,26	7,67	78,17	58,11	2646	539	61	356	104	1320	1796	2245	2694	3143	1347	1796	2245	2694	3143		
28	34,63	259,43	3,28	900,01	724,55	3648	271	128	861	497	1248	1664	1789	2496	2912	1248	1664	2080	2496	2912		
29	900,04	99,38	0,55	0,77	34,45	4069	167	0	0	247	1410	1889	1940	2833	2275	1417	1889	2361	2834	3306		
Promedio	468,15	291,74	272,87	356,11	218,63	3179	283	358	359	256	1344	1803	2204	2647	2981	1352	1803	2254	2705	3155		

n=40	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	70
0	900,04	5,88	2,42	154,78	1,45	1564	32	74	400	0	1574	2081	2642	3171	3699	1586	2114	2643	3171	3700	0	83,33
1	138,63	164,32	900,04	900,02	900,02	4044	333	355	572	472	1492	1994	2492	2991	3489	1496	1994	2493	2991	3490	0,25	63,33
2	110,29	11,06	1,20	1,56	900,03	3089	0	0	0	869	1495	1994	2492	2057	3489	1496	1994	2493	2991	3490	0,5	83,33
3	52,42	7,98	900,04	340,17	14,25	3430	85	601	116	218	1494	1994	2492	2991	3489	1496	1994	2493	2991	3490	0,75	70
4	900,04	411,54	93,38	50,94	900,02	5464	47	209	356	1326	1491	1994	2493	2991	3489	1496	1994	2493	2991	3490		
5	247,53	7,42	900,04	99,76	0,95	3059	10	220	204	0	1490	1994	2492	2991	3489	1496	1994	2493	2991	3490		
6	398,02	900,04	682,98	679,73	61,76	7975	465	279	602	167	1585	2121	2651	3182	3711	1591	2121	2651	3182	3712		
7	35,66	900,04	0,84	3,74	6,40	3177	792	0	114	89	1701	2269	2836	3403	3970	1702	2269	2836	3404	3971		
8	699,66	1,02	288,44	72,47	25,93	2266	0	176	26	151	1585	1985	2643	3171	3699	1586	2114	2643	3171	3700		
9	900,03	180,96	4,42	0,14	900,02	5828	355	48	0	189	1496	1995	2493	2992	3491	1496	1995	2494	2993	3491		
10	48,25	3,03	5,47	166,73	47,40	5426	143	0	316	239	1520	2027	2533	3041	2115	1520	2027	2534	3041	3547		
11	6,49	58,11	1,37	77,75	0,37	3343	293	34	302	54	1617	2157	2193	3235	2663	1618	2157	2696	3236	3775		
12	10,25	34,55	177,03	900,04	414,98	2970	103	273	1701	692	1715	2287	2859	3430	4002	1715	2287	2859	3431	4002		
13	31,80	45,94	900,03	125,59	1,64	4497	47	341	260	0	1715	2287	2858	3430	3656	1715	2287	2859	3431	4002		
14	34,20	34,23	900,03	3,44	70,51	3656	0	541	0	203	1481	1994	2492	2991	3489	1496	1994	2493	2991	3490		
15	900,04	900,04	1,38	900,22	900,01	4915	403	50	219	354	1494	1994	2492	2991	3489	1496	1994	2493	2991	3490		
16	235,82	5,58	900,02	552,07	9,01	3916	139	1024	425	145	1691	2254	2817	3381	3944	1691	2254	2818	3381	3945		
17	224,07	428,24	900,02	3,66	91,34	4607	105	726	84	153	1492	1994	2492	2991	3489	1496	1994	2493	2991	3490		
18	900,04	412,44	900,02	900,02	110,28	2828	285	674	720	177	1467	1994	2492	2991	3489	1496	1994	2493	2991	3490		
19	900,04	39,33	0,74	1,75	900,04	3706	200	46	0	336	1661	2215	2768	3322	3876	1661	2215	2769	3323	3876		
20	86,49	900,05	299,01	729,58	900,02	2897	506	248	426	154	1592	2124	2655	3186	3717	1593	2124	2655	3186	3717		
21	839,00	0,73	900,04	1,59	0,47	6002	0	513	0	0	1505	2006	2507	3009	3510	1505	2006	2508	3009	3511		
22	9,92	111,77	536,40	0,75	1,59	3669	511	520	0	0	1631	2175	2719	3262	3806	1631	2175	2719	3263	3806		
23	34,88	463,39	3,94	59,36	37,22	2206	553	35	89	320	1462	1994	2492	2991	2146	1496	1994	2493	2991	3490		
24	209,13	300,79	158,41	435,37	251,05	3916	11	16	97	269	1492	1994	2493	2991	3490	1496	1994	2493	2991	3490		
25	900,03	900,08	1,20	573,23	900,02	4133	791	0	698	696	1482	1994	2492	2991	3489	1496	1994	2493	2991	3490		
26	900,04	118,21	900,02	181,60	900,04	4340	28	265	194	1315	1495	1994	2492	2991	3489	1496	1994	2493	2991	3490		
27	35,92	8,62	900,02	900,02	104,73	3419	77	564	2437	140	1495	1994	2492	2991	3490	1496	1994	2493	2991	3490		
28	158,30	4,08	138,88	73,26	258,53	3775	0	476	349	323	1496	2010	2512	3015	3518	1508	2010	2513	3015	3518		
29	900,03	45,83	0,61	244,03	122,57	5339	177	0	92	210	1666	2227	2783	3341	3897	1670	2227	2784	3341	3897		
Promedio	391,57	246,84	409,95	304,45	324,42	3982	216	277	360	309	1552	2071	2579	3084	3493	1558	2077	2596	3115	3634		

n=45	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	- 0,25	50
0	900,04	70,15	2,08	361,22	444,14	4152	59	86	36	197	1842	2457	2507	3686	4300	1843	2457	3071	3686	4300	0	76,667
1	900,10	900,02	900,04	900,03	865,69	3905	928	899	810	202	1684	2261	2826	3391	3957	1696	2261	2826	3392	3957	0,25	70
2	127,44	60,98	900,04	12,97	900,04	4972	186	1424	0	189	1695	2261	2826	3391	3956	1696	2261	2826	3392	3957	0,5	83,333
3	136,35	310,09	900,05	5,88	900,05	3066	339	456	22	738	1687	2261	2826	3391	3956	1696	2261	2826	3392	3957	0,75	66,667
4	433,66	455,88	137,46	305,48	8,62	2188	438	47	463	126	1694	2261	2826	3392	3956	1696	2261	2826	3392	3957		
5	236,61	18,50	134,13	1,93	284,92	3804	76	177	0	275	1690	2261	2826	3391	3957	1696	2261	2826	3392	3957		
6	167,66	41,23	0,33	1,32	759,85	1881	256	0	0	438	1604	2139	2673	2184	3743	1604	2139	2674	3209	3743		
7	900,03	900,03	15,38	24,49	2,03	3360	819	70	124	47	1626	2169	2357	3253	3795	1627	2169	2711	3254	3796		
8	36,77	900,02	900,03	161,61	900,06	6257	391	1300	381	1012	1842	2457	3071	3686	4299	1843	2457	3071	3686	4300		
9	900,04	1,07	1,69	160,61	3,26	4901	0	0	208	0	1889	2523	3153	3785	3097	1892	2523	3154	3785	4415		
10	900,03	168,47	900,04	110,27	900,06	5363	143	625	65	498	1691	2261	2826	3391	3956	1696	2261	2826	3392	3957		
11	100,96	62,60	900,02	63,41	4,10	5932	37	632	53	0	1901	2535	3168	3802	4436	1901	2535	3169	3803	4436		
12	900,10	682,62	18,62	900,02	57,81	6669	663	59	518	41	1929	2572	3215	3858	2642	1929	2572	3215	3858	4501		
13	117,63	1,37	1,47	3,66	900,05	3995	0	0	0	313	1929	2572	3215	3394	4501	1929	2572	3215	3858	4501		
14	68,74	216,14	900,03	8,93	900,19	4932	147	2076	13	767	1690	2261	2267	3391	3956	1696	2261	2826	3392	3957		
15	900,04	307,39	1,18	21,34	636,00	3934	0	0	87	271	1688	2261	2826	3391	3957	1696	2261	2826	3392	3957		
16	900,06	23,69	7,38	900,04	66,21	5671	154	45	160	142	1832	2460	3075	3690	4305	1845	2460	3075	3690	4305		
17	900,03	900,04	129,95	347,24	900,06	5842	509	444	417	1564	1682	2261	2826	3392	2461	1696	2261	2826	3392	3957		
18	900,04	74,46	900,06	181,02	1,63	3795	169	500	57	0	1689	2261	2826	3392	3956	1696	2261	2826	3392	3957		
19	111,47	900,03	13,15	9,12	56,34	3668	508	131	203	142	1857	2477	3096	3715	4334	1858	2477	3096	3716	4335		
20	900,06	520,55	50,11	24,14	784,72	4584	309	152	72	126	1842	2457	3071	2558	4300	1843	2457	3071	3686	4300		
21	250,85	257,98	59,17	278,25	89,32	6924	62	78	280	267	1908	2544	3180	3816	4452	1908	2544	3180	3816	4452		
22	37,47	3,28	12,26	76,38	900,03	3381	0	192	240	951	1703	2286	2857	3429	4000	1715	2286	2858	3429	4001		
23	900,05	900,05	196,82	88,35	7,24	4628	645	228	208	18	1686	2261	2826	3391	3956	1696	2261	2826	3392	3957		
24	900,05	81,16	900,03	14,10	1,52	4641	193	1073	87	0	1685	2261	2826	2306	3956	1696	2261	2826	3392	3957		
25	900,04	549,34	175,19	900,03	91,38	5846	532	166	358	27	1682	2261	2826	3391	3956	1696	2261	2826	3392	3957		
26	900,06	59,28	10,31	268,83	900,05	6020	68	22	230	400	1695	2261	2826	3391	3956	1696	2261	2826	3392	3957		
27	224,68	900,04	68,23	900,02	900,04	3929	410	121	1853	492	1692	2261	2826	3391	3956	1696	2261	2826	3392	3957		
28	363,03	831,73	15,74	26,61	1,70	5026	57	121	178	0	1730	2325	2906	3038	4068	1744	2325	2906	3488	4069		
29	894,41	19,65	73,07	6,05	2,27	5010	68	177	62	0	1860	2495	3118	3742	2597	1871	2495	3119	3743	4366		
Promedio	560,28	370,60	307,47	235,44	438,98	4609	272	377	240	308	1754	2346	2883	3381	3891	1760	2346	2933	3520	4106		

n=50	Tiempo de ejecución de Gurobi					Total Tardiness de A					Makespan de B					Valores de Q					Optimalidad (%)	
Instancias	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	0	0,25	0,5	0,75	-0,25	36,66
0	900,04	28,08	179,45	216,66	900,03	5683	188	226	76	128	1976	2635	3293	3953	4611	1976	2635	3294	3953	4611	0	60
1	283,17	63,00	896,85	2,94	61,42	4402	32	536	0	114	1831	2442	3053	3663	4273	1832	2442	3053	3663	4274	0,25	83,33
2	900,04	900,03	470,03	281,03	900,04	5286	664	273	496	524	1821	2442	3053	3663	4273	1832	2442	3053	3663	4274	0,5	70
3	900,03	7,50	98,30	100,66	900,02	6088	0	85	235	984	1831	2442	3052	3663	2554	1832	2442	3053	3663	4274	0,75	63,33
4	593,65	27,80	209,29	1,59	25,67	5581	128	146	0	110	1828	2442	3053	3663	4273	1832	2442	3053	3663	4274		
5	900,02	110,15	158,80	72,58	900,03	7154	124	60	96	39	1806	2442	3052	3663	4273	1832	2442	3053	3663	4274		
6	139,11	13,03	166,27	406,39	286,21	3087	70	136	384	470	1878	2515	3143	3773	4401	1886	2515	3144	3773	4401		
7	900,04	900,05	9,60	3,11	1,96	6352	687	42	69	0	1941	2588	3235	2648	2738	1941	2588	3235	3882	4529		
8	900,04	900,06	322,27	22,57	21,68	5935	377	340	54	0	1976	2635	3294	3952	4611	1976	2635	3294	3953	4611		
9	900,03	211,60	900,02	900,03	900,04	5893	114	423	509	1518	2082	2776	3470	4164	4858	2082	2776	3470	4164	4858		
10	900,03	900,02	368,95	258,49	486,26	5065	305	33	4	369	1817	2442	3053	3663	4274	1832	2442	3053	3663	4274		
11	520,22	2,29	1,27	10,08	148,96	4099	0	0	100	87	1951	2606	3257	3909	4560	1955	2606	3258	3909	4561		
12	100,56	900,06	900,04	80,47	2,36	8199	1711	422	119	57	2075	2767	3458	4150	4842	2075	2767	3459	4151	4842		
13	900,04	7,77	80,83	900,03	59,50	10259	23	197	1042	76	2075	2767	3458	4150	4842	2075	2767	3459	4151	4842		
14	763,25	13,55	90,50	900,02	0,20	3162	0	109	249	0	1828	2442	3052	3663	4273	1832	2442	3053	3663	4274		
15	900,02	394,54	1,25	900,03	255,63	6551	387	0	1113	385	1831	2442	3052	3663	4274	1832	2442	3053	3663	4274		
16	215,87	16,81	83,27	900,04	900,07	7737	64	193	1345	1369	2032	2709	3386	4063	4740	2032	2709	3386	4064	4741		
17	235,81	159,49	11,18	28,58	26,20	4309	245	0	21	70	1830	2442	3052	2541	4273	1832	2442	3053	3663	4274		
18	900,04	19,18	238,02	900,04	288,02	1858	49	104	622	185	1821	2442	3053	3663	4273	1832	2442	3053	3663	4274		
19	172,47	900,03	900,03	256,55	900,04	2774	370	328	330	295	1724	2299	2873	3449	4023	1724	2299	2874	3449	4023		
20	900,04	900,02	27,10	2,38	900,03	3956	460	159	0	413	1976	2635	3293	3952	4611	1976	2635	3294	3953	4611		
21	232,70	86,12	1,30	14,68	6,73	2812	4	0	21	72	1761	2348	2935	3522	4109	1761	2348	2935	3522	4109		
22	900,05	900,03	138,22	18,95	900,05	2697	637	198	73	373	2025	2711	3388	4066	4744	2033	2711	3389	4067	4744		
23	900,04	900,03	900,02	553,61	900,02	4446	728	531	377	483	1822	2442	3052	3663	4273	1832	2442	3053	3663	4274		
24	900,04	900,03	92,88	900,04	0,95	5490	929	37	317	0	1831	2442	3052	3663	4273	1832	2442	3053	3663	4274		
25	900,02	900,01	73,26	900,02	78,25	6095	1162	37	1735	85	1824	2442	3052	3663	4273	1832	2442	3053	3663	4274		
26	900,04	1,69	637,76	900,03	5,24	7143	0	161	276	0	1816	2442	3053	3663	4273	1832	2442	3053	3663	4274		
27	410,83	900,05	8,70	99,35	900,03	4423	554	0	134	377	1821	2442	3052	3663	4273	1832	2442	3053	3663	4274		
28	900,04	40,93	4,62	2,17	32,35	8873	103	76	0	213	1947	2606	3257	3524	4560	1955	2606	3258	3909	4561		
29	900,03	1,46	900,04	0,59	10,97	7194	0	254	0	6	2079	2774	3467	4161	4854	2081	2774	3468	4161	4855		
Promedio	692,28	400,18	295,67	351,12	389,96	5420	337	170	327	293	1895	2533	3166	3709	4316	1900	2533	3167	3800	4434		