



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO  
Instituto Tecnológico de Cd.  
Victoria



# Instituto Tecnológico de Ciudad Victoria

**División de Estudios de Posgrado e Investigación**

## **TESIS**

**Programación de tareas en una línea de producción  
Utilizando búsqueda Tabú**

Presenta

**Ing. Martín Josué Castillo Montes**

Como requisito para la obtención del grado de  
**Maestro en Sistemas Computacionales**

Directora de tesis

**Dra. Adriana Mexicano Santoyo**

Codirectora de tesis

**Dra. Hilda Castillo Zacatelco**

Ciudad Victoria, Tamaulipas, México. **Mayo** de 2020.

Oficio de Aceptación de documento de tesis (escaneado)

Oficio Autorización de tesis (escaneado)

## **Dedicatoria**

Este trabajo está dedicado a mi familia, especialmente a mi padre y a mi esposa, que fueron pieza clave y un importante sustento moral y espiritual. Agradezco a Dios por las personas que fueron ese punto de encuentro en este espacio multifactorial que llamamos tiempo, que aportaron para realizar este proyecto de investigación y de vida por las experiencias adquiridas, a la Dra. Adriana, al Dr. Osvaldo, quienes encausaron las ideas, que son el punto de partida del conocimiento científico.

## **Agradecimientos**

Agradezco al Instituto Tecnológico de Ciudad Victoria, por las atenciones y apoyo prestados para llevar a cabo este recorrido de formación profesional, a la empresa *Springs Window Fashions* por proporcionar datos útiles para el estudio del problema y el diseño de su solución.

## RESUMEN

En este trabajo se aborda el problema *Single Machine Scheduling* el cual consiste en un solo recurso que atiende una lista de trabajos a ejecutar. Existen diferentes métodos según el objetivo a mejorar u optimizar. El problema que se atiende en este trabajo es el de la tardanza total ponderada o *total weighted tardiness problem*, donde se requiere reducir la tardanza total a una serie de trabajos y obtener una secuencia que elimine o reduzca la tardanza. Los primeros métodos para atender problemas de secuencia se basaban en reglas de ordenamiento y programación entera. Posteriormente, con la era informática, varios autores utilizaron enfoques de búsqueda local, heurísticas y metaheurísticas que van más allá de las reglas de despacho. En este trabajo se propone un algoritmo de Búsqueda Tabú que hace uso de una regla que selecciona como solución inicial la mejor solución obtenida al aplicar tres reglas de despacho. Dicho algoritmo se probó con instancias sintéticas de 40, 50 y 100 tareas. Los resultados mostraron que el algoritmo propuesto obtiene mejores resultados que los métodos tradicionales.

Después de validar el desempeño del algoritmo propuesto, se probó con datos reales de producción. Dichos datos muestran el panorama variante en el área de producción de una empresa, dónde siempre debe personalizarse y adaptarse una solución a las condiciones existentes. En el documento se muestra la aplicación del algoritmo propuesto en una jornada de trabajo, donde generalmente se aplica FIFO. Los resultados marcan una diferencia respecto a ordenamientos basados en reglas tradicionales, dado que el algoritmo desarrollado Tabú M, ofrece una mejora de casi 4 veces menos tardanza en los valores reales utilizados, respecto a los métodos basados únicamente en reglas de despacho. De tal forma que resulta una herramienta de utilidad en la planeación de tareas al reducir la tardanza total.

## **ABSTRACT**

This work addresses the Single Machine Scheduling problem which consists of a single resource that serves a list of jobs to run. There are different methods among the objective to improve or optimize. The problem addressed in this work is the total weighted tardiness or total weighted tardiness problem, where it is required to reduce the total tardiness of a job series and obtain a sequence that eliminates or reduces the delay. The earliest methods for deal with sequence problems were based on dispatch rules and integer programming. Subsequently, with the computational age, several authors used local, heuristic and metaheuristic search approaches that go beyond dispatch rules. In this work, a Tabu Search algorithm is proposed, this use a rule that selects as the initial solution the best solution obtained by applying three dispatch rules. This algorithm was tested with synthetic instances of 40, 50 and 100 tasks. The results showed that the proposed algorithm achieves better results than traditional methods.

After validating the performance of the proposed algorithm, it was tested with a real production data. These data show the changing panorama in the production area of a company, where a solution must always be customized and adapted to existing conditions. The document shows the application of the proposed algorithm in a work journal, where FIFO is generally applied. The results make a difference versus the systems based on traditional rules, since the algorithm developed Tabu M, offers an improvement of almost 4 times less delay in the real data used compared to methods based solely on dispatch rules. In such a way that it is a useful tool for scheduling and reducing the total tardiness.

## TABLA DE CONTENIDO

Capítulo 1.....	1
1.1 DESCRIPCIÓN DEL PROBLEMA.....	2
1.2 HIPÓTESIS DE LA INVESTIGACIÓN.....	3
1.3 OBJETIVO GENERAL.....	3
1.4 OBJETIVOS ESPECÍFICOS.....	3
1.5 JUSTIFICACIÓN.....	4
1.6 ALCANCES Y LIMITACIONES.....	5
1.6.1 ALCANCES.....	5
1.6.2 LIMITACIONES.....	5
1.7 ORGANIZACIÓN DEL DOCUMENTO.....	6
Capítulo 2.....	7
MARCO TEÓRICO.....	7
2.1 FUNDAMENTOS.....	8
2.1.1 CLASIFICACIÓN DE PROBLEMAS COMBINATORIOS SEGÚN SUS CARACTERÍSTICAS.....	8
2.1.2 PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA.....	9
2.1.3 PROGRAMACIÓN DE TAREAS.....	10
2.1.4 HEURISTICAS.....	10
2.1.5 BÚSQUEDA TABÚ.....	11
2.2.1 Tabla comparativa.....	16
Capítulo 3.....	18
3.1.1 INICIALIZACIÓN.....	19
3.1.2 SELECCIÓN.....	20
3.1.3 ACTUALIZACIÓN DE LA LISTA TABÚ.....	21
Capítulo 4.....	33
RESULTADOS EXPERIMENTALES.....	33
4.2.3 Simulación de una corrida de una jornada de trabajo.....	43
4.2.3 SIMULACIÓN DE UNA JORNADA DE TRABAJO COMPLETA.....	45
Capítulo 5.....	54
CONCLUSIONES.....	54
Anexos.....	57
Anexo A: Resultados de los conjuntos de instancias WT40, WT50 y WT100.....	57
REFERENCIAS.....	68



## LISTA DE TABLAS

Tabla 1 comparativa de los trabajos relacionados .....	16
Tabla 2 Lista de trabajos a realizar, los tiempos están expresados en días y el orden de los trabajos es dado en la manera que se reciben los pedidos .....	20
Tabla 3 Espacio de soluciones para la solución inicial .....	21
Tabla 4 Estructura de Lista Tabú .....	21
Tabla 5 Ejemplo de estructura de memoria de Búsqueda Tabú .....	22
Tabla 6 iteración 1 .....	22
Tabla 7 nuevo espacio de soluciones .....	23
Tabla 8 Estructura tabú de la Iteración 2 .....	23
Tabla 9 Espacio de soluciones de la iteración 3 .....	23
Tabla 10 Iteración número 3 .....	24
Tabla 11 Solución inicial aplicando la regla <i>EDD</i> .....	25
Tabla 12 Solución al realizar intercambio de tareas después de la regla <i>EDD</i> .....	25
Tabla 13 variables utilizadas .....	28
Tabla 14 función objetivo .....	29
Tabla 15 algoritmo .....	29
Tabla 16 Total de mejores resultados obtenidos por algoritmo .....	35
Tabla 17 Total de mejores resultados obtenidos por algoritmo .....	36
Tabla 18 Total de mejores resultados obtenidos por algoritmo .....	37
Tabla 19 óptimos encontrados por los algoritmo .....	38
Tabla 20 Número de mejores resultados encontrados .....	38
Tabla 21 secuencia de ejemplo .....	42
Tabla 22 trabajo reingresado a línea de producción por defecto o error de especificación .....	42
Tabla 23 tarea movida al inicio de la secuencia .....	42
Tabla 24 extracto de las últimas 10 tareas de una instancia real de 125 tareas .....	43
Tabla 25 Últimas 10 tareas de la secuencia + tareas agregadas .....	44
Tabla 26 solución obtenida por el algoritmo Tabú M .....	44
Tabla 27 Corrida 1 de 100 trabajos .....	45
Tabla 28 segunda corrida de 100 tareas + 5 tareas para solventar de la secuencia anterior .....	46
Tabla 29 tercera corrida de 100 tareas + 7 tareas .....	46
Tabla 30 cuarta corrida de 100 tareas + 7 tareas .....	47
Tabla 31 quinta corrida de 100 tareas + 9 tareas .....	48
Tabla 32 sexta corrida de 100 tareas + 8 tareas agregadas .....	48
Tabla 33 séptima corrida de 100 tareas +8 agregados .....	49
Tabla 34 octava corrida de 100 tareas + 8 tareas agregados .....	49
Tabla 35 novena corrida de 100 tareas + 10 tareas agregados .....	50
Tabla 36 décima corrida de 100 tareas + 9 tareas agregados .....	51
Tabla 37 Resultados de la simulación de 1000 tareas y 10 corridas .....	51
Tabla 38 Resumen de Resultados .....	52
Tabla 39 Tardanza obtenida por los diferentes algoritmos en la instancia real .....	53
Tabla 40 Resultados completos de la instancia WT40 .....	57
Tabla 41 Resultados completos de la instancia WT50 .....	60
Tabla 42 Resultados completos de la instancia WT100 .....	64

## LISTA DE FIGURAS

Figura 1 Asignación de tareas a un solo recurso.....	3
Figura 2 Solución inicial en forma de permutación natural o FIFO.....	20
Figura 3 Comparación de valores entre diferentes ordenamientos vs soluciones iniciales .....	27
Figura 4 Formato de instancias sintéticas .....	34
Figura 5 Resultados para el conjunto de instancias de 40 trabajos .....	35
Figura 6 Total de mejores resultados obtenidos por algoritmo.....	36
Figura 7 Resultados para el conjunto de instancias de 100 trabajos .....	37

# Capítulo 1

## INTRODUCCIÓN

---

La planeación y programación de tareas, es un proceso que busca asignar recursos de la mejor manera para atender requerimientos de demanda, ayudando así a que se alcancen los objetivos de alguna compañía [1]. A pesar de los esfuerzos que realizan las compañías para realizar los procesos de la mejor manera, es común encontrar dificultades, las cuales se reflejan cómo incumplimiento y retrasos de fechas de entrega, sobrecosto de producción, etc. Por lo tanto al no lograr las metas de corto plazo, se ven frustrados los objetivos de mediano y largo plazo. Una buena programación de tareas, refleja a corto plazo beneficios, sin embargo, muchas veces dicha programación se realiza de manera manual, lo cual resulta impráctico, debido a la cantidad de trabajos a programar y sus posibles combinaciones de secuencia. Este problema es conocido en la literatura cómo: *Single Machine Scheduling*, el cual ha sido abordado mediante reglas de despacho [2], metaheurísticas cómo *simulated annealing*, *dynasearch* y otras como se podrán encontrar en la recopilación realizada por Bo Chen en [3].

En particular este trabajo propone la implementación de un algoritmo de Búsqueda Tabú para secuenciar tareas en una empresa de manufactura de componentes electrónicos en México, con la finalidad de reducir los atrasos de los trabajos a ejecutar, ya que se ha demostrado en la literatura que puede obtener buenos

resultados en tiempos razonables [4], en comparación con la programación manual o por métodos exactos.

## 1.1 DESCRIPCIÓN DEL PROBLEMA

En las líneas de producción se procesan diferentes productos y tamaño de pedidos, lo cual requiere de una planeación para ejecutar una secuencia de trabajos. Cada trabajo a secuenciar tiene atributos como una fecha de entrega y tiempo de proceso, por lo tanto, el orden de la secuencia tiene efectos sobre la fecha de terminación de cada tarea, lo que define si el trabajo se entrega a tiempo o no.

La tarea de encontrar una secuencia que minimice el tiempo total de atraso y el número de trabajos atrasados es compleja. Ya que requiere calcular las  $n!$  combinaciones diferentes en que se pueden secuenciar los trabajos para procesarlos y elegir el mejor que minimice el atraso total. En la Figura 1 puede visualizarse cómo el orden de ejecución afecta la fecha de terminación de los trabajos. Los trabajos 1, 2 y 3, tienen asignadas las fechas de entrega (*Due Date*) los días 4, 9 y 8, respectivamente. El eje horizontal corresponde a las unidades de tiempo, el eje vertical “ordenamientos” muestra las reglas de prioridad FIFO (*First In First Out*), EDD (*Earliest Due Date*) y SPT (*Shortest Processing Time*).

El ordenamiento SPT asigna el tiempo más corto de proceso en orden ascendente, donde 2 trabajos se terminarían con atraso. FIFO corresponde a los trabajos atendidos en orden de llegada, donde existe un atraso en 3 trabajos, por lo que atender las tareas en orden de llegada no resulta adecuado. El ordenamiento EDD asigna la fecha más próxima de entrega en orden ascendente, donde todos los trabajos ejecutados se terminan a tiempo lo cual se observa en la Figura 1.

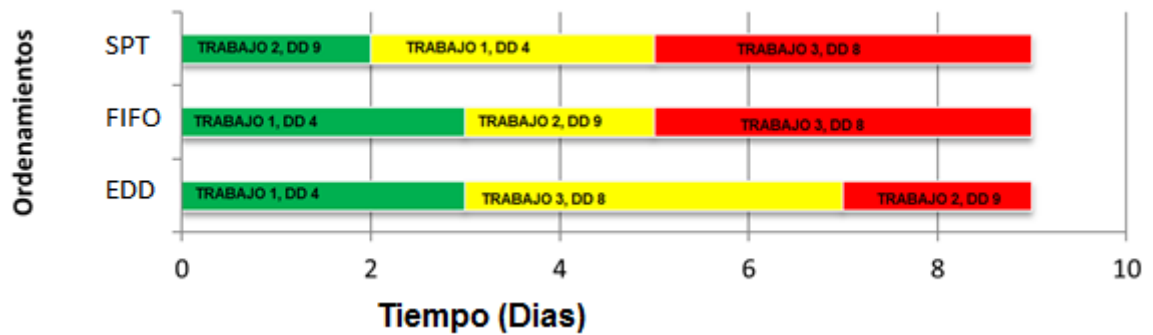


Figura 1 Asignación de tareas a un solo recurso

Es importante resaltar que para este ejemplo existen 6 posibles soluciones, si se tratase de un ejemplo de 5 existirían 120 posibles soluciones, observándose un incremento notable con sólo 2 tareas más a secuenciar.

## 1.2 HIPÓTESIS DE LA INVESTIGACIÓN

El uso de la metaheurística Búsqueda Tabú en el problema *Single Machine Scheduling* como una herramienta de planeación en una línea de producción de una empresa de fabricación de persianas, permitirá reducir el tiempo de terminación al menos un 5%.

## 1.3 OBJETIVO GENERAL

Desarrollar un algoritmo de Búsqueda Tabú, el cual hace uso de reglas de despacho, enfocado a reducir la Tardanza Total del problema *Single Machine Scheduling* en la empresa de manufactura.

## 1.4 OBJETIVOS ESPECÍFICOS

- Analizar el problema de planeación de tareas.
- Obtención de instancias de prueba utilizadas en la literatura *OR-Lib* [5].
- Implementación de ordenamientos mediante reglas de despacho para la generación de la solución inicial.

- Análisis de la metaheurística Búsqueda Tabú.
- Diseño de un algoritmo de búsqueda tabú con un método de selección de la mejor solución inicial con la finalidad de reducir el valor de tardanza de la función objetivo.
- Validación del funcionamiento del algoritmo, utilizando el conjunto de pruebas *OR-Lib*.
- Pruebas del algoritmo con datos reales de una empresa de fabricación de persianas.
- Comparación con otro algoritmo de búsqueda local variable de vecinos, la cual utiliza heurísticas similares a Tabú.

## 1.5 JUSTIFICACIÓN

En la literatura se ha demostrado que el problema *Single Machine Scheduling* puede tener características de tipo NP-hard [6]. En este trabajo se utiliza la Búsqueda Tabú para encontrar soluciones que permiten reducir el atraso total y el número de trabajos atrasados en una línea de producción. El uso de esta metaheurística ha demostrado su efectividad en el problema *Single Machine Scheduling*, el cual se aplica en las áreas de producción para mejorar su eficiencia.

El hecho de encontrar una secuencia eficiente que minimice el atraso total y el número de trabajos atrasados en una línea de producción, es una tarea compleja por el hecho de que a mayor número de tareas, aumenta considerablemente el número de combinaciones posibles y aún más complejo si se realiza de forma completamente manual.

La aplicación de las reglas de despacho no es suficiente porque ordenan de acuerdo a algún criterio como el menor tiempo de proceso o fecha más próxima de entrega los trabajos que se ejecutan, sin embargo, de manera general no toman en cuenta ambos factores y estos criterios son aplicados por el encargado de la producción, lo cual requiere tiempo considerable y es propenso a errores por la

cantidad de trabajos que se llega a procesar en cierto periodo de tiempo.

## 1.6 ALCANCES Y LIMITACIONES

Los alcances y limitaciones de este trabajo de investigación, corresponden a un problema de líneas de producción, en la cual se analizaron datos proporcionados por personal del área de producción de la empresa de manufactura.

### 1.6.1 ALCANCES

- La solución se centra en el desarrollo del algoritmo Búsqueda Tabú para resolver el problema *Single Machine Scheduling* con fechas de entrega y minimización de la Tardanza Total.
- La herramienta devuelve en forma de lista el orden de los trabajos a secuenciar.
- Se utilizaron las librerías *OR-Lib* de J. E. Beasley para el problema *Total Weighted Tardiness* para las pruebas del algoritmo.

### 1.6.2 LIMITACIONES

- El equipo utilizado para las pruebas del algoritmo es un portátil con procesador AMD A8, 4Gb de RAM y 500 GB de HDD.
- El tiempo de desarrollo del proyecto se limitó al tiempo de duración de la maestría que es de 2 años.
- Se proporcionó un conjunto de datos por personal de la empresa de manufactura dónde se procesan alrededor de 300 pedidos diarios en promedio, que corresponden en aproximadamente 2300 productos diarios que produce la planta.

## **1.7 ORGANIZACIÓN DEL DOCUMENTO**

El presente documento se estructura de la siguiente forma, en el Capítulo 2 se muestra el marco teórico que fundamenta los conceptos y enfoques de la solución propuesta al problema de la tardanza total. El Capítulo 3 muestra la estructura de la búsqueda tabú, su funcionamiento y su adaptación al problema de la tardanza total. El Capítulo 4, muestra los resultados de los experimentos realizados con el algoritmo a partir de instancias sintéticas y la solución propuesta para solucionar un problema de producción así como datos extraídos de la empresa. Por último, el Capítulo 5 contiene las conclusiones obtenidas del desarrollo del presente documento.



# Capítulo 2

## MARCO TEÓRICO

---

El problema *Single Machine Scheduling*, es un problema asociado a la programación de tareas, dónde un solo recurso procesa una serie de trabajos. Existen varios modelos, que según las necesidades de producción, sobre todo las de manufactura esbelta, penalizan la tardanza, por esta razón el presente trabajo trata el problema de tardanza total. Uno de los trabajos pioneros en tratar el problema con un enfoque metaheurístico de búsqueda tabú fue Manuel Laguna [7], quien define algunos métodos como heurísticas de intercambio e inserción, para su solución mediante este enfoque. Cabe mencionar que uno de los trabajos de la literatura que inspiraron la solución planteada es el realizado por Neelam Tyagi [8], quien realizó un análisis de 5 reglas de despacho para resolver el problema *Single Machine Scheduling with Total Weighted Tardiness* (TWT) mediante un algoritmo *Branch and Bound* (B&B).

En esta sección se describen los diferentes temas relacionados con el problema *Single Machine Scheduling*. En la sección 2.1 se describe los conceptos básicos para la comprensión de la problemática y los temas relacionados al tema de estudio. La sección 2.2 contiene los trabajos de investigación relacionados con el problema y las soluciones propuestas por la comunidad científica.

## 2.1 FUNDAMENTOS

A continuación se describen los fundamentos que dan sustento al trabajo de tesis desarrollado.

### 2.1.1 CLASIFICACIÓN DE PROBLEMAS COMBINATORIOS SEGÚN SUS CARACTERÍSTICAS

Según Richard M. Karp [9] una gran parte de los problemas implica la determinación de las propiedades de los grafos, dígrafos, enteros, arreglos de enteros, familias finitas de agrupaciones finitas, fórmulas booleanas y elementos de otros dominios contables, de tal manera que considerando lo anterior al codificar los problemas en palabras de un alfabeto finito, pueden ser convertidos en un lenguaje de reconocimiento de problemas para poder identificar su complejidad computacional.

Existen varios tipos de problemas los cuales se pueden clasificar por:

- Los que no tienen solución, estos casos son clasificados como problemas indecidibles (o imposibles de ser resueltos).
- Los que tienen solución algorítmica y que por lo tanto se pueden resolver paso a paso, codificando y aplicando los algoritmos para su resolución, ya sea en forma manual, ya sea con apoyo de equipos digitales.
- Y un tercer grupo que no pertenece a ninguno de los dos anteriores. En este grupo se consideran:
  - Aquellos en que la solución algorítmica tiene complejidad NP-completa.
  - Aquellos en que los seres humanos son capaces de resolver con mayores o menores dificultades, a pesar de que no pueden formularse algoritmos que permitan resolver la situación en forma más o menos automatizada.

### 2.1.2 PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA

La optimización combinatoria es una rama de la optimización en matemáticas aplicadas y en ciencias de la computación, relacionada a la investigación de operaciones, teoría de algoritmos y teoría de complejidad computacional [10].

En el área computacional se asocia generalmente a los problemas de tipo NP-hard, los cuales se pueden describir como aquellos que tienen problemas de decisión por lo menos tan difíciles como de tipo NP.

Una instancia de un problema de optimización combinatoria puede ser descrita formalmente como una tupla (lista ordenada de elementos),  $\{X, P, Y, f, extr\}$  donde:

- $X$  es el espacio de soluciones (en el cual  $f$  y  $P$  están definidos)
- $P$  es la factibilidad predicado.
- $Y$  es el conjunto de soluciones factibles.
- $f$  es la función objetivo.
- $extr$  es el extremo (normalmente min o max).

Expuesto lo anterior un problema de optimización consiste en minimizar o maximizar el valor de una variable. Considerando que la variable que se desea minimizar o maximizar debe ser expresada como función de las variables relacionadas en el problema, esto se conoce también como función objetivo. En ocasiones es preciso considerar las restricciones que se tengan en el problema, ya que éstas generan igualdades entre las variables que permiten la obtención de la función de una variable que se quiere minimizar o maximizar. Según la dificultad del problema y sus características se pueden clasificar en problemas conocidos, por ejemplo: *bin packing* (acomodo de paquetes en un contenedor), problema de la mochila (*knapsack*), problema del agente viajero (visita de nodos mediante el camino más corto), etc. [11].

### 2.1.3 PROGRAMACIÓN DE TAREAS

La programación de tareas, es una parte fundamental en los procesos de una empresa, porque mediante la programación, es posible estimar los tiempos de producción y entrega de los pedidos que se generan, minimizando así los retrasos. En la teoría clásica, se asumen tiempos de procesamiento fijos y valores constantes, en la práctica es posible encontrarse con el fenómeno de la deterioración lo cual impacta en los tiempos de procesado, además de los posibles caminos para programar tareas [12].

Existen varias reglas de prioridad en la programación de tareas, éstas se aplican según el objetivo a alcanzar o prioridad, por ejemplo; minimizar la cantidad de trabajos atrasados, minimizar el atraso promedio, minimizar el atraso máximo, minimizar el tiempo de flujo promedio, etc., sin embargo, el *makespan* o tiempo requerido para completar los trabajos será idéntico independiente de la regla de prioridad dado que es la suma del tiempo requerido para completar todas las tareas.

### 2.1.4 HEURISTICAS

Heurística es un término que se deriva de la palabra griega *heuriskein* la cual significa encontrar o descubrir. Sin embargo, científicamente se utiliza la palabra heurística para calificar un procedimiento al que se tiene un alto grado de confianza de su capacidad para encontrar buenas soluciones con un costo computacional razonable, aunque no se garantice un resultado óptimo e inclusive no se pueda establecer la cercanía a un resultado óptimo. Por lo tanto un procedimiento heurístico es una contraposición a un procedimiento exacto [13].

El uso de heurísticas se considera, cuando existen ciertos elementos en un problema de optimización, por ejemplo:

- Cuando no existe un método exacto para su resolución.
- Cuando existe un método exacto, pero el costo computacional es alto o inviable

- Si el método heurístico es más flexible que el método exacto y permite la incorporación de condiciones de difícil modelado
- El modelo matemático es muy grande, demasiado no lineal o muy complejo desde un punto de vista lógico

### 2.1.5 BÚSQUEDA TABÚ

La búsqueda tabú es una estrategia para resolver problemas de optimización combinatoria cuyas aplicaciones son aplicables en el rango de teoría de grafos, problemas de programación entera, etc., es un procedimiento adaptativo con la habilidad de hacer uso de muchos otros métodos, como algoritmos de programación lineal y heurísticas especializadas [7].

Para describir la búsqueda tabú se utiliza la siguiente notación

( $P$ ) Minimizar  $c(x)$ :  $x \in X$

Dónde  $x \in R$  el objetivo de la función  $c(x)$  puede ser lineal o no lineal, y la condición  $x \in X$  se asume para restringir componentes específicos de  $x$  a valores discretos, en algunos ajustes ( $P$ ) puede representar una forma modificada del problema original, dónde  $X$  es un superconjunto de los vectores que se consideran factibles,  $c(x)$  es una función de “castigo” diseñada para asegurar soluciones óptimas a ( $P$ ), las cuales también son óptimas para el problema del que se derivó [14].

La búsqueda tabú se puede definir de la siguiente manera: Un procedimiento de búsqueda local dotado de memoria para evitar óptimos locales que utiliza 2 filosofías de búsqueda; intensificación, la cual busca zonas con buenas soluciones y diversificación, la visita de zonas nuevas no exploradas.

Búsqueda Tabú hace uso de memoria adaptiva en función de tiempo, recordando los movimientos anteriores y deshabilitándolos temporalmente. Este tipo de memoria utiliza para restringir el acceso a una posible solución ya encontrada, evitando así un

ciclo en un óptimo local. Por lo tanto permite la búsqueda de soluciones en otro espacio, de tal manera que aumenta la posibilidad de encontrar una mejor solución que la que se encontró anteriormente.

La Búsqueda Tabú además utiliza diversas técnicas heurísticas, dependiendo del problema que se pretende resolver. Por ejemplo, el intercambio de 2 tareas, es una solución si representa una mejora que minimice la función objetivo, esto permite ir recorriendo el espacio de soluciones buscando opciones que sean factibles de acuerdo a los criterios que se pretendan minimizar.

A continuación se describen los elementos y los pasos que conforman esta metaheurística:

Una lista de candidatos, la cual puede ser generada de manera determinística o estocástica, es importante considerar que el tamaño de la lista define la eficiencia o la exactitud de los resultados dependiendo el tamaño de los elementos que conforman el problema.

Memoria de corto plazo, razón a la que se debe el nombre de la metaheurística, dado que es dónde se colocan los elementos tabú, con la finalidad de evitar óptimos locales y promover la exploración del espacio de soluciones.

Memoria de mediano plazo, es posible implementarla cómo complemento de la memoria de corto plazo con la finalidad de intensificar la exploración de áreas con potencial del espacio de soluciones, también llamado criterio de aspiración.

Memoria de largo plazo, generalmente hace uso de una estructura de frecuencia, contabilizando las veces que se visita un espacio de soluciones, dicha estructura funciona como un complemento de la memoria a corto plazo, lo cual permite la intensificación o la diversificación según se defina la estrategia:

- Cuando se trata de intensificación, dicha estructura de memoria considera un grupo de subconjuntos “Elite”, los cuales contienen movimientos con buenos atributos.
- Cuando se trata de diversificación, la estrategia promueve evitar los movimientos que fueron etiquetados como tabú para moverse hacia espacios intactos o poco frecuentados en búsqueda de soluciones.

Aunque en principio pareciera que son estrategias contrarias la diversificación y la intensificación, en realidad se complementan de diversas maneras: por ejemplo, la memoria a largo o mediano plazo puede tener etiquetados movimientos que se han realizado anteriormente aunque no se encuentren marcados como tabú (puede expresarse como una lista tabú complementaria o persistente) y movimientos poco o nada frecuentados, lo cual permitiría elegir según los criterios de diseño del algoritmo, visitar los espacios no frecuentados, logrando de esta manera expandir el espacio de movimientos en busca de buenas soluciones.

## 2.2 ESTADO DEL ARTE

A continuación se muestran algunos trabajos relacionados con la problemática abordada en este documento, los cuales proponen diferentes soluciones y métodos para minimizar la Tardanza Total del problema *Single Machine Scheduling*.

Richard K. Congram [15] en su trabajo *An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem*, propone una nueva técnica de búsqueda de vecindades llamada *Dynasearch*, la cual en lugar de realizar un solo movimiento por iteración, realiza varios movimientos a la vez obteniendo una mejora de alrededor del 5% para el problema *Single Machine Scheduling*.

Felipe Riquelme [16] propone en su trabajo Desarrollo de una solución logística para la programación de operaciones en una compañía siderúrgica, el uso del algoritmo de búsqueda tabú para la programación de tareas de las laminadoras de acero. En el cual pudo mejorar la programación de las tareas al no realizarse de manera manual por el encargado del departamento de producción, dicho algoritmo utiliza diversas estrategias como memoria atributiva (la cual guarda información del movimiento que generó una solución en lugar de la solución), el uso de *Path Relinking*, guardando una lista pequeña de las mejores soluciones las cuales se denominan Elite, las cuales después pueden ser tomadas como solución inicial para la búsqueda de buenas soluciones, los resultados obtenidos reducen el tiempo de planeación hasta en más de un 70%.

Manuel Laguna [7] en su trabajo *Tabu search methods for a Single Machine Scheduling*, propone 3 estrategias de búsqueda local para resolver el problema *Single Machine Scheduling*, se describe el método de *hill climbing*, intercambio de pares y movimientos de inserción, los cuales hacen uso de la estructura de memoria adaptiva de la búsqueda tabú, mostrando que TS se desempeña mejor que el algoritmo GRASP que utilizó en su comparativa, donde TS obtiene en 60 segundos una mejor solución, mientras que GRASP necesita 249 segundos para obtener un resultado muy similar.

Tamer Eren [17] en su trabajo *Minimizing total Tardiness in a scheduling problem with a learning effect*, propone el uso del efecto de aprendizaje en procesos repetitivos, lo cual minimiza el tiempo de ejecución.

Christos Koulamas [18] en su trabajo *the single-machine total tardiness scheduling problem: Review and extensions*, realiza una serie de revisiones (incluso extensiones) de varios desarrollos teóricos para el problema de la Tardanza Total, lo cual incluye algoritmos exactos, esquemas de aproximación completamente polinomiales, algoritmos heurísticos y casos especiales.



Tsung-Chyan Lai [19] en su trabajo *Minimizing Total Tardiness for Single Machine Sequencing*, propone una nueva regla a la cual denomina: MDD (*Modified Due Date*), mediante la cual asegura que es superior a ciertas heurísticas en el sentido del peor caso posible.

Álvarez-Valdés [20] en su trabajo Algoritmos exactos y heurísticos para minimizar el adelantamiento y el retraso ponderados en una máquina con fecha común, expone que el adelantamiento y el retraso en la ejecución de tareas, son efectos no deseables en la planificación JIT (*Just In Time*), considerando los costos que implican ambos para la industria.

Bo Chen [21] en su trabajo *A Review of Machine Scheduling: Complexity, Algorithms and Approximability*, expone los diferentes tipos de problemas para la secuenciación de tareas en una sola máquina, su complejidad, algoritmos enumerativos y de búsqueda local, en el caso particular de la Tardanza Total. Expone diferentes trabajos enfocados en la búsqueda local como métodos heurísticos, en los cuales se mencionan las metaheurísticas Recocido Simulado y Búsqueda Tabú.

Neelman Tyagi [22] en su trabajo *Single Machine Scheduling with Total Tardiness Problem*, propone 5 reglas de despacho y un algoritmo de ramificación y poda, con el objetivo de reducir la Tardanza Total y el número de trabajos atrasados, mostrando que la regla de despacho EDD, es la que resuelve mejor el problema que expone el trabajo.

Panneerselvam [23] en su trabajo *Simple heuristic to minimize total tardiness in a Single Machine Scheduling problem*, propone una heurística sencilla la cual también llaman *Greedy Heuristic* para minimizar la Tardanza Total. La solución obtenida comparada con una solución óptima demuestra que la diferencia es muy cercana a dicho valor.

### 2.2.1 Tabla comparativa

En la Tabla 1 se presenta la comparativa de los trabajos relacionados y las soluciones que proponen los autores para el problema *Single Machine Scheduling*, En la primera columna se enuncia el trabajo de publicación, en la columna 2 se muestran los algoritmos que se estudian, en la columna 3 si se hace uso de reglas de despacho (ordenamientos por prioridad) y por último una breve descripción de los resultados o motivo de estudio.

Tabla 1 comparativa de los trabajos relacionados

TRABAJO	ALGORITMO	REGLA DE DESPACHO	DESCRIPCIÓN
<i>An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem</i>	<i>Dynasearch</i>	no	realiza varios movimientos de búsqueda a la vez
Desarrollo de una solución logística para la programación de operaciones en una compañía siderúrgica	<i>Tabu Search</i>	no	Búsqueda Tabú con uso de <i>Path Relinking</i>
<i>Tabu search methods for a Single Machine Scheduling</i>	<i>Tabu Search</i>	no	Uso de una estrategia híbrida de intercambios e inserción de posiciones
<i>Minimizing total Tardiness in a scheduling problem with a learning effect</i>	<i>Tabu Search</i>	no	Uso del efecto de aprendizaje en procesos de repetición
<i>the single-machine total tardiness scheduling problem: Review and extensions</i>	describe varios métodos de solución	<i>EDD, DEC, MDD</i>	Propone modelos teóricos para la construcción de búsqueda local, además de heurísticas de descomposición
<i>Minimizing Total Tardiness for Single Machine Sequencing</i>	<i>MDD</i>	<i>EDD</i>	Se propone una nueva regla de despacho derivada del <i>EDD</i>
Algoritmos exactos y heurísticos para minimizar el adelantamiento y el retraso ponderados en una máquina con fecha común	no	no	Propone modelos para resolver diferentes casos
<i>A Review of Machine Scheduling: Complexity, Algorithms and Approximability</i>	Programación Dinámica, <i>Branch and Bound, Simulating Annealing, Threshold Accepting, Tabu Search &amp; G.A.</i>	<i>EDD, SPT.</i>	Se exponen varios algoritmos y el uso de reglas utilizados por varios autores.

<i>Single Machine Scheduling Model with Total Tardiness Problem</i>	<i>Branch and Bound</i>	<i>SPT, EDD, LPT, MST, FCFS</i>	Compara diversas reglas de despacho, desarrolla un algoritmo de ramificación y poda
<i>Simple heuristic to minimize total tardiness in a Single Machine Scheduling problem</i>	Greedy Heuristic	no	Propone un algoritmo simple que compara con una solución exacta mostrando su efectividad
Programación de tareas en una línea de producción aplicando búsqueda tabú	<i>Tabu Search</i>	<i>EDD, SPT, FIFO</i>	Uso de una regla de comparación de ordenamientos para elegir una solución inicial y búsqueda tabú

En la Tabla 1 puede observarse que en los trabajos que hacen uso de métodos heurísticos o metaheurísticos, no se hace uso de reglas de despacho para la generación de soluciones iniciales, en el caso de trabajo realizado por Tyagi [22], ella refiere que las reglas de despacho son una aproximación para solucionar el problema de la Tardanza Total, a partir de ahí compara el tiempo de ejecución y resultado del algoritmo *Branch and Bound*, donde se ejecuta mejor a partir de la regla de despacho EDD: a partir de este concepto, se construye la idea de proponer una solución basada en reglas de despacho como solución inicial para el algoritmo de búsqueda tabú desarrollado en este trabajo, dónde se generó una rutina que permite elegir entre las diferentes reglas de despacho, la que tiene menor valor de la función objetivo y a partir de esa solución inicial, se ejecuta el algoritmo búsqueda Tabú.

# Capítulo 3

## Algoritmo Búsqueda Tabú para Reducir la Tardanza Total

---

El problema *Single Machine Scheduling* (SMS), es un enfoque en el cual una sola máquina realiza una serie de trabajos y de acuerdo a la secuencia elegida, puede variar la cantidad de trabajos atrasados [24]. Por otra parte existen variantes del problema, como en este caso particular, enfocado en la tardanza total y la tardanza total ponderada, el cual busca reducir el atraso del tiempo de terminación de todos los trabajos. En la literatura se ha demostrado que el problema de la Tardanza Total es de tipo *NP-Hard* [1], lo cual justifica el uso de técnicas heurísticas o metaheurísticas, además que el uso de métodos exactos como *Branch and Bound*, cuando se trata de más de 50 tareas, el tiempo de cálculo se incrementa notablemente y difícilmente pueden obtenerse valores óptimos a partir de dicho número. Por esta razón el uso de búsqueda Tabú resulta eficiente para encontrar buenas soluciones. Para este trabajo se desarrolló una variante que utiliza un procedimiento de generación de soluciones iniciales basado en reglas de despacho, en el cual se selecciona el que tiene el menor valor de la función objetivo, con la finalidad de agrupar y minimizar el número de iteraciones para encontrar una buena solución.

El capítulo se divide en 2 secciones principales, la Sección 3.1 describe el funcionamiento de Búsqueda Tabú con un ejemplo motivacional. La Sección 3.2 describe la implementación del algoritmo desarrollado en este trabajo.

### 3.1 PROCEDIMIENTO DE BÚSQUEDA TABÚ

El procedimiento de Búsqueda Tabú consta de varias fases, las cuales se pueden resumir en 3 pasos: Inicialización, Selección y Actualización de la Lista Tabú. Estos pasos se describen en las Subsecciones 3.1.1, 3.1.2 y 3.1.3.

#### 3.1.1 INICIALIZACIÓN

Este paso consiste en inicializar una lista de la misma forma que se utilizaría con otro método de búsqueda local, generalmente se utiliza el conjunto de datos sin modificar su orden o por algún ordenamiento aleatorio, en el caso particular de este trabajo se utiliza una evaluación de la lista que corresponde a la regla de despacho FIFO, la cual no modifica el orden que se tiene de los trabajos, por lo tanto se atienden en orden de llegada, de ese mismo modo, se evalúa las reglas EDD y SPT, las cuales atienden por la prioridad del tiempo más próximo de entrega y el tiempo más corto de proceso respectivamente, cada ordenamiento se evalúa por medio de la función objetivo expresada en la ecuación 1:

$$f = \sum_{j=1}^n w_j [C_j - d_j]^+ \quad (1)$$

En la ecuación 1, la sumatoria de la secuencia de trabajos de 1 hasta n se evalúa multiplicando su peso o valor de penalización  $w$  por  $[C_j - d_j]^+$  la cual es equivalente a la función  $\max\{0, C_j - d_j\}$ ; dónde  $C_j$  es el tiempo de finalización de la tarea  $j$  y  $d_j$  que es su fecha de entrega, asimismo es importante señalar que  $C_j$  es igual al tiempo de procesamiento de la tarea  $j$  más la suma de los tiempos de procesamiento de todas las tareas que se realizan antes que  $j$ , por lo que su valor es acumulativo de acuerdo a la posición de una tarea en la lista.

En la Tabla 5 se muestra una secuencia FIFO dónde cada tarea tiene asignado un tiempo de procesamiento y una fecha establecida de entrega, para este tipo de problema se asume que todas las tareas llegan en tiempo 0 y una penalización igual a 1, el objetivo es minimizar el número de tareas atrasadas.

Tabla 2 Lista de trabajos a realizar, los tiempos están expresados en días y el orden de los trabajos es dado en la manera que se reciben los pedidos

Trabajo	Tiempo de proceso	Fecha de entrega
1	10	50
2	30	45
3	15	25
4	5	22
5	20	40

La solución inicial se representaría cómo un vector en la Figura 2:

1	2	3	4	5
---	---	---	---	---

Figura 2 Solución inicial en forma de permutación natural o FIFO

En la Figura 2 se asume que la secuencia es de tipo FIFO, a partir de esta solución inicial se calcula el valor de la función objetivo de acuerdo a la ecuación.

Si se sustituye los valores correspondientes a cada tarea en la secuencia se tendría para la función objetivo:

- $f = \{1 [10-50]^+ \} + \{1 [40-45]^+ \} + \{1 [55-25]^+ \} + \{1 [60-22]^+ \} + \{1 [80-40]^+ \}$
- $f = + 0 + 0 + 30 + 38 + 40$
- $f = 108$

Es necesario señalar que además, la función evalúa  $\max\{0, C_j - d_j\}$ , por lo que se tomarían solo los valores positivos diferentes a cero, por lo que el valor de la función objetivo sería de 108.

### 3.1.2 SELECCIÓN

Esta fase del algoritmo es dónde se ejecuta la búsqueda tabú, se realiza de manera iterativa la búsqueda local, en particular el algoritmo desarrollado hace uso de la heurística GPI (*Generalized Pairwise Interchange*) o intercambio de pares, dado su simplicidad de ejecución.

Búsqueda tabú utiliza métodos de generación de espacios de soluciones a partir de la solución inicial hasta la actual, en cada intercambio de pares se realiza el cálculo de la función objetivo, el valor obtenido se asocia a dicho movimiento, en la Tabla 4 puede observarse el valor de cada intercambio.

Tabla 3 Espacio de soluciones para la solución inicial

Tareas		$f$	Valor de movimiento
i	j		
1	2	108	0
1	3	83	-25
1	4	75	-33
1	5	123	15
2	3	88	-20
2	4	60	-48
2	5	83	-25
3	4	98	-10
3	5	118	10
4	5	123	15

En la Tabla 4 se observan ciertos movimientos con valores negativos, unos más que otros, esos valores indican la diferencia entre el valor de la solución inicial respecto al valor de intercambio, por lo que entre mayor sea, indica que es un mejor candidato para construir la nueva solución. Una vez que se selecciona el candidato se genera la nueva solución y el movimiento se mueve a una lista de movimientos recientes, donde se prohíbe dicho movimiento por cierta cantidad de iteraciones.

### 3.1.3 ACTUALIZACIÓN DE LA LISTA TABÚ

En esta fase, después de elegir un movimiento de intercambio, se agrega a la lista tabú, dicho movimiento se prohíbe por cierto número de iteraciones y se decrementa en uno por cada nueva iteración, transcurrido el periodo de prohibición, puede realizarse nuevamente dicho intercambio.

Tabla 4 Estructura de Lista Tabú

	2	3	4	5
1				
	2		3	
		3		
			4	

La Tabla 5 describe la estructura de movimientos marcados como tabú, que en este caso corresponde al intercambio de posición de la tareas 2 y 4, que en la Tabla 6 tiene un valor de -48, siendo el mejor candidato disponible, al seleccionar

el movimiento se agrega a la lista y se prohíbe en este caso por 3 iteraciones.

Es necesario señalar que un elemento marcado como tabú, no significa que bajo ninguna circunstancia pueda ser utilizado, en este punto puede usarse un criterio de aspiración, el cual consiste en alguna regla de selección de los elementos disponibles candidatos a una nueva solución, por ejemplo: el elemento más reciente o antiguo agregado a la lista tabú, el mejor candidato de la lista disponible, etc.

A continuación se mostrará una serie de iteraciones usando la búsqueda tabú:

Tabla 5 Ejemplo de estructura de memoria de Búsqueda Tabú

1		2	3	4	5		
2	1					2-4	-48
3		2				1-4	-33
4			3			1-3	-25
5				4		2-5	-25

En la Tabla 6 se muestra en la columna 1 la solución inicial, la segunda estructura corresponde a la memoria tabú, mientras que la tercera estructura corresponde a los primeros 5 movimientos candidatos, en la primera columna se muestran los intercambios por pares de tareas y la segunda estructura tiene los valores que corresponden a cada movimiento de intercambio.

Tabla 6 iteración 1 con el primer intercambio de tareas, el cual se encuentra marcado en negritas.

1		2	3	4	5		
<b>4</b>	1					<b>2-4</b>	<b>-48</b>
3		2		<b>3</b>		1-4	-33
<b>2</b>			3			1-3	-25
5				4		2-5	-25

Al realizar el primer intercambio tomando en cuenta el valor más negativo, la función objetivo pasa a tener un valor de 60 y se marca cómo tabú el movimiento realizado durante 3 iteraciones, de igual manera que con la solución inicial, el espacio de soluciones cambia sus valores para cada movimiento que se realiza y se genera una nueva lista de candidatos para la nueva función objetivo.



Tabla 7 nuevo espacio de soluciones

Tareas		$f$	Valor de movimiento
i	j		
1	5	60	0
1	3	55	-5
1	2	88	28
1	5	73	13
4	3	63	3
4	2	108	48
4	5	108	48
3	2	75	15
3	5	75	15
2	5	50	-10

En la Tabla 8 se muestra el nuevo espacio de soluciones las cuales definen la siguiente iteración y su estructura tabú:

Tabla 8 Estructura tabú de la Iteración 2

1		2	3	4	5		
4	1						2-5 -10
3		2		2	3		1-3 -5
5			3				1-5 0
2				4			4-3 3

En la Tabla 9 se muestra la nueva solución con su movimiento respectivo, puede observarse que el movimiento anterior 1-4 se le resta 1 iteración en la memoria tabú y se agrega el movimiento 2-5 marcado como tabú por 3 iteraciones, así mismo la nueva función objetivo toma el valor de 50.

Tabla 9 Espacio de soluciones de la iteración 3

Tareas		$f$	Valor de movimiento
i	j		
1	4	50	0
1	3	45	-5
1	5	53	3
1	2	98	48
4	3	53	3
4	5	83	33
4	2	123	73
3	5	60	10
3	2	80	30
5	2	60	10

En la Tabla 9 se muestra el espacio de soluciones para la tercera iteración la cual queda como se muestra en la tabla 10

Tabla 10 Iteración número 3

3		2	3	4	5		
4	1		3			1-3	-5
1		2		1	2	1-4	0
5			3			1-5	3
2				4		4-3	3

En la Tabla 10 se muestra el movimiento realizado para la tercera iteración, la cual produce una mejora al cambiar la función objetivo a 45, al igual que la iteración anterior se agrega un nuevo movimiento tabú que corresponde al movimiento realizado, además puede observarse que los movimientos tabú anteriores se decrecienta en 1 su periodo con cada iteración.

### 3.1.4 CONDICIÓN DE PARADA

La condición de parada puede establecerse mediante 3 criterios, uno es mediante cierta cantidad de movimientos que no produzcan una mejora, por un periodo de tiempo establecido o un número de iteraciones predefinido, este último caso, es el utilizado en el algoritmo desarrollado, la solución óptima para el problema de ejemplo es de un valor de 35.

## 3.2 SOLUCIÓN AL PROBLEMA DE LA TARDANZA TOTAL

En la sección 3.1, se describe el procedimiento de solución para el problema *Single Machine Scheduling with Total Weighted Tardiness*, aplicando búsqueda tabú, así mismo en la sección 1.1, se mostró que las reglas de despacho generan soluciones aceptables, aunque depende de ciertas variaciones en los problemas de la vida real, dado que son procesos estocásticos, puede existir variación en la tardanza, el tiempo de proceso o sus fechas de entrega. Dado este fenómeno, ciertas reglas pueden funcionar mejor que otras a pesar que su propósito sea diferente, cómo el caso de la regla SPT que da prioridad al tiempo más corto de proceso con la finalidad de reducir el tiempo de flujo, mientras que la regla EDD busca reducir la tardanza dando prioridad a la fecha más próxima de entrega.

En este trabajo se diseñó un algoritmo dónde se comparan los valores de la función objetivo para 3 reglas, FIFO, SPT y EDD, a continuación se muestra la utilidad del uso de una regla de despacho EDD, dado que se utiliza para reducir la tardanza, con el propósito de acotar el espacio de búsqueda de buenas soluciones del algoritmo.

**Tabla 11 Solución inicial aplicando la regla EDD**

Trabajo	Tiempo de proceso	Fecha de entrega	Tiempo de flujo	Atraso
4	5	22	5	0
3	15	25	20	0
5	20	40	40	0
2	30	45	70	25
1	10	50	80	30

Utilizando el ejemplo de la Tabla 5, dónde el orden es FIFO, al aplicar un ordenamiento basado en EDD, como en la Tabla 12 se observa que se reduce el número de tareas en esa solución inicial con un valor de 30, pero al generar un movimiento de intercambio en las tareas 2 y 3 de la secuencia se obtiene una mejora como se observa en la Tabla 12

**Tabla 12 Solución al realizar intercambio de tareas después de la regla EDD**

Trabajo	Tiempo de proceso	Fecha de entrega	Tiempo de flujo	Atraso
4	5	22	5	0
3	15	25	20	0
5	20	40	40	0
1	10	50	50	0
2	30	45	80	35

En la Tabla 12 se observa la mejora obtenida, además se obtiene la solución en una sola iteración, por lo que se reduce la cantidad de cómputo necesario para obtener una buena solución.

El fenómeno anterior se observó al momento de generar experimentos con las instancias obtenidas de la librería OR-Lib, dónde se comportaban de manera

distinta cada regla produciendo desviaciones en ciertas instancias por sus características.

El uso de las reglas de despacho, en la literatura estudiada, no es utilizado en algoritmos metaheurísticos, pero en el caso particular del trabajo de Neelam Tyagi hace uso de dichas reglas en un algoritmo exacto, en [22] se observa que EDD obtiene el mejor resultado en la mayoría de sus experimentos, por lo cual, se utiliza como una solución inicial para ejecutar un algoritmo *Branch and Bound* con el objetivo de reducir el tiempo de ejecución.

En el caso particular de este trabajo, se observó que la regla EDD, sobre todo al usarlo como solución inicial con búsqueda tabú, encontraba una cantidad mayor de soluciones óptimas. Sin embargo, en algunas instancias, se producían desviaciones considerables.

Al probar diferentes reglas, se observó que en ciertas instancias se producía un mejor resultado, lo que motivó a generar un algoritmo de selección y generación de soluciones iniciales, el cual genera ordenamientos usando 3 reglas de prioridad, FIFO, EDD, SPT. Para cada una de las reglas se genera su valor de tardanza con la función objetivo de la ecuación 1, después se comparan sus valores y se elige el de menor valor como solución inicial.

En la Figura 3 se observa una gráfica que muestra con líneas punteadas los valores obtenidos al ejecutar la instancia WT100, aplicando las reglas de prioridad EDD, SPT y FIFO, las líneas sólidas representan los resultados obtenidos después de aplicar búsqueda Tabú después de cada ordenamiento inicial.

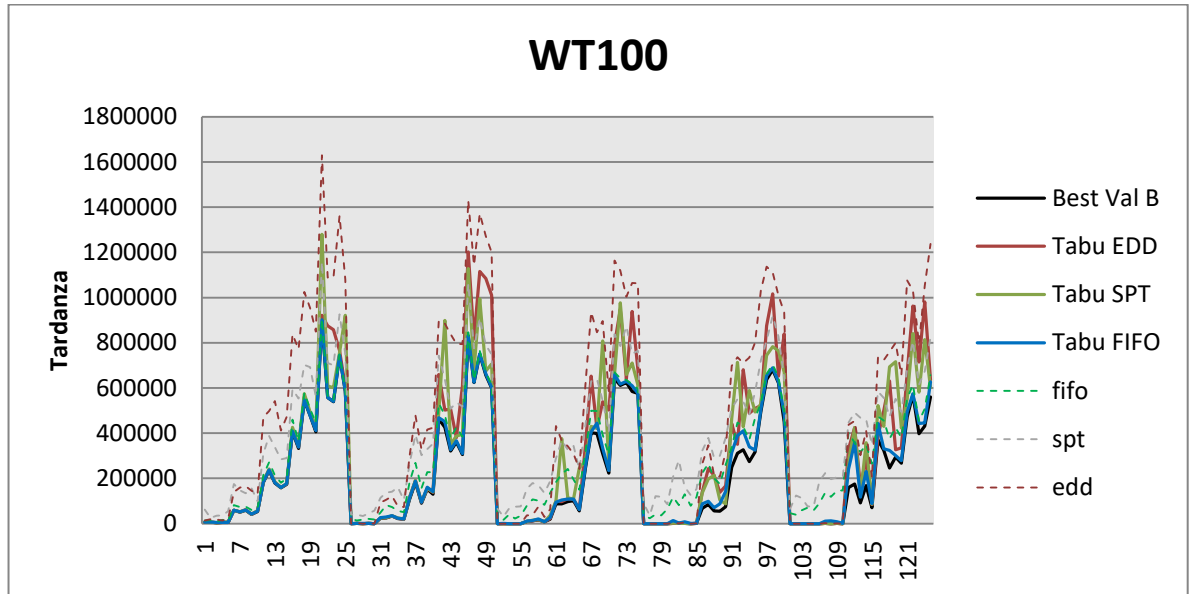


Figura 3 Comparación de valores entre diferentes ordenamientos vs soluciones iniciales

En la Figura 3 puede observarse cómo se reduce la tardanza mediante el uso de búsqueda tabú a partir de cualquier regla, no obstante se siguen observando desviaciones, esto debido a las distribuciones usadas para cada instancia, dónde los valores varían para la tardanza, fecha de entrega y tiempo de proceso.

El algoritmo desarrollado en el presente trabajo calcula el valor de la función para cada las diferentes reglas de despacho y al final se compara el valor obtenido de la función objetivo eligiendo el menor de todos, a partir de ahí se inicia el procedimiento iterativo de búsqueda de soluciones. En ese procedimiento iterativo se encuentra la lista tabú y los procedimientos de verificación de las listas para su correcta ejecución.

El algoritmo utiliza una estrategia agresiva *Best Improve Descent*, aunque es conocido además que en algoritmos de búsqueda local se obtienen mejores resultados mediante *First Improve Descent*. En el caso de búsqueda tabú por su memoria atributiva y en la parte experimental realizada en este trabajo, se observó que tiene mejor comportamiento la primera estrategia.

Se utiliza un número de iteraciones prohibidas o tabú pequeño, en este caso de 15 iteraciones, ya que los mejores resultados se obtienen con dicho valor. Además la

lista de candidatos usa un número pequeño de 10 elementos, lo cual reduce el número de ciclos de intercambio por iteración.

Una de las características de TS es el uso de movimientos de no mejora, en el caso particular de la estrategia diseñada, hace uso de movimientos candidatos con valores menores a la solución actual solamente.

La condición de parada del algoritmo consiste en una corrida de 300 iteraciones, ya que en los diversos experimentos se encontró que dicho valor obtiene las mejores soluciones y a partir de ahí ya no se generan mejoras.

La Tabla 13 muestra las variables utilizadas para ejecutar el algoritmo, principalmente se utilizan estructuras de tipo vector, esto con la finalidad de simplificar los movimientos por pares del algoritmo.

Tabla 13 variables utilizadas

<i>contador</i>	Índice auxiliar para contabilizar elementos
<i>i</i>	Contador para ciclo for
<i>f</i>	Función objetivo
<i>newf</i>	Variable que actualiza el valor de la función objetivo
<i>C</i>	Tiempo de proceso para la tarea <i>i</i>
<i>a</i>	Variable auxiliar para índices de intercambio
<i>tp</i>	Valor del periodo tabú
$vPt = \{t_1, \dots, t_n\}$	Vector que contiene los tiempos de proceso <i>t</i> de cada tarea, donde el valor se asocia según su posición de entrada del archivo contenedor de las instancias que va desde 1 hasta <i>n</i> , el cual se observa en la sección 4.1
$vWj = \{w_1, \dots, w_n\}$	Vector que contiene los pesos <i>w</i> de cada tarea, donde el valor se asocia según su posición de entrada del archivo contenedor de las instancias que va desde 1 hasta <i>n</i> , el cual se observa en la sección 4.1
$vDd = \{d_1, \dots, d_n\}$	Vector que contiene las fechas de entrega <i>d</i> de cada tarea, donde el valor se asocia según su posición de entrada del archivo contenedor de las instancias que va desde 1 hasta <i>n</i> , el cual se observa en la sección 4.1
$vTask = \{j_1, \dots, j_n\}$	Vector que contiene el índice de las tareas en el caso de las instancias sintéticas, en las instancias reales se sustituye por el número de orden

La Tabla 14 muestra la función objetivo, la cual acumula el valor de tardanza de cada tarea, que es equivalente a la Tardanza Total Ponderada.

Tabla 14 función objetivo

Función	Descripción
$f \leftarrow f + (Wj_i * (\max(0, (c - Dd_i))))$	Acumula el valor de tardanza de cada tarea $i$ en $f$ devolviendo el valor de tardanza total, $W$ corresponde al peso asignado al trabajo $j$

En la Tabla 15 se muestra el algoritmo desarrollado en pseudocódigo.

Tabla 15 algoritmo

Algoritmo: Búsqueda Tabú con regla de selección de solución inicial	
Entrada: $vPt$ , $vWj$ , $vDd$	
Salida: $vTask$ , $newf$	
1.	INICIO
2.	$n \leftarrow 0$
3.	$C \leftarrow 0$
4.	$newf \leftarrow 0$
5.	$vPt \leftarrow \{\}$
6.	$vWj \leftarrow \{\}$
7.	$vDd \leftarrow \{\}$
8.	$vTask \leftarrow \{\}$
9.	$vPt_i \leftarrow \{t_i, \dots, t_n\}$
10.	$vWj_i \leftarrow \{w_i, \dots, w_n\}$
11.	$vDd_i \leftarrow \{d_i, \dots, d_n\}$
12.	$n \leftarrow$ tamaño $vPt$
13.	$i \leftarrow 0$
14.	Para $i \leftarrow 0$ hasta $n$
15.	$C \leftarrow C + (vPt_i)$
16.	$f \leftarrow f + (Wj_i * (\max(0, (c - Dd_i))))$
17.	$vTask \leftarrow i$
18.	$newf \leftarrow f$
19.	Fin para
20.	$C \leftarrow 0$
21.	$f \leftarrow 0$
22.	Ordena $vPt$ de menor a mayor
23.	Para $i \leftarrow 0$ hasta $n$
24.	$C \leftarrow C + (vPt_i)$
25.	$f \leftarrow f + (Wj_i * (\max(0, (c - Dd_i))))$
26.	Si $newf \leftarrow f$
27.	entonces $newf \leftarrow f$
28.	Fin Si
29.	Fin para

30.	$C \leftarrow 0$
31.	$f \leftarrow 0$
32.	Ordena $vDd$ de menor a mayor
33.	Para $i \leftarrow 0$ hasta $n$
34.	$C \leftarrow C + (vPt_i)$
35.	$f \leftarrow f + (Wj_i * (\max(0, (c - Dd_i))))$
36.	Si $newf \leftarrow f$
37.	entonces $newf \leftarrow f$
38.	Fin Si
39.	Fin para
40.	$C \leftarrow 0$
41.	$f \leftarrow 0$
42.	$listatabu \leftarrow \{ \}$
43.	$listacandidatos \leftarrow \{ \}$
44.	$tp \leftarrow \{ \}$
45.	$a \leftarrow 0$
46.	Para $i \leftarrow 0$ hasta $i \leftarrow 300$ //iteraciones preestablecidas
47.	Si $listatabu$ no está vacía entonces
48.	Para cada $x \in listatabu$
49.	Decrementa $tp_x$
50.	Si $tp_x \leq 0$ ,
51.	elimina $listatabu_x$
52.	Fin si
53.	Fin para
54.	Fin si
55.	Si $listacandidatos$ no está vacía
56.	Ordena $listacandidatos$ menor a mayor
57.	Para cada $y \in listacandidatos$
58.	compara si cada $x \in listatabu$
59.	Si $x_i = y_i$
60.	entonces $a \leftarrow a+1$
61.	Fin Si
62.	Si $a =$ número de elementos de $listacandidatos$
63.	entonces $a \leftarrow 0$
64.	Fin Si
65.	Fin compara
66.	Fin para
67.	$vTask_a \leftarrow Listacandidatos_a$
68.	$vPt_a \leftarrow Listacandidatos_a$
69.	$vWj_a \leftarrow Listacandidatos_a$
70.	$vD_a \leftarrow Listacandidatos_a$
71.	$listatabu \leftarrow listacandidatos_a$
72.	$tp_a \leftarrow 7$
73.	Fin si
74.	$listacandidatos \leftarrow \{ \}$
75.	$aux \leftarrow 0$
76.	Para $i \leftarrow 0$ hasta $n$
77.	$aux \leftarrow vTask\{j_i\}$



78.	$vTask\{j_i\} \leftarrow vTask\{j_{i+1}\}$
79.	$vTask\{j_{i+1}\} \leftarrow aux$
80.	$aux \leftarrow vPt\{t_i\}$
81.	$vPt\{t_i\} \leftarrow vPt\{t_{i+1}\}$
82.	$vPt\{t_{i+1}\} \leftarrow aux$
83.	$aux \leftarrow vWj\{w_i\}$
84.	$vWj\{w_i\} \leftarrow vWj\{w_{i+1}\}$
85.	$vWj\{w_{i+1}\} \leftarrow aux$
86.	$aux \leftarrow vDd\{d_i\}$
87.	$vDd\{d_i\} \leftarrow vDd\{d_{i+1}\}$
88.	$vDd\{d_{i+1}\} \leftarrow aux$
89.	$C \leftarrow C + (vPt_i)$
90.	$f \leftarrow f + (Wj_i * (\max(0, (c - Dd_i))))$
91.	Si $newf \leftarrow f$
92.	$listacandidatos \leftarrow x$
93.	Fin Si
94.	$aux \leftarrow vTask\{j_{i+1}\}$
95.	$vTask\{j_{i+1}\} \leftarrow vTask\{j_i\}$
96.	$vTask\{j_i\} \leftarrow aux$
97.	$aux \leftarrow vPt\{t_{i+1}\}$
98.	$vPt\{t_{i+1}\} \leftarrow vPt\{t_i\}$
99.	$vPt\{t_i\} \leftarrow aux$
100.	$aux \leftarrow vWj\{w_{i+1}\}$
101.	$vWj\{w_{i+1}\} \leftarrow vWj\{w_i\}$
102.	$vWj\{w_i\} \leftarrow aux$
103.	$aux \leftarrow vDd\{d_{i+1}\}$
104.	$vDd\{d_{i+1}\} \leftarrow vDd\{d_i\}$
105.	$vDd\{d_i\} \leftarrow aux$
106.	Fin para
107.	Fin para
108.	Imprime $vTask$
109.	Imprime $newf$
110.	Fin

En la Tabla 15 se muestra la estructura lógica del algoritmo desarrollado, desde la primera línea hasta la 11, se asigna el tamaño de la instancia en la variable *inst*, en los vectores *vPt*, *vWj*, *vDD*, se guarda la información relacionada con el tiempo de proceso, la ponderación y la fecha de entrega, respectivamente, además se inicializan las variables y vectores necesarios. La línea 12 calcula el tamaño del vector para ejecutar los movimientos de los ciclos for que calculan la función objetivo, de acuerdo al tamaño del vector que contiene las tareas a ejecutar asignando su valor a *n*.

Las líneas 14 hasta la 19 corresponden al primer ordenamiento FIFO al cual se le asignan índices para identificar las tareas en orden ascendente. Desde la línea 22 a la 39 ordena de manera ascendente las tareas de acuerdo al tiempo de proceso o SPT para obtener el valor de tardanza, si es menor, se sustituye, si no es así, ordena las tareas por fecha de entrega o EDD para comparar el valor de tardanza y si es menor se sustituye, en caso contrario se mantiene la solución anterior que represente al menor valor de la función objetivo.

Las líneas 40 hasta la 106, corresponden a la estructura de búsqueda tabú, la cual se ejecuta en un ciclo de 300 iteraciones, de la línea 43 a la 53, comprueba si la lista tabú contiene elementos, si es el caso, decrementa el periodo tabú en 1 para cada trabajo contenido en la lista, si en algún elemento  $tp$  es igual a 0, se elimina de la lista tabú, pudiendo utilizarse nuevamente, el criterio de aspiración utilizado es el elemento más reciente en la lista y en dado caso que no existan candidatos alternativos, se utiliza el primero encontrado.

De la línea 54 a la 72, comprueba si existen elementos en la lista de candidatos, si es así ordena las tareas en la lista de acuerdo al valor de función objetivo que representa, de menor a mayor, dado que si su valor es más negativo representa una mejor solución. Entonces se busca en la lista tabú, si no es tabú se elige como solución, en caso contrario, se busca otro elemento que no se encuentre.

Las líneas 73 a la 105, corresponde a la rutina para crear el espacio de soluciones, dónde se intercambian pares y con cada intercambio se calcula la función objetivo, si el valor del movimiento produce un valor menor de la función objetivo, se agrega el movimiento a la lista de candidatos, cuyos elementos se vacían cada iteración antes de iniciar el ciclo tabú.

Las líneas 107 y 108, imprimen la secuencia obtenida mostrando el vector  $vTask$  y el valor de tardanza obtenido al terminar de ejecutarse el algoritmo.

# Capítulo 4

## RESULTADOS EXPERIMENTALES

---

El uso de procedimientos metaheurísticos, permite encontrar buenas soluciones en un tiempo de ejecución mucho menor que con procedimientos exactos, sobre todo en el área de la combinatoria aunque no garanticen resultados óptimos [24].

En esta sección se muestran los resultados del algoritmo, en la sección 4.1, se muestran las gráficas de los resultados obtenidos para cada instancia de 40, 50 y 100 trabajos, respectivamente. La sección 4.2 expone la aplicación en un caso real aplicado en la secuencia de una línea de producción de una empresa dedicada a la fabricación de persianas.

### 4.1 INSTANCIAS SINTÉTICAS OR-LIB

Estas instancias utilizadas se componen de 3 conjuntos para 40, 50 y 100 trabajos, cada conjunto contiene 125 problemas diferentes, los cuales varían su dificultad de acuerdo al factor de tardanza, además de la fecha relativa de entrega la cual viene dada por el conjunto {0.2, 0.4, 0.6, 0.8 y 1.0}, entonces el conjunto de problemas consiste en 5 instancias de tamaño, 40, 50 o 100 trabajos por cada

combinación del conjunto de factor de tardanza y el conjunto de fecha relativa de entrega.

El formato comprende una serie de valores listados uno después de otro, por ejemplo, para el archivo de instancias de 40 trabajos, los primeros 40 valores enteros son los tiempos de proceso para cada trabajo en la primera instancia de las 125 disponibles en el archivo, los siguientes 40 enteros son los pesos y los siguientes 40 valores corresponden a la fecha de entrega, según la descripción del autor.

Como puede observarse en la Figura 4, los valores se encuentran listados en forma de fila, consecutivos y separados por espacios.

26	24	79	46	32	35	73	74	14	67	86	46	78	40	29	94	64	27	90	55
35	52	36	69	85	95	14	78	37	86	44	28	39	12	30	68	70	9	49	50
1	10	9	10	10	4	3	2	10	3	7	3	1	3	10	4	7	7	4	7
5	3	5	4	9	5	2	8	10	4	7	4	9	5	7	7	5	10	1	3
1588	1620	1731	1773	1694	1487	1566	1844	1727	1636	1599	1539	1855	1645	1709	1660	1582	1836	1484	1559
1772	1510	1512	1795	1522	1509	1598	1658	1826	1628	1650	1833	1627	1528	1541	1497	1481	1446	1579	1814
56	25	76	35	28	52	21	32	64	67	48	100	94	87	39	18	78	80	56	72
4	70	36	46	85	31	96	30	66	92	33	18	19	34	18	4	42	94	4	89
1	9	9	9	5	1	4	3	2	8	2	3	7	5	3	5	2	2	2	4
9	9	6	8	9	7	5	2	1	6	4	6	1	2	10	10	6	5	4	3
1687	1738	1663	1480	1504	1826	1722	1660	1594	1445	1704	1660	1715	1701	1679	1516	1658	1611	1502	1685
1614	1647	1689	1615	1524	1800	1654	1752	1456	1452	1801	1713	1761	1513	1759	1484	1821	1448	1666	1611

Figura 4 Formato de instancias sintéticas

Las instancias y su descripción, para el problema *Single Machine Scheduling with total weighted tardiness* pueden encontrarse en [25].

En las Subsecciones 4.1.1, 4.1.2 y 4.1.3 se muestran los resultados obtenidos para cada conjunto de instancias.

#### 4.1.1 CONJUNTO DE INSTANCIAS WT 40

En esta sección se muestran los resultados para las 125 instancias de 40 trabajos cada una, en la Figura 5 se muestran las gráficas comparativas entre búsqueda tabú con cada regla de despacho de manera aislada y la versión mejorada con la regla de selección de soluciones iniciales.

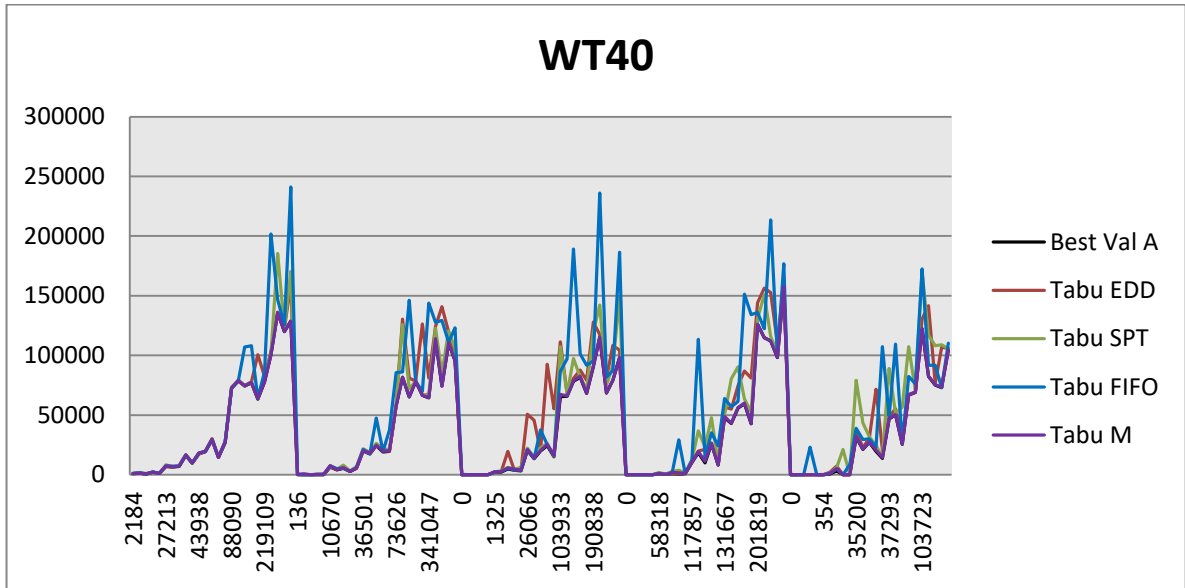


Figura 5 Resultados para el conjunto de instancias de 40 trabajos

En la Figura 5 se observa el comportamiento del algoritmo mejorado Tabú M respecto al algoritmo Tabú basado en una sola regla de despacho, se observa que Tabú SPT tiende a desviarse menos respecto a las otras 2 reglas de despacho, el algoritmo mejorado en cambio muestra aún menor desviación, encontrando en este caso 24 óptimos y 88 mejores resultados que los algoritmos tabú con una sola regla de ordenamiento, los mejores resultados se comparan con Best ValA que son los valores óptimos y mejores valores reportados.

En la Tabla 16 se muestran los mejores valores obtenidos para cada algoritmo respecto a los otros ejecutados, así como la cantidad de óptimos encontrados para el conjunto WT40, en cual contiene 125 instancias. La columna izquierda debajo de cada algoritmo contiene los mejores valores y la derecha los óptimos obtenidos.

Tabla 16 Total de mejores resultados obtenidos por algoritmo

Tabu FIFO		Tabu SPT		Tabu EDD		TabuM	
29	17	16	11	42	23	88	23

Los valores de la Tabla 16 son comparativos, estos corresponden a la cantidad de valores menores respecto a los obtenidos por las diferentes versiones de tabú probadas, a fin de medir la eficiencia del algoritmo mejorado.

### 4.1.2 CONJUNTO DE INSTANCIAS WT 50

En esta sección se muestran los resultados obtenidos para las 125 instancias de 50 trabajos cada una. En la gráfica de la Figura 5 se observan entre Tabú con las soluciones iniciales EDD y SPT, dónde FIFO muestra la mayor desviación, mientras que la versión mejorada Tabú M obtiene 84 mejores resultados y 22 óptimos alcanzados respecto a los algoritmos con una sola regla.

La Tabla 17 muestra los mejores valores obtenidos para cada algoritmo respecto a los otros ejecutados, así como la cantidad de óptimos encontrados para la instancia WT50. La columna izquierda debajo de cada algoritmo contiene los mejores valores y la derecha los óptimos obtenidos.

Tabla 17 Total de mejores resultados obtenidos por algoritmo

Tabu FIFO		Tabu SPT		Tabu EDD		TabuM	
25	14	23	11	40	21	84	21

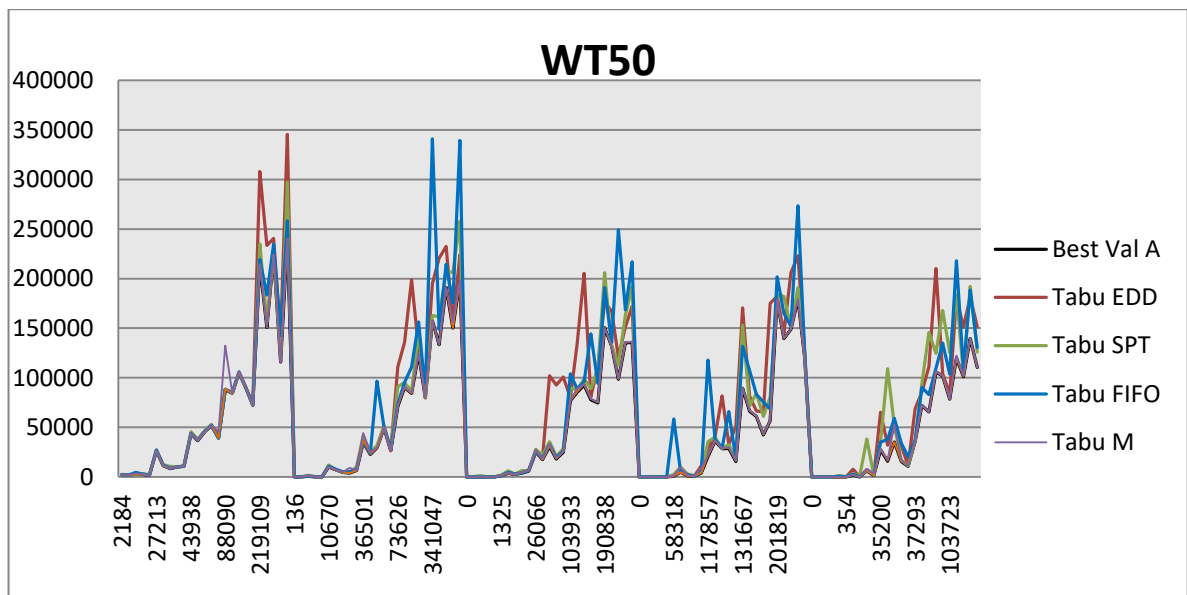


Figura 6 Total de mejores resultados obtenidos por algoritmo

### 4.1.3 CONJUNTO DE INSTANCIAS WT100

En esta sección se muestran los resultados obtenidos para las 125 instancias de 100 trabajos cada una, en este caso particular se observa que tabú con la solución inicial FIFO tiene menor desviación que las otras soluciones iniciales, Tabú M sigue mostrando mejores resultados, encontrando 25 óptimos y 84 mejores resultados respecto a los algoritmos con una sola regla.

La Tabla 17 muestra los mejores valores obtenidos para cada algoritmo respecto a los otros ejecutados, así como la cantidad de óptimos encontrados para la instancia WT50. La columna izquierda debajo de cada algoritmo contiene los mejores valores y la derecha los óptimos obtenidos.

Tabla 18 Total de mejores resultados obtenidos por algoritmo

Tabu FIFO		Tabu SPT		Tabu EDD		TabuM	
32	16	26	22	35	22	84	22

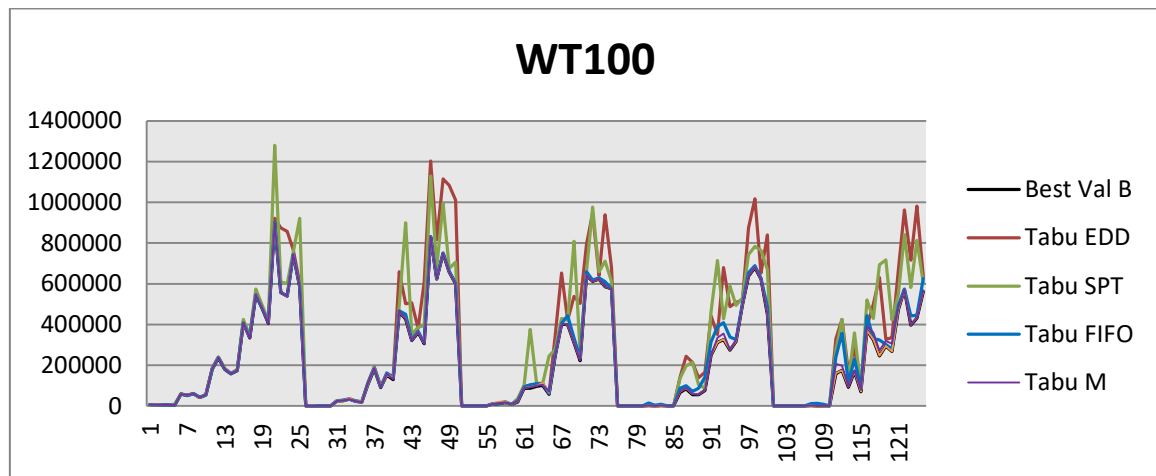


Figura 7 Resultados para el conjunto de instancias de 100 trabajos

### 4.1.4 Análisis de resultados obtenidos

De las pruebas mostradas en las Secciones 4.1.1, 4.1.2 y 4.1.3 puede observarse que en los 3 conjuntos de instancias el algoritmo Tabú M se desempeña de manera satisfactoria, tolerando bien las variables existentes en las instancias, lo

cual indica que su diseño puede ser ejecutado en un problema real de líneas de producción y producir una mejora respecto a la tardanza.

En la Tabla 19 se muestra la cantidad de óptimos obtenidos para cada conjunto, la cual se iguala con la cantidad obtenida por Tabú EDD, esto debido a que el algoritmo elige la mejor solución inicial y en ciertos casos el resultado óptimo se obtiene desde una regla de despacho.

**Tabla 19 óptimos encontrados por los algoritmo**

Conjunto de prueba	N° de instancias	Tabu FIFO	Tabu SPT	Tabu EDD	TabuM
WT40	125	17	11	23	23
WT50	125	14	11	21	21
WT100	125	16	22	22	22
Total	375	45	54	66	66

En la Tabla 20 se muestra el total de mejores resultados encontrados, estos resultados corresponden a la versión mejorada Tabú M respecto a los algoritmos probados con una sola regla de ordenamiento, prácticamente duplica el mejor de los caso de los otros algoritmos.

**Tabla 20 Número de mejores resultados encontrados**

Conjunto de prueba	N° de instancias	Tabu FIFO	Tabu SPT	Tabu EDD	TabuM
WT40	125	29	16	42	88
WT50	125	25	23	40	84
WT100	125	32	26	35	84
Total	375	131	65	117	256

En la Tabla 20 los mejores valores se refieren al mejor resultado obtenido entre los diferentes algoritmos, entre los cuales incluyen los valores óptimos y valores iguales.

Tabú M representa una mejora a Tabú con la solución inicial Tabú + EDD dado



que cómo pudo observarse, presenta menos desviaciones respecto a los valores óptimos, encontrando mejores valores y comportándose de manera más lineal consumiendo la misma cantidad de tiempo. En el Anexo A se muestran las tablas con los valores obtenidos para las WT40, WT50 y WT100 que complementan los datos de las Tablas 19 y 20.

## 4.2 APLICACIÓN DE TABÚ M A UN PROBLEMA DE PRODUCCIÓN

El problema de la tardanza total puede presentarse por muchos factores. El factor humano muchas veces se ve cómo un defecto, pero lo cierto es que en realidad es un elemento muy importante, pero con ciertas limitaciones, las cuales se pueden afrontar desde un punto de vista computacional, haciendo uso de la combinatoria, dado que la cantidad de posibilidades es generalmente de orden factorial, por lo tanto en un sentido humano, difícilmente pueden probarse una cantidad importante de posibles soluciones de manera manual, es por esta razón que el uso de un algoritmo cómo el diseñado en este trabajo toma relevancia.

En la empresa dónde se realizaron las observaciones, el problema radica en la línea de fabricación de persianas de una pulgada, aquí el problema principal se observaba en la cantidad de *scrap* o desperdicio generado, esto debido generalmente al *setup* necesario al cambiar de bobina cuando se requería un color diferente, la solución se encontró en agrupar los colores y ejecutarse en lotes del mismo color. Si bien soluciona directamente el problema de *scrap* y el problema de tiempos de *setup* de manera indirecta, en el área de empaque y envío se presentaba el problema de atraso, esto dado por la fragmentación generada a partir de la etapa de ensamble *Fast Line*, que es dónde se detectan los problemas o fallas en los pedidos y se devuelven a SANI (línea de corte de hojas) o a RAM (corte de rieles), por lo que entrar nuevamente en la secuencia es generalmente al finalizar las tareas, dado que la prioridad actual es no romper el FIFO, por lo que la reorganización de la secuencia resulta aplicable para hacer más eficiente la salida de los trabajos y evitar el atraso.

La capacidad de producción de la línea es de 2300 persianas diarias en promedio, procesando un total de 300 pedidos en promedio diariamente, con diferentes características de medida, grosor (calibre), color y número de escalerillas.

#### **4.2.1 EL PROBLEMA DE GENERACIÓN DE LA TARDANZA**

El problema del atraso se presenta en la línea de la siguiente forma: se genera una secuencia basada en el sistema FIFO a partir del área de programación, en este punto se descompone y se clasifican los pedidos por color y calibre de la bobina, de acuerdo a los filtros establecidos en el sistema que realiza el “*mix*” que es la secuencia obtenida, por lo que el encargado procede a imprimir las ordenes.

A partir del proceso anterior, esa secuencia se entrega al coordinador de la fase RAM que es el inicio de la cadena de producción, en este punto se realiza el corte de los rieles que soportan la estructura de la persiana de acuerdo a las especificaciones del pedido.

El segundo proceso conocido como SANI, es cuando llega el riel y se corta de acuerdo a las especificaciones las láminas que componen la persiana, las especificaciones corresponden al largo, color, calibre y altura de la persiana.

El tercer proceso corresponde a *Fast Line*, en este punto llegan todas las piezas para su ensamble, en este punto se verifica que todo corresponda a las especificaciones del pedido, por lo que si se detecta un defecto o un error en estas, se devuelve el pedido a la fase correspondiente para su corrección, en caso que todo esté correcto, pasa a la siguiente fase que corresponde al empaque para su envío, en dónde deben de reunir todas las persianas del pedido para su proceso, ya que es imperativo que estén todas las piezas para su embalaje y envío.

En este punto es posible observar el atraso ya que los pedidos se acumulan en el área de empaque en espera de completarse y en las fases anteriores, ya que se

quedan en espera de poder incorporarse en la secuencia.

Lo anteriormente expuesto produce que una sola tarea atrase completamente el pedido, por lo que si un pedido consta de por ejemplo 14 persianas se genera un atraso para todo ya que si las otras 13 persianas se encuentran en espera, no puede darse por terminada la orden hasta que llegue la persiana faltante.

#### **4.2.2 PROPUESTA DE SOLUCIÓN**

En esta sección se expone la estrategia que permitiría mejorar la eficiencia en la línea de producción, los siguientes puntos.

Como restricción: el agrupamiento por colores, por lo que para cada conjunto se debe ejecutar un ordenamiento, esto con la finalidad de no afectar la solución aplicada por la empresa a su problema de *scrap*.

Aplicar penalizaciones o pesos, dando prioridad al asignar un mayor peso a los trabajos con la fecha más próxima y a los que se regresan a alguna etapa anterior para su corrección, asegurando que sean los primeros en atenderse.

La fecha de entrega del pedido se sustituye por el tiempo de terminación o makespan, esto debido principalmente a que las corridas se realizan por turnos y es imperativo terminar los trabajos durante la jornada.

Por lo anteriormente descrito, la solución se basaría en ejecutar ordenamientos desde el área de programación a través del algoritmo con la finalidad de tener una secuencia con mayor holgura, además dado que cada fase de fabricación tiene un coordinador que se encarga de distribuir la secuencia, realizar corridas con la información de las ordenes proporcionada desde el área de programación o de la fase precedente donde se encuentra actualmente el trabajo, reincorporando los trabajos a corregir asignando una mayor prioridad o peso y ejecutar el algoritmo para obtener una secuencia que permita reducir el atraso.

Además dado que existen diferentes sistemas, se sugiere al encargado del área de informática, unifique la información en una sola base de datos, lo cual permite 2 cosas, extraer información completa para la obtención de secuencias, además de poder dar seguimiento a cada pedido y saber dónde se encuentra, si fue atendido y su estatus actual, dando así control a la empresa sobre su producción.

En la Tabla 21 se muestra un extracto de los trabajos a secuenciar, no existe tardanza dado que el *due date* es el mismo y se establece por la cantidad de trabajos correspondientes a la corrida:

**Tabla 21 secuencia de ejemplo**

Índice	Pt	Wj	Dd	Trabajo	f
121	14	1	2000	2353052	0
122	14	1	2000	2353093	0
123	14	1	2000	2353254	0
124	28	1	2000	2353803	0
125	42	3	2000	2353597	0

En cambio sí queda pendiente un trabajo de una corrida anterior y se agrega a la secuencia, se genera tardanza dado que es un trabajo adicional y además en la línea de empaque se atrasa el pedido hasta que el trabajo pendiente sea procesado.

**Tabla 22 trabajo reingresado a línea de producción por defecto o error de especificación**

Índice	Pt	Wj	Dd	Trabajo	f
121	14	1	2000	2353052	0
122	14	1	2000	2353093	0
123	14	1	2000	2353254	0
124	28	1	2000	2353803	0
125	42	3	2000	2353597	0
126	168	1	2000	2342417	168

En la Tabla 22 se observa claramente que existe un atraso (registro 126 con 168 unidades de atraso) por lo que, para minimizar la tardanza total, sería necesario atenderlo antes que las otras tareas.

**Tabla 23 tarea movida al inicio de la secuencia**

Índice	Pt	Wj	Dd	Trabajo	f
121	14	1	2000	2353052	0
122	14	1	2000	2353093	0
123	14	1	2000	2353254	0

124	28	1	2000	2353803	0
125	168	1	2000	2342417	0
126	42	3	2000	2353597	126

En la Tabla 23 la tarea se movió a la penúltima posición para demostrar que el intercambio de posición de las tareas permite reducir el valor de tardanza, mostrando así que la propuesta cumple el objetivo de reducir la tardanza total. Por lo que se propone utilizar el algoritmo en corridas de 1 a 2 horas para agregar las atrasadas que se generan al regresar productos defectuosos y asignarles un mayor peso, para dar prioridad a que sean atendidas. De tal manera que el algoritmo encuentre una secuencia que minimice la tardanza total ponderada y el atraso en la línea de empaque.

#### 4.2.3 Simulación de una corrida de una jornada de trabajo.

Para simular una corrida parcial de una jornada de trabajo, se generó una instancia con datos reales de 125 tareas, que corresponden a una hora de trabajo, aproximadamente. Se extrajeron los últimos 10 trabajos de manera representativa, dado que al final de la secuencia es dónde se encuentra la tardanza. Los trabajos se muestran en la Tabla 24, puede notarse que devuelve un valor óptimo, pero el cálculo del *Due Date* se realizó conforme a la empresa lo realizaba, es decir los tiempos promedios de producción, los cuales se estiman en su capacidad de producción, por lo que no se detectaba la tardanza generada. El tiempo de entrega (columna Dd) no se modifica dado que es el tiempo de finalización de las tareas correspondientes a la corrida que se está calculando, las tareas agregadas reflejan el valor de tardanza respecto al tiempo de entrega Dd, por lo que no debe modificarse para no reflejar falsos óptimos.

Tabla 24 extracto de las últimas 10 tareas de una instancia real de 125 tareas

Índice	Pt	Wj	Dd	Trabajo	f
116	14	5	4410	2337762	0
117	14	5	4410	2337833	0
118	504	3	4410	2337836	0
119	28	5	4410	2337846	0
120	210	4	4410	2337850	0
121	14	5	4410	2337866	0
122	70	5	4410	2337931	0

123	14	5	4410	2337932	0
124	14	5	4410	2337957	0
125	28	4	4410	2337966	0

Si a la misma instancia mostrada en la Tabla 24, se le agregan otros trabajos, ya sean trabajos pendientes del turno anterior o trabajos que se regresan a etapas previas para su corrección, en este punto se comienza a generar la tardanza. Para observarlo se agregaron al final de la secuencia 3 tareas de manera aleatoria, pueden observarse en la Tabla 25 así como la tardanza correspondiente a los trabajos insertados en la secuencia (registros 126, 127 y 128).

Tabla 25 Últimas 10 tareas de la secuencia + tareas agregadas

Índice	Pt	Wj	Dd	Trabajo	f
116	14	5	4410	2337762	0
117	14	5	4410	2337833	0
118	504	3	4410	2337836	0
119	28	5	4410	2337846	0
120	210	4	4410	2337850	0
121	14	5	4410	2337866	0
122	70	5	4410	2337931	0
123	14	5	4410	2337932	0
124	14	5	4410	2337957	0
125	28	4	4410	2337966	0
126	14	5	4410	2325534	70
127	56	5	4410	2333790	350
128	42	5	4410	2333743	630

Al aplicar el algoritmo de búsqueda Tabú M, se obtiene un valor de tardanza de **336** unidades, lo que representa una reducción de más de un 70 por ciento del valor inicial obtenido al agregar las tareas, que es **1,498**, dejando solamente una tarea con atraso. Lo anterior se puede observar en la Tabla 26 que contiene las últimas 13 tareas del conjunto de 125 dónde se agregaron las 3 tareas pendientes adicionales, dando un total de 128 tareas.

Tabla 26 solución obtenida por el algoritmo Tabú M

Índice	Pt	Wj	Dd	Trabajo	f
116	14	4	4410	2337642	0
117	14	5	4410	2337644	0
118	42	5	4410	2337740	0
119	14	5	4410	2337762	0
120	14	5	4410	2337833	0
121	434	5	4410	2336287	0

122	28	5	4410	2337846	0
123	28	4	4410	2337966	0
124	14	5	4410	2337866	0
125	70	5	4410	2337931	0
126	42	5	4410	2333743	0
127	252	5	4410	2334237	0
128	504	3	4410	2337836	336

En la Tabla 26 se obtiene una mejora satisfactoria al reducir a una sola tarea con atraso. La Tabla corresponde a una hora de producción aproximadamente.

Aunque el escenario es variable en la cantidad de tareas por la cantidad de persianas que requiere cada corrida, así como su prioridad y cantidad de trabajos atrasados.

#### 4.2.3 SIMULACIÓN DE UNA JORNADA DE TRABAJO COMPLETA

Para comprobar el funcionamiento se diseñó un experimento con 10 corridas simulando una carga de trabajo diaria, cada corrida consta de 100 trabajos, a partir de la segunda corrida se agregan cantidades variables de trabajos de la corrida anterior, los cuales se extrajeron de manera aleatoria de la corrida anterior, dichas tareas simulan tareas pendientes o con defecto que se devuelven para subsanar errores y se enlistan en la corrida posterior. Para obtener una secuencia se utilizó el algoritmo Tabú M con la finalidad de medir su eficiencia respecto al ordenamiento *FIFO*.

De la misma manera que el ejemplo anterior, se muestran los últimos 10 elementos de la secuencia original de 100 más los que se agregan a la secuencia y el valor de tardanza obtenido en cada secuencia resultante, además se han agregado 2 columnas al final, dónde se muestra la secuencia obtenida por la secuencia *FIFO* (columna Trabajo *FIFO*) y el valor de tardanza obtenido (*f FIFO*).

Tabla 27 Corrida 1 de 100 trabajos

Índice	Pt	Wj	Dd	Trabajo	<i>f (Tabu M)</i>	Trabajo ( <i>FIFO</i> )	<i>f(FIFO)</i>
91	28	5	11606	2313943	0	2313943	0
92	14	5	11606	2313956	0	2313956	0

93	14	6	11606	2313988	0	2313988	0
94	14	6	11606	2314129	0	2314129	0
95	14	6	11606	2314210	0	2314210	0
96	1218	6	11606	2314266	0	2314266	0
97	28	6	11606	2314549	0	2314549	0
98	28	6	11606	2314552	0	2314552	0
99	14	6	11606	2314772	0	2314772	0
100	28	6	11606	2315327	0	2315327	0

En la Tabla 27 se muestra la primera corrida, de esa secuencia se extrajeron 5 tareas de manera aleatoria y se agregaron a la segunda corrida, en la Tabla 28 se muestra la secuencia obtenida por el algoritmo Tabu M y la tardanza obtenida para la segunda corrida.

Tabla 28 segunda corrida de 100 tareas + 5 tareas para solventar de la secuencia anterior

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
91	28	6	4074	2322299	0	2324508	0
92	28	6	4074	2317514	0	2324545	0
93	560	4	4074	2320377	0	2324577	0
94	28	5	4074	2325389	0	2324586	0
95	28	6	4074	2317477	0	2324775	0
96	28	6	4074	2321142	0	2324929	84
97	28	4	4074	2279007	0	2324984	140
98	28	6	4074	2308024	0	2325013	210
99	28	5	4074	2317149	0	2325029	504
100	14	5	4074	2316049	0	2325088	490
101	42	5	4074	2318162	0	2325132	560
102	42	5	4074	2320379	0	2325252	756
103	56	5	4074	2317145	0	2325327	840
104	56	5	4074	2324250	0	2325329	1008
105	504	4	4074	2320000	784	2325389	980

En la segunda corrida, el atraso obtenido al calcular la tardanza con el ordenamiento *FIFO* (usado por la empresa) es de **5,572**. El valor resultante de la secuencia obtenida por Tabú M es de **784**. Es observable la reducción de la tardanza mediante el uso del algoritmo Tabú M. De esta segunda corrida se extrajeron 7 tareas de manera aleatoria para insertarse en la secuencia de la Tabla 29 de 100 tareas que corresponde a la tercera corrida.

Tabla 29 tercera corrida de 100 tareas + 7 tareas

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
--------	----	----	----	---------	-------------	----------------	-----------



					<i>M)</i>	<i>(FIFO)</i>	
91	756	3	5250	2330604	0	2330604	0
92	14	5	5250	2330658	0	2330658	0
93	14	5	5250	2330762	0	2330762	0
94	28	5	5250	2330821	0	2330821	0
95	14	5	5250	2330938	0	2330938	0
96	14	5	5250	2330973	0	2330973	0
97	14	6	5250	2330976	0	2330976	0
98	14	5	5250	2331020	0	2331020	0
99	14	5	5250	2331040	0	2331040	0
100	14	5	5250	2331064	0	2331064	0
101	14	5	5250	2331113	0	2331113	0
102	14	5	5250	2331191	0	2331191	0
103	28	5	5250	2331303	0	2331303	0
104	14	5	5250	2331377	0	2331377	0
105	686	3	5250	2331400	0	2331400	378
106	98	5	5250	2327420	0	2331439	700
107	1596	2	5250	2326802	308	2331504	770

En el caso de la Tabla 29 el atraso del ordenamiento *FIFO* es de **1,848**, el cual se redujo a **308** aplicando el algoritmo Tabú M. De la tercera corrida se extrajeron 7 tareas de manera aleatoria, las cuales se agregaron a la cuarta corrida correspondiente a la Tabla 30.

Tabla 30 cuarta corrida de 100 tareas + 7 tareas

Índice	Pt	Wj	Dd	Trabajo	<i>f (Tabu M)</i>	<i>Trabajo (FIFO)</i>	<i>f(FIFO)</i>
91	42	5	2716	2331608	0	2335063	6650
92	42	5	2716	2334907	0	2335069	6930
93	42	5	2716	2333743	0	2335116	7000
94	42	5	2716	2331881	0	2335141	7070
95	56	5	2716	2334382	0	2335188	7140
96	56	5	2716	2333792	0	2335206	7210
97	56	5	2716	2335473	0	2335216	7280
98	56	5	2716	2333181	0	2335226	7350
99	56	5	2716	2335069	0	2335242	7490
100	70	5	2716	2333503	0	2335265	7560
101	70	3	2716	2332520	0	2335273	7630
102	70	5	2716	2331920	0	2335375	7770
103	84	5	2716	2329716	0	2335405	6552
104	84	4	2716	2335405	0	2335435	8400
105	112	5	2716	2334474	0	2335450	8470
106	252	5	2716	2334237	840	2335457	8540
107	1596	2	2716	2326802	4368	2335473	8820

En la Tabla 30 el valor de tardanza aplicando *FIFO* es de **279,860**, el cual después de aplicar el algoritmo Tabú M, se redujo a **4,638**. En la quinta corrida se agregaron 9 tareas de manera aleatoria extraídos de la corrida 4.

Tabla 31 quinta corrida de 100 tareas + 9 tareas

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
91	28	5	3808	2331870	0	2337977	0
92	28	5	3808	2335867	0	2337982	0
93	28	5	3808	2335889	0	2338024	0
94	28	5	3808	2336070	0	2338153	0
95	28	5	3808	2336285	0	2338394	0
96	28	5	3808	2335960	0	2338412	0
97	56	5	3808	2336066	0	2338452	56
98	28	5	3808	2335715	0	2338486	210
99	28	5	3808	2335714	0	2338527	280
100	28	5	3808	2335712	0	2338530	280
101	28	4	3808	2336240	0	2338546	420
102	28	5	3808	2336149	0	2338555	560
103	28	5	3808	2334952	0	2338576	630
104	42	5	3808	2332623	0	2338597	560
105	42	5	3808	2336870	0	2338615	770
106	28	4	3808	2337966	0	2338619	840
107	56	5	3808	2338394	0	2338632	910
108	168	5	3808	2335638	0	2338669	1050
109	504	3	3808	2337836	672	2338674	1120

En la quinta corrida el valor de tardanza aplicando *FIFO* es de **7,686**, después de aplicar el algoritmo Tabú M, el valor se redujo a **672**. De esta corrida se extrajeron 9 tareas los cuales se agregaron a la sexta corrida que corresponde a la Tabla 32.

Tabla 32 sexta corrida de 100 tareas + 8 tareas agregadas

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
91	14	5	4494	2341575	0	2341575	0
92	28	5	4494	2341588	0	2341588	0
93	14	5	4494	2341613	0	2341613	0
94	14	5	4494	2341620	0	2341620	0
95	14	5	4494	2341640	0	2341640	0
96	84	5	4494	2341644	0	2341644	0
97	56	5	4494	2341649	0	2341649	0
98	14	5	4494	2341651	0	2341651	0
99	14	5	4494	2341655	0	2341655	0
100	14	5	4494	2341670	0	2341670	0
101	28	5	4494	2341688	0	2341688	0

102	42	5	4494	2341718	0	2341718	0
103	56	5	4494	2341746	0	2341746	0
104	42	5	4494	2341751	0	2341751	140
105	28	5	4494	2341756	0	2341756	280
106	56	5	4494	2341760	0	2341760	560
107	28	5	4494	2335710	0	2341762	630
108	28	5	4494	2336444	0	2341779	700
109	182	4	4494	2338817	672	2341785	840

En la sexta corrida el valor de tardanza *FIFO* es de **3,150**, después de aplicar el algoritmo Tabú M, el valor se redujo a **672**. De esta corrida se extrajeron 8 tareas que se agregaron a la séptima corrida que corresponde a la Tabla 33.

Tabla 33 séptima corrida de 100 tareas +8 agregados

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
91	42	5	2898	2344107	0	2344107	0
92	14	5	2898	2344204	0	2344204	0
93	28	5	2898	2344213	0	2344213	0
94	98	5	2898	2343046	0	2344223	0
95	14	5	2898	2344260	0	2344260	0
96	28	5	2898	2344270	0	2344270	0
97	14	5	2898	2344289	0	2344289	0
98	14	3	2898	2344307	0	2344307	0
99	84	5	2898	2344352	0	2344352	0
100	84	4	2898	2344367	0	2344367	224
101	28	5	2898	2338904	0	2344370	350
102	14	4	2898	2344377	0	2344377	336
103	14	5	2898	2344384	0	2344384	490
104	14	5	2898	2344389	0	2344389	560
105	42	5	2898	2344425	0	2344425	770
106	84	5	2898	2342332	0	2344434	840
107	84	4	2898	2343001	168	2344458	910
108	168	3	2898	2342889	798	2344466	1050

En la séptima corrida el valor de tardanza *FIFO* es de **5,530**, después de aplicar el algoritmo Tabú M, el valor se redujo a **798**. De esta corrida se extrajeron 8 tareas que se agregaron a la octava corrida que corresponde a la Tabla 34.

Tabla 34 octava corrida de 100 tareas + 8 tareas agregados

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
91	14	3	2296	2346294	0	2346563	0
92	14	5	2296	2346492	0	2346610	0

93	14	4	2296	2346500	0	2346647	0
94	14	5	2296	2346610	0	2346651	0
95	14	5	2296	2346647	0	2346744	0
96	56	5	2296	2345182	0	2346821	56
97	14	5	2296	2346303	0	2346836	112
98	14	4	2296	2346885	0	2346848	280
99	14	4	2296	2346821	0	2346859	280
100	14	4	2296	2346836	0	2346866	490
101	14	4	2296	2346271	0	2346870	560
102	14	4	2296	2346859	0	2346885	504
103	14	5	2296	2346261	0	2346902	700
104	14	4	2296	2344910	0	2346924	770
105	42	5	2296	2346221	0	2346945	840
106	84	4	2296	2344367	336	2346959	980
107	98	5	2296	2344924	1246	2346960	1120
108	84	4	2296	2346302	2310	2346984	1330

En la octava corrida el valor de tardanza *FIFO* es de **8,022**, después de aplicar el algoritmo Tabú M, el valor se redujo a **2,310**. De esta corrida se extrajeron 10 tareas que se agregaron a la novena corrida que corresponde a la Tabla 35.

Tabla 35 novena corrida de 100 tareas + 10 tareas agregados

Índice	Pt	Wj	Dd	Trabajo	$f(Tabu M)$	Trabajo (FIFO)	$f(FIFO)$
91	14	3	2198	2350800	0	2351298	0
92	14	4	2198	2351378	0	2351302	0
93	14	4	2198	2351372	0	2351309	0
94	14	4	2198	2350847	0	2351310	0
95	14	4	2198	2351353	0	2351322	0
96	14	4	2198	2351309	0	2351326	0
97	14	3	2198	2351310	0	2351334	0
98	70	4	2198	2351187	0	2351352	0
99	14	5	2198	2351352	0	2351353	0
100	14	3	2198	2350874	0	2351370	224
101	14	3	2198	2351334	0	2351372	280
102	14	3	2198	2351210	0	2351378	336
103	28	4	2198	2350981	0	2351393	448
104	28	4	2198	2349294	0	2351398	616
105	42	4	2198	2351398	0	2351400	672
106	28	3	2198	2350620	0	2351411	784
107	28	3	2198	2350988	84	2351422	630
108	56	3	2198	2350804	336	2351427	896
109	70	4	2198	2349008	952	2351436	952
110	98	4	2198	2347164	1960	2351441	756

En la novena corrida el valor de tardanza *FIFO* es de **6,594**, después de aplicar el algoritmo Tabú M, el valor se redujo a **1,960**. De esta corrida se extrajeron 9 tareas que se agregaron a la novena corrida que corresponde a la Tabla 36.

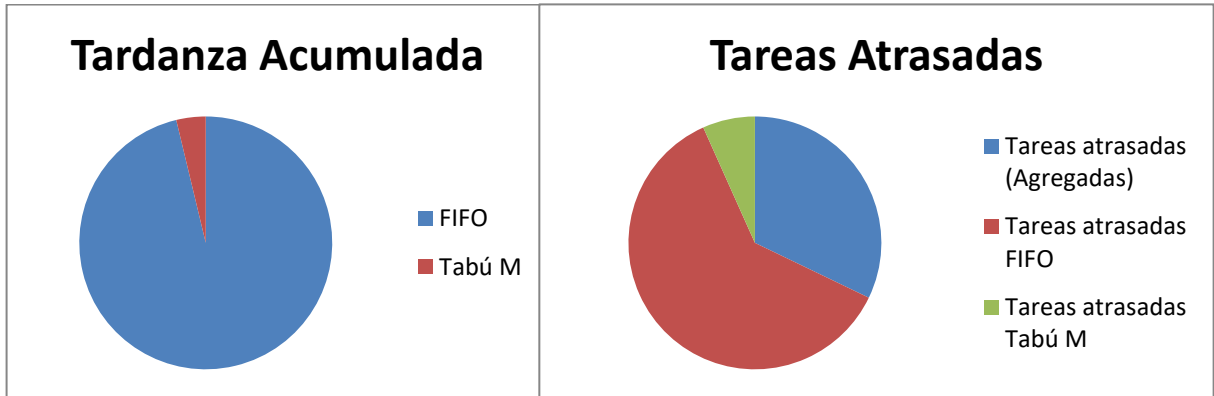
Tabla 36 décima corrida de 100 tareas + 9 tareas agregados

Índice	Pt	Wj	Dd	Trabajo	$f(\text{Tabu M})$	Trabajo (FIFO)	$f(\text{FIFO})$
91	14	4	2142	2353355	0	2353355	0
92	14	4	2142	2353387	0	2353387	0
93	14	3	2142	2353388	0	2353388	0
94	28	4	2142	2353392	0	2353392	0
95	14	4	2142	2353406	0	2353406	0
96	14	4	2142	2353420	0	2353420	0
97	14	4	2142	2353431	0	2353431	0
98	28	4	2142	2353433	0	2353433	0
99	14	4	2142	2353453	0	2353453	0
100	84	4	2142	2353462	0	2353462	224
101	14	3	2142	2348997	0	2353466	280
102	14	4	2142	2353477	0	2353477	336
103	14	4	2142	2353479	0	2353479	392
104	70	4	2142	2349008	0	2353483	504
105	14	4	2142	2353492	0	2353492	560
106	14	4	2142	2353520	0	2353520	616
107	28	4	2142	2351411	0	2353522	672
108	70	4	2142	2351498	168	2353532	728
109	154	4	2142	2351504	952	2353535	784

En la décima corrida el valor de tardanza *FIFO* es de **5,096**, después de aplicar el algoritmo Tabú M, el valor se redujo a **952**. En este punto terminan las ejecuciones de la simulación obteniendo los siguientes resultados:

Tabla 37 Resultados de la simulación de 1000 tareas y 10 corridas

Tardanza acumulada <i>FIFO</i>	Tardanza acumulada Tabú M	Tareas atrasadas (agregadas)	Tareas atrasadas <i>FIFO</i>	Tareas atrasadas Tabú M
<b>328888</b>	<b>12824</b>	<b>72</b>	<b>137</b>	<b>15</b>



Los resultados de la Tabla 37 representan una reducción del valor original a casi un 4% ( $12824 \cdot 100 / 328888 = 3.899$ ) del valor obtenido por *FIFO*, cuando es usado el algoritmo Tabú M. Las tareas agregadas en un principio se agregan al final de la secuencia, después de aplicar *FIFO* nuevamente se modifica su orden, moviéndose al principio de la secuencia, lo cual modifica el valor de tardanza, la cual se expresa como Tardanza Acumulada dado que es la suma de las tardanzas de los 10 conjuntos de prueba.

En relación a las tareas atrasadas se redujo en más de 3 veces el número de tareas respecto a las tareas agregadas, después de aplicar *FIFO* se modifica el valor a 148 tareas atrasadas por lo que se reduce a un 10% el número de tareas atrasadas mediante Tabú M ( $15 \cdot 100 / 137 = 10.94$ ). Aunque cabe destacar que en 4 de 10 corridas el algoritmo Tabú M obtuvo solamente fue una tarea atrasada y en 8 de los 9 casos que es dónde se insertaron tareas, menos de 5 las tareas mostraron atraso.

En la Tabla 38 se muestran los valores promedio de las corridas y sus valores obtenidos, resumiendo así los resultados de cada corrida del experimento ejecutado. El tiempo de ejecución corresponde al del algoritmo Tabú M, el cual utiliza como valor inicial la solución obtenida por el ordenamiento *FIFO*.

Tabla 38 Resumen de Resultados

Corrida	Tareas agregadas	Tareas atrasadas	Tareas atrasadas	Tardanza <i>FIFO</i>	Tardanza Tabú M	Tiempo de ejecución
---------	------------------	------------------	------------------	----------------------	-----------------	---------------------

		<b>FIFO</b>	<b>Tabú M</b>			<b>(Seg)</b>
<b>1</b>	0	0	0	0	0	2.887
<b>2</b>	5	10	1	5572	784	3.382
<b>3</b>	7	3	1	1848	308	3.541
<b>4</b>	7	59	2	279860	4368	3.49
<b>5</b>	9	13	1	7686	672	3.74
<b>6</b>	9	9	1	3150	672	3.702
<b>7</b>	8	9	2	5530	798	3.592
<b>8</b>	8	13	3	8022	2310	3.667
<b>9</b>	10	11	4	6594	1960	3.791
<b>10</b>	9	10	2	5096	952	3.7
<b>PROMEDIO</b>	<b>7.2</b>	<b>13.7</b>	<b>1.7</b>	<b>32335.8</b>	<b>1282.4</b>	<b>3.5492</b>

La diferencia obtenida entre las diferentes versiones de Tabú respecto a Tabú M se muestran en la Tabla 39, dónde se observa la linealidad del algoritmo desarrollado, lo que puede garantizar que en la mayoría de los casos no se producirán mayores desviaciones.

**Tabla 39 Tardanza obtenida por los diferentes algoritmos en la instancia real**

<b>Instancia</b>	<b>FIFO</b>	<b>Tabú +EDD</b>	<b>Tabú +SPT</b>	<b>Tabú +FIFO</b>	<b>Tabú M</b>
<b>1</b>	0	0	0	0	<b>0</b>
<b>2</b>	5572	784	784	784	<b>784</b>
<b>3</b>	1848	1820	308	308	<b>308</b>
<b>4</b>	279860	52332	4368	25802	<b>4368</b>
<b>5</b>	7686	1456	672	966	<b>672</b>
<b>6</b>	3150	3052	672	672	<b>672</b>
<b>7</b>	5530	1134	798	798	<b>798</b>
<b>8</b>	8022	2254	2240	2338	<b>2310</b>
<b>9</b>	6594	2044	2002	2002	<b>1960</b>
<b>10</b>	5096	952	952	952	<b>952</b>

Puede observarse que en el caso de la instancia 8 no produce el menor de los resultados, el cual es obtenido por Tabú +EDD, con una diferencia de 56 unidades, en cambio en la instancia 9 el algoritmo reporta una mejora respecto a los otros algoritmos de 42 unidades en el caso de Tabú +SPT y Tabú +FIFO.

# Capítulo 5

## CONCLUSIONES

---

En esta sección se muestran los resultados de la aplicación de Búsqueda Tabú al problema de programación de tareas con datos sintéticos y datos extraídos de una empresa de fabricación de persianas, en la sección 5.1 se muestran las conclusiones obtenidas en este trabajo, la sección 5.2 muestra los trabajos futuros y finalmente la sección 5.3 muestra las publicaciones desarrolladas durante el periodo de desarrollo de esta tesis.

### 5.1 CONCLUSIONES

En base a lo observado a lo largo del trabajo de tesis, se puede concluir lo siguiente:

- En el caso de los datos sintéticos se observa cómo las reglas de despacho no son capaces de resolver todos los casos eficientemente.
- La metaheurística Búsqueda Tabú puede obtener mejores resultados por si sola, pero, utilizando soluciones iniciales cómo las reglas de despacho, en el caso de la versión Tabú M, obtiene más del 20% de óptimos del total de instancias probadas y una desviación aceptable de los óptimos y mejores valores reportados.



- Es posible obtener buenos resultados y mejorar la eficiencia de producción al reducir la tardanza, utilizando métodos metaheurísticos en las empresas de manufactura, como puede constatarse en el experimento mostrado con instancias reales, dónde se redujo en alrededor de 4 veces el valor de tardanza respecto a la regla *FIFO*.
- Los ambientes de producción son muy variables, por lo que una heurística sencilla simplifica el tratamiento de los datos necesarios para buscar una solución, tal es el caso de este estudio, dónde se modificaron parámetros sencillos para poder tratar con un problema de producción complejo.
- El factor humano es el punto de partida para determinar el problema y la solución; es un factor importante dado que el personal conoce las áreas de trabajo y su protocolo de actuación, el apoyo en una herramienta puede hacer más eficiente su desempeño.
- El uso de metaheurísticas es necesario en las empresas: el volumen de pedidos que se maneja es considerable, por lo que encontrar una secuencia de manera manual no siempre es factible, el uso de reglas de despacho como se mostró en este trabajo, tiene resultados variables y no pueden garantizar una aproximación u optimalidad en todos los casos.
- Las mejoras pueden obtenerse prácticamente de cualquier recurso aplicado, como en el caso de este trabajo, al aplicar reglas de despacho, se observó cómo se comportaban los resultados iniciales y cómo modificaban el desempeño de la búsqueda tabú.

En el caso real, la parte más difícil consistió en encontrar los cuellos de botella y el tratamiento de datos para generar una propuesta, esto debido a que muchas veces la información se encuentra fragmentada, además las estimaciones de los tiempos de proceso y la planeación, puede aparentar que el trabajo se realiza de manera óptima, por lo que la tardanza puede no apreciarse, por otra parte la tardanza en realidad es un modelado que permite medir la eficiencia de la producción, por lo que la planeación de tareas con mediciones realistas, es una

parte fundamental de la cadena de valor y aún más en la parte que corresponde a la producción.

## **5.2 TRABAJOS FUTUROS**

El mundo de las metaheurísticas y la optimización combinatoria, son ampliamente susceptibles a mejoras, además que estos trabajos como el realizado, permiten conocer el problema e inspirar a nuevas ideas, por lo que en el futuro, se pretende realizar lo siguiente:

- Mejoras al algoritmo y más casos de estudio de los ambientes de producción para obtener ideas más generales que permitan entender estas áreas de oportunidad.
- Desarrollo de otras metaheurísticas, sean de tipo exploratoria o explotatoria.

## **5.3 PUBLICACIONES DESARROLLADAS**

Se publicaron 2 trabajos de divulgación y 2 ponencias durante el lapso de la maestría los cuales son:

- “Modelado conceptual de una aplicación Web usando la metodología OOWS: un caso práctico”. En: Tecointelecto 15(2): 19- 28, Cd. Victoria, México, Diciembre 2018.
- “Uso de la metaheurística búsqueda tabú para resolver el problema de la tardanza total”, publicación de artículo y videoconferencia, Academia Journals Morelia, Mayo de 2020.
- “Programación de tareas en una línea de producción aplicando Búsqueda Tabú”, conferencia en el marco del 44 aniversario del Instituto Tecnológico de Ciudad Victoria, Octubre 2019.

## Anexos

---

### Anexo A: Resultados de los conjuntos de instancias WT40, WT50 y WT100

En esta sección se muestran las Tablas que contienen los resultados completos de las instancias sintéticas usadas para la experimentación con el algoritmo de búsqueda Tabú M, así como los otros algoritmos tabú utilizados para la experimentación con los conjuntos de instancias.

Las Tablas 28, 29 y 30 muestran para los conjuntos WT40, WT50 y WT100 respectivamente. En la primera columna el valor óptimo reportado por los autores de las instancias, de la segunda a la cuarta columna corresponde a los valores obtenidos por reglas de despacho, de la quinta a la octava columna se muestran los resultados obtenidos por los diferentes algoritmos tabú implementados, además como dato complementario de la novena a la doceava columna muestran las diferencias entre el valor óptimo y el obtenido por el algoritmo, en amarillo se resaltan los mejores valores obtenidos por cada algoritmo tabú y en verde se resaltan los mejores valores obtenidos del algoritmo mejorado Tabú M respecto a los otros algoritmos probados.

Tabla 40 Resultados completos de la instancia WT40

OPT	FIFO	SPT	EDD	Tabu FIFO	Tabu SPT	Tabu EDD	Tabú M	≠ Opt vs T FIFO	≠ Opt vs T SPT	≠ Opt vs T EDD	≠ Opt vs T M
913	16672	7755	1588	956	1024	1003	956	43	111	90	43
1225	11036	5993	5226	1495	1894	1239	1478	270	669	14	253
537	13174	4486	3051	597	1085	573	760	60	548	36	223
2094	10444	6849	5527	2391	2127	2181	2238	297	33	87	144
990	12006	4650	4030	990	992	990	1102	0	2	0	0
6955	47616	26491	23691	7232	8289	7068	7522	277	1334	113	567
6324	46420	20099	33547	7217	6665	6447	6608	893	341	123	284
6865	39586	17549	23032	6882	7661	7707	6976	17	796	842	111
16225	44572	32629	34596	16250	17012	16499	16659	25	787	274	434
9737	58317	30049	44178	10017	10248	9990	9958	280	511	253	221

17465	79327	34386	67845	18321	18089	17492	18203	856	624	27	738
19312	66049	45255	50511	19512	19821	19838	19334	200	509	526	22
29256	105394	41628	75119	29464	29824	30223	29634	208	568	967	378
14377	71689	27579	37395	14746	14534	14746	14544	369	157	369	167
26914	94544	43627	56964	27320	27943	27241	27562	406	1029	327	648
72317	159881	98819	184435	72825	73414	72587	72784	508	1097	270	467
78623	151472	104992	134094	78635	78964	79348	79003	12	341	725	380
74310	134666	100007	133571	106912	74477	74675	74616	32602	167	365	306
77122	158682	133968	167058	108046	77753	77911	77607	30924	631	789	485
63229	147278	88602	163608	63968	64143	100628	63494	739	914	37399	265
77774	142996	96171	158684	86255	81796	82343	77821	8481	4022	4569	47
100484	199480	145186	198900	201736	101249	103204	100683	101252	765	2720	199
135618	229561	181155	192330	146862	185386	136360	135653	11244	49768	742	35
119947	239418	137020	216677	125442	122885	122401	120028	5495	2938	2454	81
128747	241293	160908	252546	241041	170284	160479	128756	112294	41537	31732	9
108	12039	9872	144	144	223	144	144	36	115	36	36
64	5202	6078	64	70	140	64	681	6	76	0	0
15	7116	4152	15	48	58	15	15	33	43	0	0
47	3311	7687	47	396	481	47	396	349	434	0	0
98	7620	5487	832	392	392	105	461	294	294	7	363
6575	53825	34851	19866	7443	7647	6591	7587	868	1072	16	1012
4098	46008	21022	14536	5094	4653	4455	4263	996	555	357	165
5468	52517	27017	25293	5713	8122	5703	5820	245	2654	235	352
2648	32682	11555	16424	2955	3159	2872	2673	307	511	224	25
5290	54001	21325	16788	5983	6643	5318	5595	693	1353	28	305
19732	86846	52151	73316	21152	21802	20197	20111	1420	2070	465	379
17349	69017	40363	56141	17513	18841	17509	18375	164	1492	160	1026
24499	80112	51625	78933	47459	26371	25503	24600	22960	1872	1004	101
19008	116259	52179	41577	19531	21837	20243	19894	523	2829	1235	886
19611	56897	41398	43557	37234	21157	19842	19993	17623	1546	231	382
57640	103520	88889	153250	85523	58394	57957	57661	27883	754	317	21
81462	155999	124730	142135	86181	125935	130304	81812	4719	44473	48842	350
65134	162754	112145	129718	146143	71896	81060	65170	81009	6762	15926	36
78139	194017	107300	150485	78363	78462	78320	78293	224	323	181	154
66579	139335	113160	124213	69623	67154	126297	66587	3044	575	59718	8
64451	143678	79479	105109	143678	67277	80987	64624	79227	2826	16536	173
113999	197835	140973	223314	127595	122842	124294	114091	13596	8843	10295	92
74323	164185	92878	190251	129218	85099	140809	74432	54895	10776	66486	109
110295	187320	132851	164106	111080	120428	119704	110379	785	10133	9409	84
95616	171560	132234	163750	123140	105777	97467	95653	27524	10161	1851	37
0	30987	26039	0	0	588	0	0	0	588	0	0

0	15511	8137	0	0	55	0	0	0	55	0	0
0	9941	13691	0	0	0	0	0	0	0	0	0
0	9594	7503	0	0	0	0	0	0	0	0	0
0	22615	10463	0	0	24	0	24	0	24	0	24
2099	36520	22551	9189	2555	2548	2359	2492	456	449	260	393
2260	53664	18311	4783	2786	3183	2485	2780	526	923	225	520
4936	60306	30798	14008	5345	6264	19526	5617	409	1328	14590	681
3784	71739	38703	19933	3980	4593	3934	4847	196	809	150	1063
3289	41538	28777	9008	3996	6279	3967	3394	707	2990	678	105
20281	80865	59045	47616	21233	22308	50743	20932	952	2027	30462	651
13403	73185	44441	39124	13703	14654	45499	13879	300	1251	32096	476
19771	70852	57441	58438	37576	22592	20103	20285	17805	2821	332	514
24346	85176	80641	85363	25228	26170	92572	25338	882	1824	68226	992
14905	78121	46375	49776	16059	15555	55392	16289	1154	650	40487	1384
65386	128936	106539	135664	86567	107274	111327	67359	21181	41888	45941	1973
65756	161791	98691	122880	97121	67886	68531	65985	31365	2130	2775	229
78451	189625	128496	160171	189091	97396	80691	78504	110640	18945	2240	53
81627	171833	124326	170858	101011	81926	87709	82674	19384	299	6082	1047
68242	128250	110661	141555	91825	71367	79130	68443	23583	3125	10888	201
90486	175627	115834	141972	95322	119510	127883	90841	4836	29024	37397	355
115249	236054	155545	246607	235994	142076	117616	115430	120745	26827	2367	181
68529	163003	99687	130115	81933	75227	80823	68586	13404	6698	12294	57
79006	169002	97397	186084	87155	91427	108484	79096	8149	12421	29478	90
98110	186661	147113	161402	186566	148336	104237	98177	88456	50226	6127	67
0	10914	6682	0	0	0	0	0	0	0	0	0
0	12974	10940	0	0	0	0	0	0	0	0	0
0	14437	4045	0	0	0	0	0	0	0	0	0
0	23679	6765	0	0	0	0	0	0	0	0	0
0	25784	20497	0	0	0	0	0	0	0	0	0
684	29222	20318	1375	1213	1945	720	979	529	1261	36	295
172	44199	20677	172	216	549	172	216	44	377	0	44
798	58209	47754	1200	1266	2754	944	1876	468	1956	146	1078
617	61108	44212	1004	29333	3728	709	1540	28716	3111	92	923
776	18418	26262	4754	977	1693	1028	1048	201	917	252	272
10262	89622	61707	31609	11954	12651	10466	10523	1692	2389	204	261
18646	113468	67692	53206	113552	36677	19983	19219	94906	18031	1337	573
10021	86057	43404	34755	14852	23150	21208	11295	4831	13129	11187	1274
25881	111998	88179	63828	35087	47669	26518	26409	9206	21788	637	528
8159	68531	54270	56036	24158	9902	8452	8323	15999	1743	293	164
47683	132899	83736	128828	63951	50037	56865	48878	16268	2354	9182	1195
43004	118033	79711	112119	56914	80599	54888	43284	13910	37595	11884	280

55730	148235	89744	111418	61521	90410	74571	56207	5791	34680	18841	477
59494	151270	95815	143599	151270	63576	86899	60205	91776	4082	27405	711
42688	134072	81670	88920	134072	52600	80899	42931	91384	9912	38211	243
126048	210514	162102	205234	135958	131023	144060	126182	9910	4975	18012	134
114686	198301	148874	138686	122221	150838	156511	114777	7535	36152	41825	91
112102	213580	143170	200558	213487	115986	152630	112177	101385	3884	40528	75
98206	205317	141026	177362	99196	102234	110041	98325	990	4028	11835	119
157296	272218	194943	239252	176896	163964	161843	157390	19600	6668	4547	94
0	25439	21548	0	0	0	0	0	0	0	0	0
0	25933	15631	0	0	0	0	0	0	0	0	0
0	34620	12940	0	0	135	0	0	0	135	0	0
0	39400	29406	0	23088	0	0	0	23088	0	0	0
0	25532	11055	0	0	0	0	0	0	0	0	0
0	73304	32043	0	0	156	0	0	0	156	0	0
516	91557	61465	1236	978	1222	2228	1174	462	706	1712	658
3354	56237	48674	6105	4864	5800	7324	4989	1510	2446	3970	1635
0	47536	25991	0	0	21071	0	0	0	21071	0	0
0	27314	24475	0	8469	25	0	0	8469	25	0	0
31478	126371	85816	69834	39017	79042	36365	31727	7539	47564	4887	249
21169	75244	70799	42715	29569	43438	22814	21349	8400	22269	1645	180
27077	106439	78696	55459	30156	31900	30697	27388	3079	4823	3620	311
19648	103129	74203	67148	23120	24388	71513	20681	3472	4740	51865	1033
13774	107453	70926	31206	107338	15098	20876	14538	93564	1324	7102	764
46770	149782	87072	99424	48828	88995	48595	46918	2058	42225	1825	148
50364	123301	103189	111761	109348	51524	55175	50539	58984	1160	4811	175
25460	92682	52234	75672	26687	56627	30377	25567	1227	31167	4917	107
66707	182743	102958	141489	82351	107228	67469	66779	15644	40521	762	72
69019	165416	109629	116647	76099	70756	69239	69217	7080	1737	220	198
122266	241391	163719	199266	172435	169673	130699	122441	50169	47407	8433	175
82456	199402	116088	160220	91714	116894	141550	82572	9258	34438	59094	116
75118	180905	109462	147134	91649	108201	76517	75380	16531	33083	1399	262
73041	134942	108177	144122	73090	108854	106734	73589	49	35813	33693	548
104531	191852	132217	199432	110388	105564	105971	104661	5857	1033	1440	130

Tabla 41 Resultados completos de la instancia WT50

OPT	FIFO	SPT	EDD	Tabu FIFO	Tabu SPT	Tabu EDD	Tabu M	≠ Opt vs T FIFO	≠ Opt vs T SPT	≠ Opt vs T EDD	≠ Opt vs T M
2134	22931	5676	7306	2184	2395	2607	2374	50	261	473	240
1996	17406	12276	7219	2011	2011	1996	2127	15	15	0	131

<b>2583</b>	17176	7432	4983	4684	2648	2810	2948	2101	65	227	365
<b>2691</b>	21051	13944	6423	3020	3320	2878	2787	329	629	187	96
<b>1518</b>	19952	10016	6257	1728	2396	1640	1647	210	878	122	129
<b>26276</b>	81335	54734	57759	27213	27152	26832	26740	937	876	556	464
<b>11403</b>	60459	30356	41594	11995	12178	11733	11997	592	775	330	594
<b>8499</b>	49632	36679	41492	8779	10532	8544	8711	280	2033	45	212
<b>9884</b>	50930	23660	35347	9935	9976	9964	10038	51	92	80	154
<b>10655</b>	61112	38212	33773	11097	10961	10879	10908	442	306	224	253
<b>43504</b>	136153	79547	100309	43938	45509	43850	44104	434	2005	346	600
<b>36378</b>	135441	80288	124903	36725	38006	37585	36811	347	1628	1207	433
<b>45383</b>	128823	78777	134513	46180	46382	46095	45616	797	999	712	233
<b>51785</b>	149488	89133	121646	52054	52892	52204	52509	269	1107	419	724
<b>38934</b>	92094	69852	102835	39257	39455	39381	39219	323	521	447	285
<b>87902</b>	255981	116121	180438	88090	88638	88354	95333	188	736	452	7431
<b>84260</b>	236534	132201	205270	85183	84514	84685	84525	923	254	425	265
<b>104795</b>	226285	145551	224623	105605	105792	105669	105402	810	997	874	607
<b>89299</b>	236756	142333	188949	89519	89548	89479	90272	220	249	180	973
<b>72316</b>	175152	109188	151729	72598	72942	72775	75354	282	626	459	3038
<b>214546</b>	395660	262387	335932	219109	235101	307881	257490	4563	20555	93335	42944
<b>150800</b>	261188	190545	243582	183256	156562	233584	159359	32456	5762	82784	8559
<b>224025</b>	404725	275305	403807	234884	226349	240511	244752	10859	2324	16486	20727
<b>116015</b>	282048	146286	286212	142867	120109	142962	152154	26852	4094	26947	36139
<b>240179</b>	344453	296162	365991	258345	297244	345601	248960	18166	57065	105422	8781
<b>2</b>	8411	12882	2	136	319	2	2	134	317	0	0
<b>4</b>	30061	14254	4	4	4	4	4	0	0	0	0
<b>755</b>	26135	15053	1516	879	1701	907	860	124	946	152	105
<b>99</b>	16689	9859	200	226	269	109	226	127	170	10	127
<b>22</b>	24057	12267	22	95	95	22	95	73	73	0	0
<b>9934</b>	63296	56835	49008	10670	11845	11420	10150	736	1911	1486	216
<b>7178</b>	45870	42609	31248	7610	7558	7400	7493	432	380	222	315
<b>4674</b>	55638	32709	14535	5021	4833	4979	4920	347	159	305	246
<b>4017</b>	53318	17344	24117	5021	4558	5025	4045	1004	541	1008	28
<b>6459</b>	53147	30577	29860	7502	10080	6837	6824	1043	3621	378	365
<b>34892</b>	113944	102922	120388	36501	41586	35217	37074	1609	6694	325	2182
<b>22739</b>	120340	54484	79608	23533	24662	22870	23178	794	1923	131	439
<b>29467</b>	136994	65825	97087	96709	31598	30195	29740	67242	2131	728	273
<b>49352</b>	172552	110893	141370	50078	51820	49693	50626	726	2468	341	1274
<b>26423</b>	127365	73126	104094	28239	28299	26631	27701	1816	1876	208	1278
<b>71111</b>	251461	129781	204798	73626	90939	110946	71808	2515	19828	39835	697
<b>90163</b>	232390	143687	226801	95174	95105	135530	93693	5011	4942	45367	3530
<b>84126</b>	224870	130219	195545	110606	86286	198920	84227	26480	2160	114794	101

<b>123893</b>	253117	175125	225955	156200	145934	128922	125240	32307	22041	5029	1347
<b>79883</b>	186923	126214	175057	84872	80195	91261	81090	4989	312	11378	1207
<b>157505</b>	340568	192376	298539	341047	162888	194916	181307	183542	5383	37411	23802
<b>133289</b>	297192	170331	263937	148887	161795	220611	150350	15598	28506	87322	17061
<b>191099</b>	329258	233246	300562	214382	206908	232604	219415	23283	15809	41505	28316
<b>150279</b>	273976	205853	273351	175410	206468	154352	160422	25131	56189	4073	10143
<b>198076</b>	395112	257742	312393	339346	257854	223979	210166	141270	59778	25903	12090
<b>0</b>	33052	11647	0	0	39	0	0	0	39	0	0
<b>0</b>	17888	23294	0	0	326	0	0	0	326	0	0
<b>0</b>	21128	10453	0	45	1332	0	0	45	1332	0	0
<b>0</b>	38215	27634	0	0	0	0	0	0	0	0	0
<b>0</b>	27566	15293	0	0	0	0	0	0	0	0	0
<b>1258</b>	70655	12382	5083	1325	1861	1258	1781	67	603	0	0
<b>3679</b>	60810	33628	10943	4590	6236	3872	3947	911	2557	193	268
<b>2522</b>	51610	37290	11369	2832	2854	2854	3658	310	332	332	1136
<b>3770</b>	75368	41718	29872	4354	6407	4773	4693	584	2637	1003	923
<b>5904</b>	62784	45650	24357	6557	6529	6706	6628	653	625	802	724
<b>25212</b>	133171	94949	99389	26066	27899	25266	28661	854	2687	54	3449
<b>17337</b>	137458	58915	65954	18343	21815	19081	17741	1006	4478	1744	404
<b>30729</b>	132320	98001	97640	32503	35445	102008	31924	1774	4716	71279	1195
<b>18082</b>	112164	81597	84817	19286	20989	92736	18651	1204	2907	74654	569
<b>25028</b>	144177	67826	93571	26294	28442	100746	26845	1266	3414	75718	1817
<b>76878</b>	199523	128416	167215	103933	89736	79624	77474	27055	12858	2746	596
<b>85413</b>	223741	138120	170451	88828	87806	133635	88153	3415	2393	48222	2740
<b>92756</b>	220094	188654	199586	95856	98940	205478	94803	3100	6184	112722	2047
<b>77930</b>	185194	138889	187205	144489	88992	80684	78936	66559	11062	2754	1006
<b>74750</b>	197436	124886	168720	94243	108372	111752	75902	19493	33622	37002	1152
<b>150580</b>	288281	203736	304790	190838	206013	176400	153847	40258	55433	25820	3267
<b>131680</b>	287299	167503	256573	135767	132805	167168	139771	4087	1125	35488	8091
<b>98494</b>	248657	129381	230362	249307	113040	120641	102458	150813	14546	22147	3964
<b>135394</b>	309696	179617	254277	168454	163315	151322	157100	33060	27921	15928	21706
<b>135677</b>	314106	186624	270617	216897	192002	172151	161354	81220	56325	36474	25677
<b>0</b>	23424	42980	0	0	0	0	0	0	0	0	0
<b>0</b>	55834	20178	0	0	432	0	0	0	432	0	0
<b>0</b>	44045	6075	0	0	0	0	0	0	0	0	0
<b>0</b>	24963	8338	0	0	0	0	0	0	0	0	0
<b>0</b>	24748	12427	0	0	20	0	0	0	20	0	0
<b>816</b>	96160	64005	1755	58318	2160	1040	1495	57502	1344	224	679
<b>4879</b>	63742	64869	27559	6116	9929	5301	5219	1237	5050	422	340
<b>973</b>	71002	41942	4312	2191	2933	1392	980	1218	1960	419	7
<b>508</b>	84030	71858	511	703	1215	511	633	195	707	3	125



<b>3780</b>	115866	45331	8754	5126	5096	11394	4321	1346	1316	7614	541
<b>20751</b>	154339	84986	64886	117857	35534	29227	21692	97106	14783	8476	941
<b>36053</b>	145888	122432	108686	37674	39982	40977	36475	1621	3929	4924	422
<b>28268</b>	157308	99441	72774	28922	28752	81778	30485	654	484	53510	2217
<b>28846</b>	150517	106388	84640	65900	32504	32931	29595	37054	3658	4085	749
<b>15451</b>	138292	73829	63934	18056	17982	51165	16114	2605	2531	35714	663
<b>89298</b>	226660	152393	238445	131667	152945	170439	89512	42369	63647	81141	214
<b>66340</b>	205683	118798	189101	107133	71723	81244	68424	40793	5383	14904	2084
<b>61060</b>	148749	101525	120264	83327	84323	66412	61189	22267	23263	5352	129
<b>42453</b>	198197	84596	142346	75444	61147	65592	43632	32991	18694	23139	1179
<b>56522</b>	189806	90374	200231	68467	80387	174946	58506	11945	23865	118424	1984
<b>177909</b>	294381	223357	288254	201819	184011	182166	187369	23910	6102	4257	9460
<b>139591</b>	249799	179821	264421	165019	182208	144856	143357	25428	42617	5265	3766
<b>148906</b>	310059	194843	273223	151801	151516	205888	159009	2895	2610	56982	10103
<b>179264</b>	331796	225389	280394	273578	190773	223089	192587	94314	11509	43825	13323
<b>120108</b>	234739	176430	239669	125510	124079	125974	120150	5402	3971	5866	42
<b>0</b>	64841	28818	0	0	0	0	0	0	0	0	0
<b>0</b>	34075	23850	0	0	0	0	0	0	0	0	0
<b>0</b>	13843	10513	0	0	0	0	0	0	0	0	0
<b>0</b>	61225	40742	0	0	0	0	0	0	0	0	0
<b>0</b>	58396	31924	0	674	1468	0	0	674	1468	0	0
<b>0</b>	76200	40783	0	354	0	0	0	354	0	0	0
<b>1717</b>	99414	51630	6201	2722	3120	7829	1845	1005	1403	6112	128
<b>0</b>	100563	64234	0	48	862	0	0	48	862	0	0
<b>6185</b>	79563	66536	19736	6548	38220	7340	6278	363	32035	1155	93
<b>1295</b>	80717	43109	3859	1811	1537	2187	1526	516	242	892	231
<b>27310</b>	164878	89844	60262	35200	35875	65095	28393	7890	8565	37785	1083
<b>15867</b>	165982	106931	40862	37864	109171	31881	16389	21997	93304	16014	522
<b>35106</b>	145138	121895	77386	59048	51891	50055	36482	23942	16785	14949	1376
<b>15467</b>	149660	85276	72503	33857	16718	31711	16427	18390	1251	16244	960
<b>10574</b>	118075	58975	28160	20439	11396	11164	12327	9865	822	590	1753
<b>35727</b>	128906	80007	125903	37293	36734	68704	35978	1566	1007	32977	251
<b>71922</b>	193650	135228	156498	90511	94061	85544	72176	18589	22139	13622	254
<b>65433</b>	217501	144982	186395	83028	145556	111022	65567	17595	80123	45589	134
<b>106043</b>	230194	156231	215037	109451	124578	210230	109349	3408	18535	104187	3306
<b>101665</b>	222738	167641	215990	134858	168030	104185	103049	33193	66365	2520	1384
<b>78315</b>	218283	123008	184413	103723	125484	83501	79527	25408	47169	5186	1212
<b>119925</b>	235226	181334	233472	217806	183571	180925	122848	97881	63646	61000	2923
<b>101157</b>	233798	153437	233838	107727	103731	151210	104279	6570	2574	50053	3122
<b>139488</b>	292842	189661	282998	188637	192172	181943	141278	49149	52684	42455	1790
<b>110392</b>	238472	152229	213292	130531	125853	151417	110523	20139	15461	41025	131

Tabla 42 Resultados completos de la instancia WT100

Best Val B	FIFO	SPT	EDD	Tabu FIFO	Tabu SPT	Tabu EDD	Tabu M	≠ BV vs T FIFO	≠ BV vs T SPT	≠ BV vs T EDD	≠ BV B
5988	14251	62290	14138	6433	6151	6151	6646	445	163	163	658
6170	14700	25635	19096	6905	6322	6372	6446	735	152	202	276
4267	9509	35555	17538	4651	4337	4399	4286	384	70	132	19
5011	16476	36579	13328	5164	5133	5220	5246	153	122	209	235
5283	10852	54806	20218	5771	8270	5433	5558	488	2987	150	275
58258	82192	174561	139851	58651	58265	58733	58857	393	7	475	599
50972	73447	147042	160094	52171	51504	51544	53528	1199	532	572	2556
59434	74184	133600	165341	59727	60573	61178	60487	293	1139	1744	1053
40978	60923	156295	145163	42275	41574	41062	41695	1297	596	84	717
53208	69507	98212	130119	54206	57620	56171	54589	998	4412	2963	1381
181649	214390	310137	471340	184059	186279	184901	183941	2410	4630	3252	2292
234179	271018	386847	497520	237096	240978	239120	237454	2917	6799	4941	3275
178840	213154	337733	541552	181133	184339	182103	182763	2293	5499	3263	3923
157476	181193	283620	411601	158941	159784	160424	158437	1465	2308	2948	961
172995	198805	289794	471867	176250	177417	175292	174098	3255	4422	2297	1103
407703	459734	593818	837928	408217	425106	417117	409131	514	17403	9414	1428
332804	364058	552540	770601	338355	345584	339221	334033	5551	12780	6417	1229
544838	a57929 8	701355	102513 8	547977	574132	560331	546133	3139	29294	15493	1295
477684	507535	691455	954119	478252	494158	486493	480424	568	16474	8809	2740
406094	445381	561303	849299	411372	418619	414366	406425	5278	12525	8272	331
898925	907837	115822 1	162992 3	901051	127959 5	921149	899066	2126	380670	22224	141
556873	562514	710817	108881 5	557133	608554	874833	557003	260	51681	317960	130
539716	545743	700620	108865 9	539792	601115	857124	539931	76	61399	317408	215
744287	748823	926256	136124 4	744505	755845	765517	744400	218	11558	21230	113
585306	589470	750271	107968 0	586243	921003	604147	585423	937	335697	18841	117
8	20008	40570	8	8	8	8	8	0	0	0	0
718	14045	38703	1857	965	778	778	1029	247	60	60	311
27	18350	33597	27	176	27	27	99	149	0	0	72
480	20095	45078	1825	853	480	682	806	373	0	202	326
50	17878	58214	50	50	50	50	97	0	0	0	47
24202	49530	114345	93384	26114	24632	24477	25138	1912	430	275	936
25469	81547	137485	105839	29115	25945	26499	27361	3646	476	1030	1892
32964	73278	140683	120702	34937	34280	34107	33816	1973	1316	1143	852
22215	56286	160206	77989	23906	24375	23249	24283	1691	2160	1034	2068
19114	49736	104132	73370	20165	20077	19783	19851	1051	963	669	737

<b>108293</b>	185299	266833	300485	114685	117189	117003	114813	6392	8896	8710	6520
<b>181850</b>	268568	393311	479037	188026	189953	191220	185839	6176	8103	9370	3989
<b>90440</b>	156177	302290	327402	93640	95361	94003	92235	3200	4921	3563	1795
<b>151701</b>	227422	330142	415814	161565	156393	154543	158650	9864	4692	2842	6949
<b>129728</b>	227502	353227	424422	140992	139655	138939	131215	11264	9927	9211	1487
<b>462324</b>	532187	743133	896872	467840	493829	659609	464976	5516	31505	197285	2652
<b>425875</b>	479760	637991	884631	450629	898765	501878	426577	24754	472890	76003	702
<b>320537</b>	375791	507412	838345	326447	350126	506789	321687	5910	29589	186252	1150
<b>360193</b>	426797	536585	794900	366825	387788	380925	362847	6632	27595	20732	2654
<b>306040</b>	384295	517306	793470	310156	394591	609587	306803	4116	88551	303547	763
<b>829828</b>	846086	106225 0	142959 0	832464	112841 6	120227 2	830136	2636	298588	372444	308
<b>623356</b>	633648	774916	114171 1	625996	679042	816828	623742	2640	55686	193472	386
<b>748988</b>	763549	908579	137135 3	751367	995683	111535 5	749085	2379	246695	366367	97
<b>656693</b>	668717	789415	127106 8	662378	674714	108385 9	656776	5685	18021	427166	83
<b>599269</b>	614514	749304	119983 4	602299	706564	101134 6	599507	3030	107295	412077	238
<b>0</b>	31139	65368	0	0	0	0	0	0	0	0	0
<b>0</b>	16330	31066	0	335	0	0	0	335	0	0	0
<b>0</b>	40491	67384	0	0	0	0	0	0	0	0	0
<b>0</b>	22022	74948	0	0	0	0	0	0	0	0	0
<b>0</b>	35183	75850	0	0	0	0	0	0	0	0	0
<b>9046</b>	78997	154191	36149	10921	9611	9701	10414	1875	565	655	1368
<b>11539</b>	106682	178968	37783	13309	15296	12635	15116	1770	3757	1096	3577
<b>16313</b>	101010	169442	73860	21375	17700	17441	18128	5062	1387	1128	1815
<b>7965</b>	82431	134952	29427	8157	8181	8230	9600	192	216	265	1635
<b>19912</b>	134922	188060	62698	25914	35862	22173	24746	6002	15950	2261	4834
<b>86793</b>	184045	288317	431388	93715	100451	98452	88779	6922	13658	11659	1986
<b>87067</b>	218687	367223	367251	104309	375573	97460	91857	17242	288506	10393	4790
<b>96563</b>	241392	326010	342052	109036	112612	109376	99288	12473	16049	12813	2725
<b>100788</b>	195413	343361	308552	108974	109596	107702	104263	8186	8808	6914	3475
<b>56510</b>	149022	240564	241495	61568	245497	63040	69416	5058	188987	6530	12906
<b>243872</b>	317037	436196	612578	251358	277614	344136	245601	7486	33742	100264	1729
<b>401023</b>	499107	630241	934984	406604	430139	652986	403054	5581	29116	251963	2031
<b>399085</b>	497626	639908	846090	443623	421609	421189	401235	44538	22524	22104	2150
<b>309232</b>	393734	543712	895491	334324	808321	538731	311493	25092	499089	229499	2261
<b>222684</b>	304339	404709	584663	234462	261797	502629	224390	11778	39113	279945	1706
<b>640816</b>	668714	838084	116405 6	659640	660395	794013	641786	18824	19579	153197	970
<b>611362</b>	641833	785209	112013 7	617524	977102	947050	611930	6162	365740	335688	568
<b>623429</b>	635866	873069	100313 7	629059	659496	639612	623651	5630	36067	16183	222
<b>584628</b>	616880	750772	106455 3	610238	711010	938999	585451	25610	126382	354371	823

<b>575274</b>	595079	764911	106278 1	576215	612169	680224	575456	941	36895	104950	182
<b>0</b>	36505	88319	0	0	0	0	0	0	0	0	0
<b>0</b>	22848	36103	0	0	0	0	0	0	0	0	0
<b>0</b>	40322	121286	0	0	0	0	0	0	0	0	0
<b>0</b>	36391	118959	0	0	0	0	0	0	0	0	0
<b>0</b>	64038	85477	0	0	0	0	0	0	0	0	0
<b>1400</b>	117669	208082	2737	13761	1637	1561	2390	12361	237	161	990
<b>317</b>	81087	280014	1343	2863	348	348	474	2546	31	31	157
<b>1146</b>	129098	162292	5100	7422	1509	1593	2373	6276	363	447	1227
<b>136</b>	78337	125722	136	513	136	136	136	377	0	0	0
<b>284</b>	117019	152816	284	791	284	284	478	507	0	0	194
<b>66850</b>	223609	322109	263656	87281	135074	139558	71166	20431	68224	72708	4316
<b>84229</b>	265796	378383	345826	99293	197121	243340	88624	15064	112892	159111	4395
<b>55544</b>	200770	290820	213933	70560	213406	213406	58380	15016	157862	157862	2836
<b>54612</b>	174519	300318	207118	87234	104282	138625	56826	32622	49670	84013	2214
<b>75061</b>	259389	378497	316409	142391	86940	165925	76675	67330	11879	90864	1614
<b>248699</b>	336404	509565	697419	313964	463988	442101	249158	65265	215289	193402	459
<b>311022</b>	447278	549715	735540	388946	713795	350832	334521	77924	402773	39810	23499
<b>326258</b>	429258	538925	710861	409575	428421	680120	356187	83317	102163	353862	29929
<b>273993</b>	370608	482351	735057	338934	588582	487561	276060	64941	314589	213568	2067
<b>316870</b>	466370	582005	801105	325723	493899	508741	324471	8853	177029	191871	7601
<b>495516</b>	532137	721377	103002 1	503393	529327	521369	496176	7877	33811	25853	660
<b>636903</b>	662669	809390	113732 3	654209	743879	874454	637074	17306	106976	237551	171
<b>680082</b>	703381	918415	111131 4	689527	782930	101699 3	680613	9445	102848	336911	531
<b>622464</b>	649640	828242	100983 3	626779	764110	653007	622715	4315	141646	30543	251
<b>449545</b>	491953	643385	933475	484102	669638	839899	450145	34557	220093	390354	600
<b>0</b>	46858	64326	0	0	0	0	0	0	0	0	0
<b>0</b>	39365	123679	0	0	0	0	0	0	0	0	0
<b>0</b>	58439	112901	0	0	0	0	0	0	0	0	0
<b>0</b>	69724	75681	0	0	0	0	0	0	0	0	0
<b>0</b>	59555	76255	0	0	0	0	0	0	0	0	0
<b>0</b>	89528	188510	0	0	0	0	0	0	0	0	0
<b>1193</b>	137035	223545	2915	10887	1252	1335	2048	9694	59	142	855
<b>0</b>	117773	196132	0	12850	0	0	0	12850	0	0	0
<b>232</b>	145452	205293	232	8214	232	232	519	7982	0	0	287
<b>0</b>	147332	208103	0	822	0	0	99	822	0	0	99
<b>159123</b>	401649	446661	436619	241293	275001	330445	206967	82170	115878	171322	47829
<b>174367</b>	357223	490262	453461	357816	424350	424350	198003	183449	249983	249983	23626
<b>91169</b>	315347	468556	302256	116497	122876	168230	96509	25328	31707	77061	5338
<b>168266</b>	349086	459508	410352	230144	358333	282521	174317	61878	190067	114255	6020

<b>70190</b>	264795	349099	203750	89256	86070	83837	78688	19066	15880	13647	8498
<b>370614</b>	473735	584717	736095	442958	521349	387963	389468	72344	150735	17349	18837
<b>324437</b>	455843	556254	723271	330648	430240	499450	359726	6211	105803	175013	35289
<b>246237</b>	375019	472687	759978	324063	692986	630884	272366	77826	446749	384647	26123
<b>293571</b>	424384	551692	799934	301489	716676	326876	319029	7918	423105	33305	25453
<b>267316</b>	380149	493262	656100	278770	423597	335353	304918	11454	156281	68037	37592
<b>471214</b>	553377	686849	107536 5	494335	549498	666232	482298	23121	78284	195018	11084
<b>570459</b>	613518	806496	103212 7	573797	841299	962562	570697	3338	270840	392103	238
<b>397029</b>	452492	614664	795916	442487	581354	715103	398150	45458	184325	318074	1121
<b>431115</b>	507707	672976	105187 7	445588	813778	981154	432539	14473	382663	550039	1424
<b>560754</b>	656413	824975	123727 1	626535	601644	635394	567224	65781	40890	74640	6470

## REFERENCIAS

---

- [1] Hamid Noori, Russel Radford, "Administración de Operaciones y Producción", Mc Graw Hill, México, 2000.
- [2] S. S. Panwalkar and Wafik Iskander, Operations Research, 25 (1): 45-61,1977.
- [3] Chen, Bo & Potts, Chris & Woeginger, Gerhard. A Review of Machine Scheduling: Complexity, Algorithms and Approximability. 10.1007/978-1-4613-0303-9\_25. 1998.
- [4] Thevenin, Simon & Zufferey, N. & Widmer, Marino. Tabu search for a *Single Machine Scheduling* problem with discretely controllable release dates. Proceedings of the 12th International Symposium on Operational Research in Slovenia, SOR 2013. 103-106. 2013.
- [5] H. A. J. Crauwels, C. N. Potts, and L. N. Van Wassenhove, Local Search Heuristics for the Single Machine Total Weighted Tardiness Scheduling Problem, INFORMS Journal on Computing 10:3, 341-350, 1998.
- [6] Du, Jianzhong, and Joseph Y.-T. Leung. "Minimizing Total Tardiness on One Machine Is NP-Hard." Mathematics of Operations Research, 15 (3): 483–495, 1990.
- [7] Glover, F., Laguna, M., & Marti, R. Principles of tabu search. Approximation algorithms and metaheuristics, 23, 1-12. 2007.
- [8] Tyagi, Neelam & Tripathi, R. & Chandramouli, A. *Single Machine Scheduling* Model with Total Tardiness Problem. Indian Journal of Science and Technology. 9: 4-14, 2016.
- [9] Karp R.M. Reducibility among Combinatorial Problems. In: Miller R.E., Thatcher J.W., Bohlinger J.D. (eds) Complexity of Computer Computations. The IBM Research Symposia Series. Springer, Boston, MA, 1972.
- [10] William J. Cook William H. Cunningham William R. Pulleyblank Alexander Schrijver, "Combinatorial Optimization", NY, United States, 1998.
- [11] Christos H. Papadimitriou, Kenneth Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover Publications Inc., Mineola, New York, 1998.

- [12] Cai-Min Wei, Ji-Bo Wang, Ping Ji, "Single-machine scheduling with time-and-resource-dependent processing times", *Applied Mathematical Modelling*, 36 (2): 792-798, 2012.
- [13] Martí, R. Procedimientos metaheurísticos en optimización combinatoria. *Matematiques*, 1 (1): 3-62, 2003.
- [14] Fred Glover, "Tabu Search – Part II", *Operations Research Society of America Journal on computing*, 2 (1): 4-32, 1990
- [15] Richard K. Congram, Chris N. Potts, and Steef L. van de Velde, "An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem", *INFORMS Journal on Computing* 14 (1): 52-67, 2002.
- [16] Felipe Andrés Riquelme Niklitschek, "Desarrollo de una solución logística para la programación de operaciones en una compañía siderúrgica", memoria para optar al título de ingeniero civil industrial, Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería Industrial, Santiago de Chile, Enero 2009.
- [17] Tamer Eren, Ertan Güner, "Minimizing total tardiness in a scheduling problem with a learning effect", *ScienceDirect, Applied Mathematical Modelling* 31: 1351-1361, 2007.
- [18] Christos Koulamas, The single-machine total tardiness scheduling problem: Review and extensions, *European Journal of Operational Research* 202: 1-7, 2010.
- [19] Tsung-Chyan Lai, Yuh-Kwo Kuo, "Minimizing Total Tardiness for Single Machine Secuencing", *Journal of the Operations Research Society of Japan*, 39 (3): 317-321 1996.
- [20] Álvarez-Valdés, "Algoritmos exactos y heurísticos para minimizar el adelantamiento y el retraso ponderados en una máquina con fecha común", XVI Jornadas de ASEPUMA y IV Encuentro Internacional, *Actas*, 16 (1): 704-7114, 2008.
- [21] Bo Chen, "A Review of Machine Scheduling: Complexity, Algorithms and Approximability", *Handbook of combinatorial optimization*: 21-169, Springer, Boston, MA, 1998.

- [22] Tyagi, Neelam & Tripathi, R. & Chandramouli, A. (2016). *Single Machine Scheduling* Model with Total Tardiness Problem. Indian Journal of Science and Technology. 9: 4-14, 2016.
- [23] Panneerselvam, "Simple heuristic to minimize total tardiness in a *Single Machine Scheduling* problem", R. Int J Adv Manuf Technol, 30: 722-726, 2006.
- [24] Puchinger J., Raidl G.R. Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In: Mira J., Álvarez J.R. (eds) Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. IWINAC 2005. Lecture Notes in Computer Science, 3562. Springer, Berlin, Heidelberg, 2005.
- [25] <http://people.brunel.ac.uk/~mastijb/jeb/orlib/wtinfo.html>. J.E.Beasley, June 1990; Last update: February 2018, visitado en diciembre de 2019.