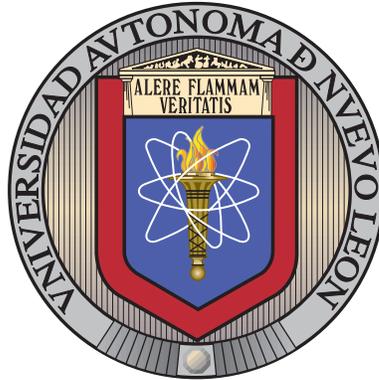


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



SECUENCIACIÓN DE TAREAS EN UNA
MÁQUINA CON TIEMPOS DE
PREPARACIÓN DEPENDIENTES DE LA
SECUENCIA Y ETAPAS FIJAS DE
MANTENIMIENTO PREVENTIVO

POR

PEDRO INES LOERA MARTÍNEZ

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

JUNIO 2015

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



SECUENCIACIÓN DE TAREAS EN UNA
MÁQUINA CON TIEMPOS DE
PREPARACIÓN DEPENDIENTES DE LA
SECUENCIA Y ETAPAS FIJAS DE
MANTENIMIENTO PREVENTIVO

POR

PEDRO INES LOERA MARTÍNEZ

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

JUNIO 2015

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Secuenciación de tareas en una máquina con tiempos de preparación dependientes de la secuencia y etapas de mantenimiento preventivo», realizada por el alumno Pedro Ines Loera Martínez, con número de matrícula 1340149, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis



Dra. Iris Abril Martínez Salazar

Asesor



Dra. Ada Álvarez Socarrás

Revisor



Dr. Francisco Angel-Bello Acosta

Revisor

Vo. Bo.

Dr. Simón Martínez Martínez

Subdirección de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, junio 2015

*A mi familia,
a mis amigos.*

ÍNDICE GENERAL

AGRADECIMIENTOS	XIII
RESUMEN	XIV
1. INTRODUCCIÓN	1
1.1. INTRODUCCIÓN A LOS PROBLEMAS DE SECUENCIACIÓN DE TAREAS	1
1.2. JUSTIFICACIÓN	4
1.3. HIPÓTESIS	5
1.4. OBJETIVOS	5
1.5. CONTRIBUCIONES	6
1.6. RESUMEN DEL CONTENIDO	7
1.7. CONCLUSIONES	7
2. MARCO TEÓRICO	9
2.1. NOTACIÓN PARA LOS PROBLEMAS DE SECUENCIACIÓN DE TAREAS	10
2.1.1. AMBIENTE DE MÁQUINA	10

2.1.2. CARACTERÍSTICAS DE LAS TAREAS	11
2.1.3. CRITERIOS DE OPTIMALIDAD	12
2.2. CATEGORÍAS PARA PROBLEMAS DE SECUENCIACIÓN DE TAREAS	13
2.3. CLASIFICACIÓN DEL CASO DE ESTUDIO	14
2.4. REVISIÓN DE LITERATURA	14
2.4.1. TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA	15
2.4.2. ETAPAS FIJAS DE MANTENIMIENTO PREVENTIVO	17
2.4.3. PROBLEMAS DE SECUENCIACIÓN EN UNA MÁQUINA CON TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA Y ETAPAS FIJAS DE MANTENIMIENTO PREVENTIVO	19
2.5. DIAGRAMA DE GANTT	19
3. DESCRIPCIÓN DEL PROBLEMA	21
3.1. PLANTEAMIENTO DEL PROBLEMA	21
3.1.1. SUPUESTOS	22
3.1.2. NOTACIÓN GENERAL	22
3.1.3. EJEMPLO GRÁFICO	23
3.2. MODELO A	24
3.2.1. OBTENCIÓN DE DESIGUALDADES VÁLIDAS	30
3.3. MODELO B	32

3.3.1. MEJORA DE LA FORMULACIÓN POR MEDIO DE LA ELIMINACIÓN DE VARIABLES	37
3.3.2. IMPLEMENTACIÓN EXTRA	42
4. PROCEDIMIENTO METAHEURÍSTICO	50
4.1. PARÁMETROS INICIALES	51
4.2. ESTRUCTURA GENERAL DEL ALGORITMO PROPUESTO . . .	52
4.2.1. MÉTODO CONSTRUCTIVO	52
4.2.2. BÚSQUEDA LOCAL	54
4.2.3. MOVIMIENTO 1	56
4.2.4. MOVIMIENTO 2	59
4.2.5. MOVIMIENTO 3	60
4.2.6. POST PROCESAMIENTO	61
5. EXPERIMENTACIÓN COMPUTACIONAL	66
5.1. INSTANCIAS	66
5.2. EXPERIMENTACIÓN DE LOS MODELOS MATEMÁTICOS . . .	67
5.3. EXPERIMENTACIÓN DEL PROCEDIMIENTO METAHEURÍSTICO	70
5.3.1. COMPARACIÓN CONTRA LA SOLUCIÓN ÓPTIMA	70
5.3.2. COMPARACIÓN CONTRA EL VECINO MÁS CERCANO .	72
6. CONCLUSIONES GENERALES	76
A. APÉNDICE A	78

A.1. RESULTADOS DE LAS FORMULACIONES MATEMÁTICAS . . .	78
B. APÉNDICE B	87
B.1. RESULTADOS DEL PROCEDIMIENTO METAHEURÍSTICO EN COMPARACIÓN CON LA SOLUCIÓN ÓPTIMA	87
C. APÉNDICE C	96
C.1. RESULTADOS DEL PROCEDIMIENTO METAHEURÍSTICO EN COMPARACIÓN CON EL VECINO MÁS CERCANO	96

ÍNDICE DE FIGURAS

2.1. Diagrama de Gantt	20
3.1. Representación gráfica del problema	23
3.2. Representación gráfica del cálculo de la función objetivo	24
3.3. Representación del problema mediante una red multinivel	25
3.4. Estructura gráfica usada en el Modelo B	33
3.5. Secuencia parcial	46
3.6. Lista de candidatos	46
3.7. Tiempos calculados en solución parcial	47
4.1. Secuencia de una solución factible	54
4.2. Proceso de búsqueda local	56
4.3. Cálculo de costos de transición	56
4.4. Representación de soluciones parciales	61

ÍNDICE DE TABLAS

3.1. Tiempos de procesamiento	40
3.2. Tiempos de preparación dependiente de la secuencia	41
3.3. Tiempos totales de procesamiento	41
5.1. Porcentaje de instancias resueltas	68
5.2. Tiempos de ejecución del Modelo A	69
5.3. Tiempos de ejecución del Modelo B	69
5.4. Experimentación 1 del método metaheurístico	71
5.5. Experimentación 2 del método metaheurístico	74
A.1. Instancias de $n = 10$ y $T = 2.25dm$	79
A.2. Instancias de $n = 10$ y $T = 2.5dm$	79
A.3. Instancias de $n = 10$ y $T = 3dm$	80
A.4. Instancias de $n = 10$ y $T = 4dm$	80
A.5. Instancias de $n = 12$ y $T = 2.25dm$	81
A.6. Instancias de $n = 12$ y $T = 2.5dm$	81
A.7. Instancias de $n = 12$ y $T = 3dm$	82

A.8. Instancias de $n = 12$ y $T = 4dm$	82
A.9. Instancias de $n = 15$ y $T = 2.25dm$	83
A.10. Instancias de $n = 15$ y $T = 2.5dm$	83
A.11. Instancias de $n = 15$ y $T = 3dm$	84
A.12. Instancias de $n = 15$ y $T = 4dm$	84
A.13. Instancias de $n = 20$ y $T = 2.25dm$	85
A.14. Instancias de $n = 20$ y $T = 2.5dm$	85
A.15. Instancias de $n = 20$ y $T = 3dm$	86
A.16. Instancias de $n = 20$ y $T = 4dm$	86
B.1. Instancias de $n = 10$ y $T = 2.25dm$	88
B.2. Instancias de $n = 10$ y $T = 2.5dm$	88
B.3. Instancias de $n = 10$ y $T = 3dm$	89
B.4. Instancias de $n = 10$ y $T = 4dm$	89
B.5. Instancias de $n = 12$ y $T = 2.25dm$	90
B.6. Instancias de $n = 12$ y $T = 2.5dm$	90
B.7. Instancias de $n = 12$ y $T = 3dm$	91
B.8. Instancias de $n = 12$ y $T = 4dm$	91
B.9. Instancias de $n = 15$ y $T = 2.25dm$	92
B.10. Instancias de $n = 15$ y $T = 2.5dm$	92
B.11. Instancias de $n = 15$ y $T = 3dm$	93
B.12. Instancias de $n = 15$ y $T = 4dm$	93

B.13. Instancias de $n = 20$ y $T = 2.25dm$	94
B.14. Instancias de $n = 20$ y $T = 2.5dm$	94
B.15. Instancias de $n = 20$ y $T = 3dm$	95
B.16. Instancias de $n = 20$ y $T = 4dm$	95
C.1. Instancias de $n = 20$ y $T = 2.25dm$	97
C.2. Instancias de $n = 20$ y $T = 2.5dm$	97
C.3. Instancias de $n = 20$ y $T = 3dm$	97
C.4. Instancias de $n = 20$ y $T = 4dm$	98
C.5. Instancias de $n = 30$ y $T = 2.25dm$	99
C.6. Instancias de $n = 30$ y $T = 2.5dm$	99
C.7. Instancias de $n = 30$ y $T = 3dm$	100
C.8. Instancias de $n = 30$ y $T = 4dm$	100
C.9. Instancias de $n = 40$ y $T = 2.25dm$	101
C.10. Instancias de $n = 40$ y $T = 2.5dm$	101
C.11. Instancias de $n = 40$ y $T = 3dm$	102
C.12. Instancias de $n = 40$ y $T = 4dm$	102
C.13. Instancias de $n = 50$ y $T = 2.25dm$	103
C.14. Instancias de $n = 50$ y $T = 2.5dm$	103
C.15. Instancias de $n = 50$ y $T = 3dm$	104
C.16. Instancias de $n = 50$ y $T = 4dm$	104

AGRADECIMIENTOS

Esta tesis representa el trabajo y dedicación de muchas personas, quienes a lo largo de estos dos años y medio colaboraron para que yo pudiera concluir este objetivo.

Principalmente me gustaría agradecer a mi asesora de tesis la Dra. Iris Abril Martínez, quien me brindo la orientación y el apoyo necesario para el desarrollo de este proyecto. Además por brindarme su amistad y confianza a lo largo de este periodo.

A los miembros del comité de tesis la Dra. Ada Álvarez y el Dr. Francisco Angel-Bello, quienes enriquecieron este trabajo con sus comentarios.

También agradezco a los profesores del Posgrado en Ingeniería de Sistemas, quienes demuestran una gran vocación y compromiso por la investigación.

Quiero agradecer a la institución, la Facultad de Ingeniería Mecánica y Eléctrica, quien me proporciono un espacio para el desarrollo de mis actividades diarias, y a su personal docente que siempre me brindo un buen servicio. Agradezco al Consejo Nacional de Ciencia y Tecnología por otorgarme una beca para realizar mis estudios.

Agradezco a mis compañeros, quienes siempre estuvieron a la disposición para apoyarme, sin duda esto hubiera sido más difícil, además, les quiero agradecer por brindarme su amistad y las anécdotas compartidas.

Finalmente, agradezco a mi familia, quienes siempre me apoyaron y respaldaron cada una de mis decisiones.

RESUMEN

Pedro Ines Loera Martínez.

Candidato para el grado de Maestría en Ciencias en
Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

SECUENCIACIÓN DE TAREAS EN UNA MÁQUINA CON TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA Y ETAPAS FIJAS DE MANTENIMIENTO PREVENTIVO

Número de páginas: 111.

OBJETIVOS: El propósito principal de esta investigación es el proporcionar un estudio detallado que sea útil como herramienta y permita obtener buenos resultados a un problema de secuenciación de tareas en una máquina simple con el objetivo de minimizar el tiempo total de flujo, considerando factores importantes como, tiempos de preparación dependientes de la secuencia y etapas fijas de mantenimiento preventivo, es decir tiempos en que la máquina no se encuentra disponible para procesar tareas.

MÉTODO DE ESTUDIO:

- Paso 1** Realizar una revisión de literatura, para identificar aquellos trabajos de investigación que presenten problemas relacionados y llevar a cabo un análisis a los procedimientos propuestos en las distintas publicaciones encontradas.
- Paso 2** Formular distintos modelos matemáticos, de manera que sea posible obtener soluciones óptimas, para cada uno de ellos se realizará un estudio que nos permitirá determinar el alcance para resolver cierta variedad y tamaño de instancias.
- Paso 3** Implementar procedimientos de eliminación de variables y el uso de desigualdades válidas, con el propósito de mejorar cada una de las formulaciones propuestas.
- Paso 4** Diseño de métodos metaheurísticos que permitan obtener resultados de calidad, en un tiempo computacional considerable.
- Paso 5** Realizar la experimentación computacional y el análisis de resultados.

CONTRIBUCIONES Y CONCLUSIONES: En este trabajo se presenta una nueva variante de un problema de secuenciación de tareas, para el cual se presentan dos modelos de programación lineal entera mixta basados en formulaciones propuestas para problemas relacionados, en la primera formulación se hace uso de desigualdades válidas para mejorar la relajación lineal del modelo, en la segunda formulación se implementa un pre-procesamiento que permite la eliminación variables, y el uso de un método heurístico que brinda una cota superior a la solución óptima del problema. También, se presenta un procedimiento metaheurístico híbrido que proporciona buenos resultados en un tiempo de computo bajo.

Debido a las implementaciones realizadas a cada uno de los modelos, se logró mejorar su desempeño computacional, así como también, aumentar el alcance de los

mismos, con respecto al tamaño de las instancias a resolver. En lo que corresponde al procedimiento metaheurístico, este proporciona resultados buenos, muy cercanos a los óptimos encontrados y brinda buenas soluciones en aquellos casos en que el modelo matemático resulta ineficiente.

Firma del asesor: _____



Dra. Iris Abril Martínez Salazar

CAPÍTULO 1

INTRODUCCIÓN

En la actualidad el sector industrial se enfrenta con un gran número de problemas como lo son los económicos, sociales, de seguridad y medio ambiente, control de calidad, entre otros. Por este motivo se recurre a distintas áreas de la investigación buscando soluciones a ellos; una de las principales es la investigación de operaciones (IO) la cual se define como la implementación del método científico en la toma de decisiones donde se busca la mejor forma de operar un sistema, considerando recursos limitados (Winston, 2006).

En una industria existe una gran cantidad de actividades, cada una de ellas puede ser representada como un sistema, los cuales operan en conjunto tratando de cumplir con las metas globales de la empresa. La eficiencia de cada uno de esos sistemas es vital para que se logren dichos objetivos, es por eso que el estudio, diseño e implementación de métodos de optimización, toman una gran relevancia en la dirección y toma de decisiones en cada uno de los proyectos que se lleven a cabo.

1.1 INTRODUCCIÓN A LOS PROBLEMAS DE SECUENCIACIÓN DE TAREAS

La planeación de la producción es una de las principales actividades que se realizan en una empresa, haciendo uso de ella como una herramienta para determinar la cantidad de producto que se va a producir en un período de tiempo, con

el objetivo de prevenir algunos problemas como: escases de mano de obra, falta de materia prima y disponibilidad de maquinaria o equipo. La correcta planeación de la producción puede traer muchos beneficios para una compañía, ayuda a calcular la falta de recursos en base a lo producido anteriormente y también permite estimar los recursos económicos necesarios para llevar a cabo la producción.

En la planeación de la producción, el problema de secuenciación de tareas o *scheduling* es uno de los principales, el cual se puede definir como la asignación de tareas a recursos escasos (Baker, 1974), esto puede ser disponibilidad de maquinaria, existencia de materia prima, mano de obra, etc. Una buena planeación de la producción ayudará a que el o los objetivos propuestos por la empresa se cumplan. En general dicha planeación no es trivial, por lo que es necesario el desarrollo de herramientas que ayuden a la toma de decisión de la asignación de dichos recursos.

Los problemas de secuenciación de tareas están directamente relacionados al problema del agente viajero y al problema del reparador. El problema del agente viajero o TSP por sus siglas en inglés (*Traveling Salesman Problem*) pertenece a los problemas del área de optimización combinatoria y ha sido intensamente investigado por varios autores (Burkard et al., 1998).

El TSP se describe de la siguiente forma, un agente viajero debe visitar n ciudades, comenzando y finalizando en su propia ciudad. Conociendo el costo de ir de cada ciudad a otra, el objetivo consiste en buscar el recorrido que proporcione el costo mínimo de la ruta. Este problema, ha sido uno de los más estudiados, probando Karp (1972) que pertenece a la clase de los problemas difíciles (*NP-hard*).

Por otro lado, el problema del reparador o TRP por sus siglas en inglés (*Traveling Repairman Problem*), también llamado problema de mínima latencia es definido como, un reparador debe visitar n clientes para realizar un servicio, partiendo desde un lugar de origen. Se conoce el costo en tiempo de ir de cada cliente a otro y un tiempo adicional, que representa el tiempo que se necesita para cubrir el servicio requerido. El objetivo del TRP es encontrar una ruta que visite todos los

clientes una sola vez, la cual minimice el tiempo total de espera, en otras palabras, la sumatoria de los tiempos de espera de cada uno de los clientes para que su servicio sea realizado.

En un problema de secuenciación de tareas, cuando se tiene la presencia de tiempos de preparación dependientes de la secuencia, resulta sencillo el realizar una analogía con el TRP, considerando las tareas como los clientes a visitar y a los tiempos de preparación de la máquina como los costos en tiempo entre dos clientes diferentes.

Este trabajo de tesis se enfoca en resolver un problema de secuenciación de tareas, que tiene como función objetivo el encontrar una secuencia que minimice el tiempo total de flujo, es decir, minimizar la suma de los tiempos en que se finaliza cada una de las tareas, y con esto reducir el tiempo que esperan cada uno de los clientes para recibir sus pedidos.

Cada tarea se tiene disponible desde el tiempo cero, además, se consideran ininterrumpidas, una vez comenzada una tarea esta debe de terminarse antes de realizar la siguiente actividad. Para cada tarea se tiene un tiempo de procesamiento, que corresponde al tiempo que se requiere para que una tarea sea realizada en la máquina.

En este problema, también se presentan otros elementos importantes al momento de realizar la planeación, como lo es la implementación de etapas fijas de mantenimiento preventivo. La mayoría de los problemas de secuenciación suponen que la maquinaria siempre estará disponible para procesar tareas, pero en la práctica esto no siempre es cierto, pueden ocurrir paros debido a fallas en la máquina, errores de mano de obra o a la implementación de un plan maestro de mantenimiento preventivo.

Otro factor importante es el de tomar en cuenta los tiempos de preparación dependientes de la secuencia, los cuales se pueden deber a procedimientos de limpieza, mantenimiento de herramientas, calibración de maquinaria, cambio de materia

prima, entre otros.

En la publicación de Ángel-Bello et al. (2011a) se presenta un problema de secuenciación de tareas conformado por la misma estructura del problema presentado en esta tesis, pero la función objetivo se enfoca en encontrar la secuencia que minimice el máximo tiempo de terminación de las tareas, dicho problema es clasificado como NP-*hard*.

Entre las investigaciones que se revisaron acerca de problemas de secuenciación de tareas, no se ha encontrado alguna publicación que trate el problema con las características propuestas en este trabajo.

1.2 JUSTIFICACIÓN

En algunas industrias como la metalurgia, petroquímica, aeroespacial, farmacéutica, electrónica, cerámica y automotriz, los sistemas de producción pueden contener una máquina “ *cuello de botella* ” que es aquella que da servicio a varias líneas de producción, la cual afecta el flujo de todos los procesos de trabajo. También puede verse todo un proceso de producción como una sola máquina, en la que el tiempo de procesamiento de un producto corresponde a su tiempo de fabricación total. Es por este motivo que resulta relevante el estudio de problemas sobre una máquina simple.

En lo que respecta a los tiempos de preparación dependientes de la secuencia, varios estudios han mostrado que algunas máquinas, si no es que todas, cuentan con tiempos de preparación dependientes de la secuencia. Dudek et al. (1974) reportaron que el 70 % de las actividades industriales incluye tiempos de preparación, mientras que, Conner (2009) recientemente comenta que de 250 proyectos industriales, el 50 % contienen tiempos de preparación.

Finalmente el mantenimiento preventivo corresponde a todas aquellas actividades que son realizadas con el objetivo de mantener en buenas condiciones los bienes

de la empresa (Cruzan, 2009). En lo que respecta a maquinaria puede ir desde el engrasado de chumaceras, cambio de filtros, pintura, entre otras actividades.

La buena planeación de un programa de mantenimiento preventivo extiende el tiempo de vida de los equipos, ayuda con la reducción de los tiempos muertos debido a fallas en la maquinaria. Es cierto que la implementación de un plan de mantenimiento preventivo recurre a un costo económico para la empresa, pero si se omite este, puede generar costos mayores a la misma, por daños en la infraestructura o como resultado de accidentes laborales en que pueden poner en peligro la vida del personal.

1.3 HIPÓTESIS

Para el problema de secuenciación de tareas propuesto en esta tesis es posible realizar un modelo matemático, el cual, es difícil de resolver de manera exacta en un tiempo de cómputo considerable haciendo uso de un software comercial. La implementación de un pre-procesamiento y el uso de desigualdades válidas beneficiaran el rendimiento de la formulación. Además y basado en la dificultad del problema, se requiere el diseño de procedimientos metaheurísticos y medios de computación adecuados para encontrar soluciones factibles y de buena calidad.

1.4 OBJETIVOS

- Identificar a través de una revisión de literatura las distintas investigaciones acerca de problemas relacionados al caso de estudio, y analizar los procedimientos propuestos y aportaciones.
- Plantear diferentes formulaciones matemáticas, que sean útiles para encontrar soluciones exactas al problema propuesto. Además determinar el alcance que tiene cada una, con respecto al tamaño de las instancias que sean resueltas a optimalidad.

- Diseñar procedimientos metaheurísticos que permitan encontrar soluciones de buena calidad en un tiempo computacional bajo, sobre todo en aquellos casos donde los distintos modelos planteados son considerados ineficientes.
- Realizar una experimentación para cada una de las metodologías y concluir sobre los resultados obtenidos.

1.5 CONTRIBUCIONES

Como contribución de este trabajo, primeramente podemos mencionar la nueva variante de un problema de secuenciación de tareas en una máquina simple, con objetivo de minimizar el tiempo total de flujo, en el que además, se toman en cuenta tiempos de preparación dependientes de la secuencia y etapas fijas de mantenimiento preventivo, hasta donde se sabe este problema con estas características no ha sido estudiado aún.

Se presentan dos modelos de programación lineal entera mixta basados en formulaciones propuestas para problemas relacionados, cada uno de los modelos es sometido a experimentación y posterior a esto se lleva a cabo la implementación de mejoras para aumentar su desempeño.

En la primera formulación se hace uso de desigualdades válidas de forma de mejorar la relajación lineal del modelo, para la segunda formulación se implementa un pre-procesamiento que permite la eliminación variables, y el uso de un método heurístico que brinda una cota superior a la solución óptima del problema.

Una vez obtenido el alcance de cada uno de los modelos matemáticos, para aquellas instancias en que resultaron ineficientes, se presenta un procedimiento metaheurístico híbrido que proporciona buenos resultados en un tiempo de computo bajo.

Se espera que los métodos de solución presentados sean útiles para solucionar

otros problemas de secuenciación de tareas o problemas con estructuras similares a este.

1.6 RESUMEN DEL CONTENIDO

En el Capítulo 2 se describe la notación y clasificación acerca de los problemas de secuenciación de tareas, además se realiza una revisión de literatura de problemas similares o relacionados con el caso de estudio.

En el Capítulo 3 se describe el problema del caso de estudio de esta tesis, se determinan los supuestos con los que se va a trabajar; además es aquí donde se lleva a cabo la explicación de las distintas formulaciones matemáticas, así como también se plantean diversos procedimientos de mejora.

En el Capítulo 4 se plantea el procedimiento metaheurístico propuesto, describiendo cada una de las estrategias que los componen para encontrar soluciones de buena calidad para este problema.

En el Capítulo 5 se muestra la experimentación general para las distintas metodologías propuestas, tanto las formulaciones matemáticas como el procedimiento metaheurístico. Esta sección tiene como anexo los apéndices A, B y C, donde se muestran los resultados individuales para las distintas instancias de prueba.

En el capítulo 6 se presentan las conclusiones generales sobre el estudio realizado en esta tesis, además se mencionan algunos puntos como trabajo futuro.

1.7 CONCLUSIONES

En esta tesis se presenta un nuevo problema de secuenciación de tareas, el cual considera factores importantes, como lo son los tiempos de preparación dependientes de la secuencia y la implementación de algún programa de mantenimiento preventivo, además de considerar una función objetivo que se enfoca principalmente en el cliente.

Este problema se considera que es difícil de resolver mediante el uso de un *software* de optimización a partir de cierto tamaño de instancias, por lo cual fue necesario el desarrollo de un procedimiento metaheurístico que explote la estructura del problema y nos permita el encontrar buenas soluciones en un tiempo de computo bajo.

CAPÍTULO 2

MARCO TEÓRICO

Hoy en día las empresas se enfrentan a una gran variedad de retos, con el objetivo de mantener la competitividad en el mercado y buenas relaciones con sus clientes.

Los problemas de secuenciación de tareas es uno de los más importantes entre las líneas de producción del sector industrial, es por ello, que en las últimas décadas se han invertido gran cantidad de recursos tratando de encontrar mejores estrategias para la toma de decisiones, que ayuden al diseño el diseño del plan de actividades.

La secuenciación de tareas es la actividad que consiste en determinar cuándo debe iniciar y terminar cada operación, de esta forma indicar en qué momento se debe realizar, basado en la escases de recursos, como tiempo o capacidad de producción. Este trabajo no es simple ni sencillo.

Al resolver un problema de secuenciación se busca realizar la asignación de tiempos de inicio y termino para diversas tareas, todo esto en escala del tiempo de máquina disponible.

La definición de secuenciación de tareas es común entre distintas investigaciones, Baker (1974) la describe como *la actividad de distribuir los recursos a lo largo del tiempo, para realizar un conjunto de tareas*. Por otro lado Morton (1993) la define como *el proceso de organizar, elegir y sincronizar los recursos, para llevar a cabo todas las actividades necesarias con el objetivo de generar los productos deseados, en el tiempo deseado, mientras se satisface cierto número de restricciones de relación*

entre las actividades, tiempos y recursos.

En este capítulo se mostrara la nomenclatura empleada para describir las distintas variantes de los problemas de secuenciación de tareas, como también la forma en que se clasifican estos problemas y algunas metodologías de solución. Posterior a esto se presenta una revisión de literatura acerca de casos relacionados con el problema de secuenciación de tareas en una máquina simple, con tiempos de preparación dependientes de la secuencia y etapas fijas de mantenimiento preventivo.

2.1 NOTACIÓN PARA LOS PROBLEMAS DE SECUENCIACIÓN DE TAREAS

Graham et al. (1979) introducen la siguiente notación para la clasificación de los problemas de secuenciación de tareas, la cual ha sido altamente empleada en otras investigaciones.

Se supone que n tareas $J_j (j = 1, \dots, n)$ tienen que ser procesadas en m máquinas $M_i (i = 1, \dots, m)$. Además se supone que cada máquina puede procesar un solo trabajo a la vez. La diversidad de tareas, máquinas y características de secuenciación se representan en 3 principales parámetros $\alpha|\beta|\gamma$. Donde α representa el ambiente de máquina, β especifica las características de las tareas y γ denota los criterios de optimalidad.

2.1.1 AMBIENTE DE MÁQUINA

En lo que corresponde al parámetro α se clasifica en dos diferentes criterios, $\alpha \in \{\alpha_1, \alpha_2\}$. Los posibles valores para α_1 son $\emptyset, P, Q, R, F, J, O, FF$. Si $\alpha_1 \in \{\emptyset, P, Q, R\}$, entonces cada trabajo consiste en una tarea simple. Si $\alpha_1 = \emptyset$, cada trabajo debe ser realizado sobre una máquina específica.

Si $\alpha_1 \in \{P, Q, R\}$, entonces se consideran máquinas en paralelo, esto es que

cada tarea puede ser procesada en cualquiera de las máquinas M_1, \dots, M_m . Si $\alpha_1 = P$, se tiene que M_1, \dots, M_m son **máquinas idénticas en paralelo**. Si el tiempo de procesamiento de la tarea J_i en la máquina M_j es denotado como $p_{i,j}$, para este caso $p_{i,j}$ es igual a p_i . Si $\alpha_1 = Q$, se cuenta con **máquinas en paralelo uniformes**, esto es $p_{i,j} = p_i/s_j$ donde s_j representa la velocidad de la máquina M_j . Si $\alpha_1 = R$, se tienen **máquinas en paralelo no relacionadas**, esto es $p_{i,j} = p_i/s_{i,j}$, donde la tarea depende de la velocidad $s_{i,j}$ de la máquina M_j .

Si $\alpha_1 \in \{F, J, O, FF\}$, se presenta un modelo de multi-operación, esto es que cada tarea J_i representa un conjunto de operaciones $O_{i,1}, \dots, O_{i,n_i}$.

En un **job shop**, indicado por $\alpha_1 = J$, se tiene una especial relación de precedencia de la forma

$$O_{i,1} \rightarrow O_{i,2} \rightarrow O_{i,3} \rightarrow \dots \rightarrow O_{i,n_i} \quad \text{for } i = 1, \dots, n.$$

El **flow shop** está indicado cuando $\alpha_1 = F$, este es un caso especial del *job shop* en el cual $n_i = m$ para $i = 1, \dots, n$. El **open shop**, denotado por $\alpha_1 = O$, es descrito como el *flow shop*, con la excepción que no hay relación de precedencia entre las operaciones. Finalmente si $\alpha_1 = FF$, representa el **flexible flow shop**.

Si el criterio α_2 es igual a un número entero positivo $1, 2, \dots$, entonces α_2 denota el número de máquinas.

En más recientes investigaciones se muestra un tercer criterio, α_3 (Ma et al., 2010), el cual es considerado para representar los tiempos de no disponibilidad de la maquinaria. Si $\alpha_3 = h_{i,l}$ indica que existen l intervalos de no disponibilidad en la máquina M_i .

2.1.2 CARACTERÍSTICAS DE LAS TAREAS

Las características de las tareas están especificadas en un grupo β que contiene a lo mas cinco elementos $\beta_1, \beta_2, \beta_3, \beta_4$ y β_5 .

β_1 indica si la interrupción de las tareas está permitida. Entendiendo por interrupción, que una tarea puede ser detenida en su proceso y ser continuada más tarde, en la misma o en alguna otra máquina. Si la interrupción es permitida $\beta_1 = pmtn$, en otro caso β_1 no aparece.

β_2 describe el *ready time* o tiempos de disponibilidad de las tareas para que estas puedan ser procesadas. Si $\beta_2 = r_j$ indica que existe un tiempo a partir del cual puede ser procesada la tarea J_j , en otro caso β_2 no aparece.

β_3 representa el tiempo de entrega de las tareas. Si $\beta_3 = d_j$ se tiene un tiempo d_j en el cual a lo mas debe ser terminada la tarea J_j , en otro caso β_3 no aparece.

β_4 indica la información que se tiene acerca la tareas. Si $\beta_4 = online$ representa que se conoce toda la información para llevar a cabo la planeación de la producción, en otro caso β_4 no aparece y esta se considera *offline*.

Finalmente β_5 muestra la existencia de tiempos de preparación dependientes de la secuencia. Si $\beta_5 = s_{j,k}$ indica que existen tiempos de preparación que dependen de la secuencia, mientras que si $\beta_5 = s_{i,j,k}$ indica que existen tiempos de preparación dependientes de la secuencia pero que además estos tiempos también dependen de la máquina, en caso de que no existan tiempos de preparación β_5 no aparece.

2.1.3 CRITERIOS DE OPTIMALIDAD

En un problema de secuenciación de tareas se busca por lo general encontrar aquella secuencia factible, la cual minimice el total de la función de costo. Se denotara como C_i al tiempo de terminación de la tarea J_i .

Las más comunes funciones objetivo son el *makespan* $\max\{C_i | i = 1, \dots, n\}$, el tiempo total de flujo $\sum_{i=1}^n C_i$, y el tiempo de flujo ponderado $\sum_{i=1}^n w_i C_i$. En estos casos se escribe como $\gamma = C_{max}$, $\gamma = \sum C_i$, $\gamma = \sum w_i C_i$, respectivamente.

Otros objetivos dependen de los tiempos límites de terminación de las tareas. Se define para cada trabajo J_i :

$L_i := C_i - d_i$	lateness
$E_i := \max\{0, d_i - C_i\}$	earliness
$T_i := \max\{0, C_i - d_i\}$	tardiness
$D_i := C_i - d_i $	absolute deviation
$S_i := (C_i - d_i)^2$	squared deviation

De esta forma se obtienen otros objetivos que son ampliamente usados como:

$$\sum T_i, \sum w_i T_i, \sum D_i, \sum w_i D_i, \sum S_i, \sum w_i S_i, \sum E_i, \sum w_i E_i.$$

2.2 CATEGORÍAS PARA PROBLEMAS DE SECUENCIACIÓN DE TAREAS

Los problemas de secuenciación de tareas pueden ser categorizados como:

- a) **Problema de secuenciación determinista:** cuando todos los elementos del problema, tales como el tiempo de disponibilidad de las tareas, tiempos de entrega, orden, tiempos de procesamiento y tiempos de disponibilidad de la maquinaria, no incluyen factores estocásticos y son determinados inicialmente.
- b) **Problema de secuenciación estático:** al igual que los problemas deterministas, con la excepción de que los tiempos de arribo de las tareas se consideran fijos a lo largo del tiempo y se encuentran disponibles desde el inicio.
- c) **Problema de secuenciación dinámico:** las tareas a procesar cambian a lo largo del tiempo y estas llegan en diferentes momentos.
- d) **Problema de secuenciación estocástico:** por lo menos algún elemento del problema incluye factores estocásticos.

2.3 CLASIFICACIÓN DEL CASO DE ESTUDIO

En base a lo presentado anteriormente en este capítulo, ahora se pretende mostrar la nomenclatura y la clasificación del problema de secuenciación de tareas tratado en esta tesis.

En la notación $\alpha|\beta|\gamma$:

$\alpha = 1, h_{1,l}$, haciendo referencia a que el proceso se realiza en una máquina simple, y se tienen periodos de no disponibilidad de maquinaria.

$\beta = s_{i,j}$, que especifica la existencia de tiempos dependientes de la secuencia entre las actividades a procesar en la máquina.

$\gamma = \sum C_j$, indica que la función objetivo del problema corresponde a la sumatoria de los tiempos de terminación de cada una de las tareas, es otras palabras al tiempo total de flujo.

Este problema esta categorizado como un problema de secuenciación estático, debido a que se conoce toda la información acerca del problema desde el inicio, así como también se consideran las tareas disponibles desde el tiempo cero.

2.4 REVISIÓN DE LITERATURA

Existe una gran variedad de estudios relacionados acerca de problemas de secuenciación de tareas en una máquina simple, los cuales buscan resolver variantes de estos, basado en la conclusión de que no todos los sistemas industriales cumplen con las mismas características.

Este tipo de problemas ha recibido mucha atención desde los trabajos publicados por Jackson (1955) y Smith (1956), se categorizan por ser difíciles de resolver y llegar a tener una gran complejidad. Además la mayoría de los problemas con m -máquinas los utilizan como sub-problemas, con el propósito de desarrollar mejores algoritmos de solución.

Por otro lado, a pesar de que hay una gran variedad de estudios relacionados con estos problemas, existen también otras variantes que no han sido estudiadas aun.

Un factor considerable al momento de realizar un plan de producción, es el de considerar los tiempos de preparación dependientes de la secuencia, que son el resultado del servicio que se debe realizar antes y después de alguna tarea. En las publicaciones de Conner (2009) y Dudek et al. (1974) se menciona la frecuencia con la que se observan estos factores en el desarrollo de proyectos en una empresa.

Otro punto importante que debe ser tomado en cuenta, son los tiempos de disponibilidad de equipo, es frecuente entre las publicaciones establezcan como supuesto, que la máquina siempre se encuentra disponible para realizar una tarea, pero en la realidad esto no es del todo cierto, pueden suceder paros debido a fallas o a servicios de mantenimiento preventivo ya programados.

2.4.1 TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA

En los últimos años, la cantidad de publicaciones, acerca de problemas de secuenciación de tareas que consideren tiempos de preparación dependientes de la secuencia, han amentado considerablemente. La investigación desarrollada por Allahverdi y Soroush (2008), enlista algunas de las publicaciones más relevantes que hacen uso de estos tiempos, así como también mencionan los beneficios obtenidos al considerar este criterio.

Entre la gran cantidad de artículos acerca de estos problemas de secuenciación de tareas en una máquina y que consideren tiempos de preparación, uno de los objetivos más frecuente en la literatura, es el de minimizar la tardanza ponderada total, el cual fue trabajado por Lin y Ying (2007), quienes proponen tres algoritmos, un recocido simulado donde implementan intercambios aleatorios con mejor inserción, un algoritmo genético que usa una búsqueda local para realizar la mutación, y una búsqueda tabú que realiza intercambios aleatorios e inserciones en la lista

tabú. Posteriormente, Anghinolfi y Paolucci (2009) proponen un algoritmo ACO (*Ant Colony Optimization*), donde aportan un nuevo método de actualización de la feromona global. Valente y Alves (2008) proponen un algoritmo basado en el método de *beam search*, donde conservan los procedimientos clásicos de búsqueda y definen un ancho variable para los filtros. Tasgetiren et al. (2009) desarrollaron un algoritmo evolutivo, el cual mejora su eficiencia mediante el empleo de diferentes esquemas de inicialización para la población, basados en algoritmos constructivos como el NEH (*Nawaz, Enscore, and Ham*) y el GRASP (*Greedy Randomized Adaptive Search Procedure*). Kirlik y Oguz (2012) presentan un modelo matemático para la representación del problema, así como el desarrollo de un heurístico de búsqueda por vecindades variables. Más recientemente, Tanaka y Araki (2013) proponen un algoritmo exacto, el cual es resultado de la extensión de otros algoritmos desarrollados por otros autores para problemas de secuenciación, pero estos no consideran tiempos de preparación dependientes de la secuencia. Guo y Tang (2015) proponen un algoritmo de búsqueda dispersa, donde se presenta un heurístico en la etapa constructiva de soluciones y una búsqueda por vecindarios variables en la etapa de mejora.

Algunos otros artículos acerca de problemas de secuenciación de tareas en una máquina y que consideren tiempos de preparación, son: la publicación de Lee y Asllani (2004) en la que se desarrolla un problema con dos objetivos, el de minimizar el número de trabajos atrasados y el *makespan*, intentando resolverlo mediante la comparación de un modelo de programación lineal entera mixta y un algoritmo genético. Gupta y Smith (2006) a su vez, proponen el problema con función objetivo de minimizar la tardanza, a través de la implementación de una búsqueda local y un algoritmo GRASP, además cuentan con un post procesamiento, donde implementan un *Path Relinking* entre tres soluciones vecinas; por otro lado Liao y Juan (2007) tratan el mismo problema y proponen un algoritmo ACO, donde presentan nuevos parámetros de la feromona inicial. Salazar-Hornig y Schriels-Abreu (2014) presentan una estrategia evolutiva para resolver el problema, teniendo como objetivo el de minimizar el *makespan*.

En lo que respecta a publicaciones, donde el problema considere tiempos de preparación dependientes de la secuencia y que además tomen en cuenta al objetivo del caso de estudio en esta tesis, que es el minimizar el tiempo total de flujo, podemos mencionar la investigación de Choobineh et al. (2006), en la que se presenta un problema multi-objetivo, que es minimizar el tiempo total de flujo, el makespan y la tardanza total, quienes proponen un algoritmo de búsqueda tabú; por otro lado Wang (2008) presenta un problema de secuenciación con tiempos de preparación dependientes de la secuencia, además considera, que los tiempos de procesamiento de las tareas están en función de las actividades ya procesadas, el demuestra que este problema puede resolverse a optimalidad para cuatro distintas funciones objetivo: minimizar *makespan*, minimizar el tiempo total de terminación, minimizar el tiempo total de terminación ponderado y la suma cuadrática de tiempos de terminación.

2.4.2 ETAPAS FIJAS DE MANTENIMIENTO PREVENTIVO

Existe un gran número de estudios acerca de problemas de secuenciación que consideran tiempos de no disponibilidad de maquinaria, los cuales son debido a distintos casos particulares, cambios de herramientas, sustituciones de materia prima, mantenimiento, entre otros; en todos ellos implica que la máquina no está disponible para realizar trabajos.

Entre la literatura se encuentran problemas de secuenciación en una máquina con tiempos aleatorios, Pinedo y Rammouz (1988) presentan un problema de secuenciación de n tareas en una máquina, la cual está sujeta a fallas y reparaciones, los tiempos de procesamiento y de disponibilidad de las tareas se consideran aleatorios; y proponen políticas de optimalidad para distintos objetivos. Adiri et al. (1991) asume que los tiempos de no disponibilidad son desconocidos, pero se tiene una distribución de probabilidad para que sean requeridos. Birge et al. (1990) a su vez, presentan algunos ejemplos para problemas de una máquina, donde se busca encontrar el costo esperado, sujeto a averías estocásticas en la maquinaria.

Qi et al. (1999) consideran el problema de secuenciación con muchos intervalos de mantenimiento y un tiempo variable entre estos; en su estudio ellos se enfocan en minimizar el tiempo total de flujo. Por otro lado, Akturk et al. (2003), Akturk et al. (2004), Ecker y Gupta (2005), Chen (2008) y Choi y Kim (2001), consideran paros en el equipo debido a cambios en el herramental, tomando como supuesto que la herramienta tiene un tiempo disponible de vida, y una vez cumplido, esta debe ser sustituida por una nueva.

En algunas líneas de producción los servicios de mantenimiento preventivo son comúnmente programados de forma regular o periódica en el sistema de producción, es por ello que se ha prestado mucho interés por parte de los investigadores a estos casos. Algunas de las publicaciones encontradas, que trabajan un problema de secuenciación de tareas consideradas ininterrumpidas, y la presencia de múltiples periodos de mantenimiento preventivo fijos, son: Liao y Chen (2003) presentan el problema con el objetivo de minimizar la máxima tardanza, donde proponen un *branch and bound* para proveer la solución óptima. Chen (2006) presenta algunos importantes teoremas para el problema con objetivo de minimizar el tiempo total de flujo, para lo cual propone un algoritmo de *branch and bound* y además desarrolla una heurística para tamaños más grandes de instancias. Ji et al. (2007) también trabajaron con la misma estructura del problema a excepción del objetivo, se enfocaron en minimizar el *makespan*. Chen (2009b) se enfoca en resolver el problema con objetivo de minimizar el número de trabajos terminados tarde, para resolverlo propone un heurístico basado en el algoritmo de Moore, además también presenta un algoritmo de *branch and bound* para encontrar soluciones óptimas; Lee y Kim (2012) trabajaron sobre el mismo problema, desarrollando un procedimiento heurístico de dos fases, basado en la modificación de un algoritmo de Moore para el problema de secuenciación de tareas sin servicios de mantenimiento. Ebrahimi Zade y Fakhrzad (2013) proponen una nueva formulación para el problema de minimizar el *makespan*, además presentan un algoritmo genético dinámico, que puede resolver escalas grandes de instancias en un tiempo considerable.

2.4.3 PROBLEMAS DE SECUENCIACIÓN EN UNA MÁQUINA CON TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA Y ETAPAS FIJAS DE MANTENIMIENTO PREVENTIVO

Pese a la importancia mencionada anteriormente, acerca de contemplar factores como lo son los tiempos de preparación entre actividades y un plan de mantenimiento preventivo al momento de realizar la planeación, se han encontrado poco número de investigaciones que los consideren.

Las principales publicaciones encontradas son: Chen (2009a) quien propone una heurística para el problema de secuenciación de n tareas en una máquina con objetivo de minimizar el *makespan*, además establece como supuesto de que las tareas pueden ser interrumpidas, es decir, una vez llegado el tiempo de realizar el mantenimiento esta debe detenerse y terminado el servicio, continuar con la tarea. Posteriormente Ángel-Bello et al. (2011a) y Ángel-Bello et al. (2011b) añaden los supuestos de que las tareas no pueden ser interrumpidas, lo cual aumenta la complejidad del problema; en el primero aportan una modelación matemática y algunas desigualdades válidas, además presenta una estrategia, donde hace uso de las herramientas disponibles en un software comercial; y en la segunda publicación proponen un procedimiento metaheurístico, basado en un GRASP y en una búsqueda tabú. Finalmente, Pacheco et al. (2013) estudiaron el mismo trabajo, y proponen un heurístico híbrido basado en una búsqueda tabú.

2.5 DIAGRAMA DE GANTT

En esta sección se presenta una breve descripción del diagrama de Gantt, el cual va ser utilizado más adelante para la explicación del problema del caso de estudio.

El diagrama de Gantt es una herramienta usual para representar un plan de

actividades que se deben realizar en una o varias máquinas. Sobre el *eje x* se representa el tiempo, y sobre el *eje y*, bloques rectangulares asemejan los trabajos a realizar en las máquinas.

El número de trabajo por lo general se escribe dentro del recuadro. El largo del rectángulo es escalado al tiempo de la máquina y representa el tiempo de procesamiento de la actividad. La Figura 2.1 muestra un ejemplo de un diagrama de Gantt, que representa la planeación en una máquina simple con n actividades por realizar.

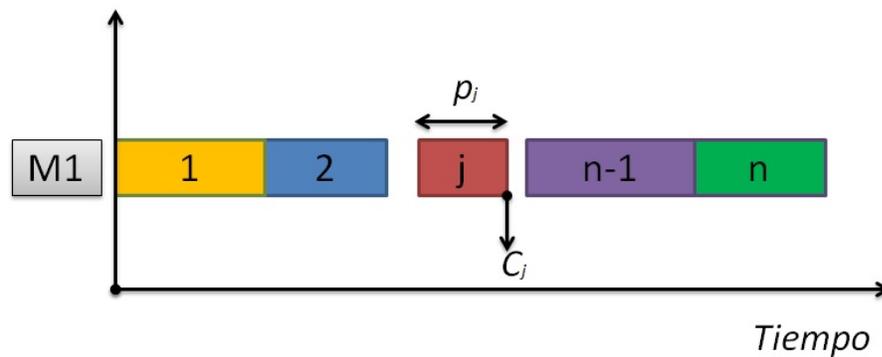


Figura 2.1: Diagrama de Gantt

CAPÍTULO 3

DESCRIPCIÓN DEL PROBLEMA

En este capítulo se explica la estructura, se establecen los supuestos y las distintas propiedades que involucra al problema de secuenciación de tareas en una máquina simple con tiempos de preparación dependientes de la secuencia y etapas fijas de mantenimiento preventivo $(1, h_{1,l} \mid s_{i,j} \mid \sum C_j)$. Se presentan dos modelaciones matemáticas y algunas implementaciones extras, que permiten mejorar el rendimiento de las formulaciones propuestas, de esta manera se aumenta el alcance en el tamaño de instancias a resolver.

3.1 PLANTEAMIENTO DEL PROBLEMA

Se tiene un conjunto de n tareas, las cuales deben ser procesadas en una máquina simple, en la cual su tiempo de operación no es siempre disponible, esto debido a servicios de mantenimiento preventivo que se debe realizar forzosamente en un intervalo de tiempo T ya establecido.

El realizar cada una de las tareas en la máquina conlleva un tiempo de procesamiento, así como también se considera un tiempo de preparación dependiente de la secuencia para realizar dicha tarea, el cual dependerá de la tarea que se realizó previamente.

El objetivo consiste en encontrar una secuencia de tareas la cual minimice el tiempo total de flujo, es decir, minimizar la suma de los tiempos en que se finaliza

cada una de las tareas, y con esto reducir el tiempo que esperan cada uno de los clientes para recibir sus pedidos.

3.1.1 SUPUESTOS

- Las tareas se encuentran disponibles desde el tiempo cero o inicial.
- Las tareas se consideran ininterrumpidas, es decir, una vez empezada una tarea esta debe terminarse antes de realizar la siguiente actividad en la máquina.
- No se consideran restricciones de precedencia, es decir, no se tiene un orden que especifique que una tarea debe realizarse antes o después de otra.
- Durante el horizonte de planeación se presentan etapas fijas de mantenimiento preventivo, las cuales requieren un tiempo de procesamiento para que sean llevadas a cabo.
- Los servicios de mantenimiento preventivo también cuentan con tiempos de preparación dependientes de la secuencia.
- El tiempo de preparación de la transición $i \rightarrow j$ ($s_{i,j}$), es diferente del tiempo de preparación de la transición $j \rightarrow i$ ($s_{j,i}$).
- Se considera que la tarea inicial es precedida por un servicio de mantenimiento preventivo.
- Se define como periodo al tiempo disponible entre el término de un servicio de mantenimiento y el comienzo del siguiente.

3.1.2 NOTACIÓN GENERAL

p_0 Tiempo de procesamiento de un servicio de mantenimiento.

p_j Tiempo de procesamiento de la tarea j .

$s_{i,j}$ Tiempo de preparación dependiente de la secuencia para la tarea j , dado que la tarea i se realizó antes de esta.

$c_{i,j}$ Tiempo total requerido para procesar la tarea j , dado que la tarea i se realizó antes de esta.

T Tiempo que existe entre el término de un servicio de mantenimiento y el término del siguiente servicio de mantenimiento.

3.1.3 EJEMPLO GRÁFICO

Para un fácil entendimiento del problema, en la Figura 3.1 se muestra una representación gráfica en un diagrama de Gantt, en *a)* se presentan cuatro tareas que deben ser procesadas en una máquina cada una con su tiempo de procesamiento asociado, en *b)* se explica la asimetría de los tiempos de preparación dependientes de la secuencia, *c)* muestra las tareas de mantenimiento preventivo a realizar en la máquina y los periodos de tiempo asignados entre estas, y por último *d)* representa una solución factible al problema.

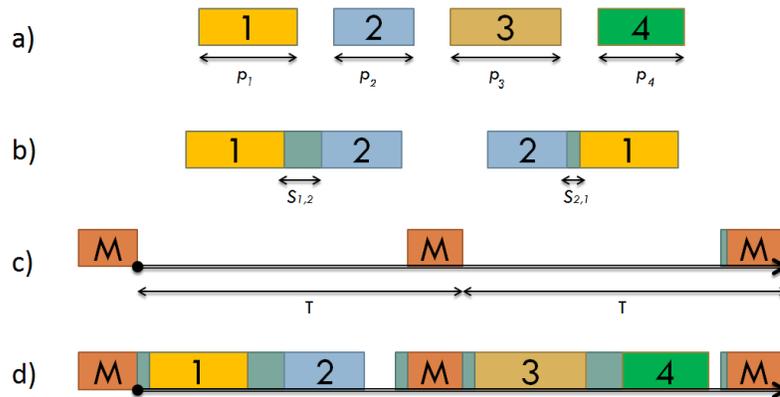


Figura 3.1: Representación gráfica del problema

Es de resaltar que para este tipo de problema se tiene la presencia de tiempos muertos, debido a que los servicios de mantenimiento preventivo son fijos y las tareas se consideran ininterrumpidas.

En la Figura 3.2, se muestra la forma en que se realiza el cálculo de la función objetivo, como se mencionó anteriormente se desea minimizar el tiempo total de flujo, por lo que se quiere minimizar la sumatoria de los tiempos de terminación de cada una de las tareas, todo esto a partir de un punto de origen.

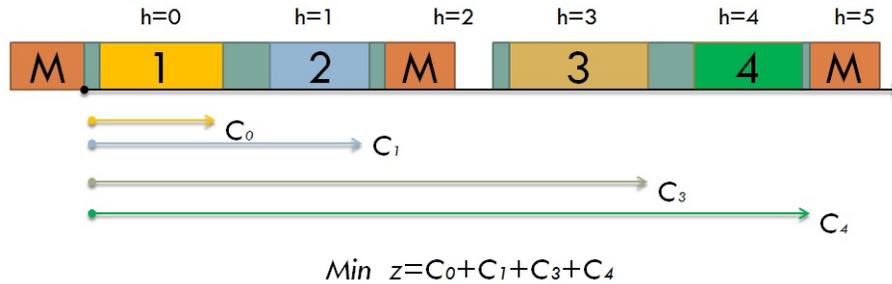


Figura 3.2: Representación gráfica del cálculo de la función objetivo

Para encontrar solución a este problema se proponen dos modelos de programación lineal entera mixta, que se denotan como Modelo A y Modelo B, los cuales están basados en formulaciones propuestas para problemas similares. A continuación se presenta cada uno de los modelos propuestos.

3.2 MODELO A

La primera formulación es una modificación al modelo propuesto por Lee y Kim (2012) para un problema de secuenciación de tareas en una máquina, el cual considera etapas fijas de mantenimiento y tiene como función objetivo el de minimizar el número de tareas entregadas tarde; además se emplea una red multinivel similar a la empleada por Picard y Queyranne (1978) para un problema de mínima latencia.

Como introducción al modelo propuesto en esta tesis, es necesario plantear primero la estructura de la red multinivel. Como se muestra en la Figura 3.3, la red está representada por $2n$ niveles, cada uno constituido por dos conjuntos de nodos, el conjunto $N = \{1, 2, \dots, n\}$, que contiene cada una de las tareas que es necesario secuenciar en la máquina, y el conjunto $Q = \{n+1, \dots, 2n\}$ que contiene n copias

idénticas del servicio de mantenimiento, esto bajo la conclusión que n es una cota superior al número de servicios de mantenimientos requeridos, la cual es alcanzada cuando solo una tarea es procesada en cada uno de los periodos. El nodo 0 mostrado en la imagen corresponde al servicio de mantenimiento precedente a la tarea inicial de la secuencia.

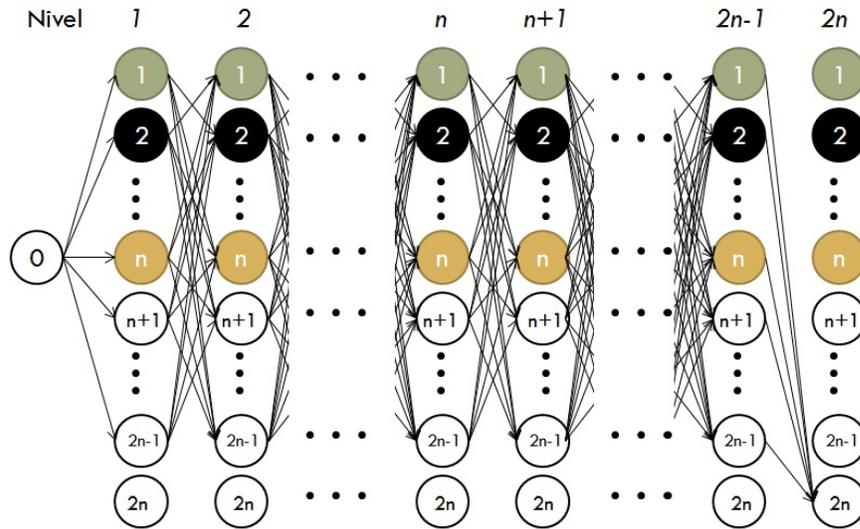


Figura 3.3: Representación del problema mediante una red multinivel

Antes de continuar es necesario plantear el siguiente teorema:

Teorema 3.1 *Considera una secuencia parcial con más de un periodo. Si en el último periodo son asignadas tareas a realizar, y previo a este existe algún otro periodo sin tareas, entonces un mayor tiempo total de flujo es obtenido.*

Prueba 3.1 *Sea una secuencia parcial $S=(M_k, J_p, M_l)$, donde J_p representa la secuencia en el periodo entre los mantenimientos k y l , N_p corresponde al número de tareas en J_p y T_p es el tiempo de flujo de J_p . Ahora $S'=(M_k, M_l, J_p, M_m)$ es obtenida desplazando J_p un periodo adelante en la secuencia. Entonces el tiempo total de flujo de S es:*

$$F(S)=T_p,$$

El tiempo total de flujo en S' es:

$$F(S') = (T * N_p) + T_p$$

Como se considera que N_p es mayor que cero, en conclusión

$$F(S') > F(S)$$

El objetivo en esta modelación consiste en encontrar una ruta partiendo desde el nodo 0, hasta llegar a algún nodo en el nivel $2n$, con la cual se obtenga el mínimo tiempo total de flujo.

Haciendo base en el Teorema 3.1, es posible afirmar que la última actividad en la secuencia corresponde a un servicio de mantenimiento, en caso contrario esto implicaría que se tuvieran periodos intermedio sin ninguna tarea secuenciada, es por ello que en el último nivel de la red, solo existen arcos de entrada a una copia de servicio de mantenimiento.

En esta formulación se hace uso de la siguiente notación.

j índice para las tareas.

i índice para las actividades previas de las tareas.

h índice para el nivel en la secuencia.

p_0 tiempo de procesamiento de un servicio de mantenimiento.

p_j tiempo de procesamiento de la tarea j .

$s_{i,j}$ tiempo de preparación dependiente de la secuencia para la tarea j , dado que la tarea i se realizó antes de esta.

$c_{i,j}$ tiempo total requerido para procesar la tarea j , dado que la tarea i se realizó antes de esta.

T tiempo que existe entre el término de un servicio de mantenimiento y el término del siguiente servicio de mantenimiento.

M es un número muy grande.

Otra cosa que es de mencionar, es que para el fácil manejo de los parámetros se trabajara con el tiempo total de procesamiento ($c_{i,j}$), que es la suma del tiempo de procesamiento de la tarea j (p_j), mas el tiempo de preparación ($s_{i,j}$), dado que la tarea i se realizó antes de j , (Ecuación (3.1)). Así como también se denota como V al conjunto que contiene todas las tareas y servicios de mantenimiento a realizar en la máquina, es decir $V=N \cup Q$.

$$c_{i,j} = p_j + s_{i,j} \quad (3.1)$$

Entre los supuestos se define p_0 como el tiempo de procesamiento del servicio de mantenimiento, puesto que en esta modelación se consideran nodos copias del servicio de mantenimiento estos también tienen un tiempo de procesamiento igual a p_0 (Ecuación (3.2)).

$$p_j = p_0 \quad \forall j \in Q \quad (3.2)$$

Para este modelo se definen las siguientes variables binarias:

$$x_{i,j,h} = \begin{cases} 1 & \text{si la tarea } j \text{ es secuenciada después de la tarea } i, \text{ en el nivel } h, \\ 0 & \text{en otro caso.} \end{cases}$$

$$y_h = \begin{cases} 1 & \text{si en el nivel } h \text{ se realiza un servicio de mantenimiento} \\ 0 & \text{en otro caso.} \end{cases}$$

Además se presentan las siguientes variables auxiliares:

C_h : tiempo de terminación de la tarea secuenciada en el nivel h .

e_h : tiempo transcurrido entre la terminación de la tarea secuenciada en el nivel h y la terminación del mantenimiento previo.

b_h : variable auxiliar para calcular el tiempo requerido de producción y mantenimiento en cada periodo.

g_h : variable auxiliar para calcular el tiempo muerto en cada periodo.

A continuación se plantea el Modelo A

Función objetivo:

$$\min z = \sum_{h \in V} C_h \quad (3.3)$$

Sujeto a:

$$\sum_{h \in V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,h} = 1 \quad \forall i \in \{0\} \cup (V \setminus \{2n\}), \quad (3.4)$$

$$\sum_{h \in V} \sum_{\substack{i \in \{0\} \cup V \\ i \neq j}} x_{i,j,h} = 1 \quad \forall j \in V, \quad (3.5)$$

$$\sum_{i \in \{0\} \cup V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,h} = 1 \quad \forall h \in V, \quad (3.6)$$

$$\sum_{\substack{i \in \{0\} \cup V \\ i \neq j}} x_{i,2n,2n} = 1 \quad (3.7)$$

$$\sum_{j \in V} x_{0,j,1} = 1 \quad (3.8)$$

$$\sum_{j \in V} x_{i,j,h} = \sum_{j \in \{0\} \cup (V \setminus \{2n\})} x_{j,e,i,h-1} \quad \forall h \in V \setminus \{1\}, \forall i \in V \setminus \{2n\}, \quad (3.9)$$

$$e_1 = \sum_{j \in V} x_{0,j,1} * c_{0,j} \quad (3.10)$$

$$e_h \geq e_{h-1} + \sum_{i \in V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,h} * c_{i,j} - M * y_{h-1} \quad \forall h \in V \setminus \{1\}, \quad (3.11)$$

$$e_h \geq \sum_{i \in \{0\} \cup V} \sum_{\substack{j \in V \\ i \neq j}} x_{i,j,h} * c_{i,j} \quad \forall h \in V, \quad (3.12)$$

$$e_h \leq T \quad \forall h \in V, \quad (3.13)$$

$$b_h \geq e_h - M * (1 - y_h) \quad \forall h \in V, \quad (3.14)$$

$$g_h \geq T - b_h - M * (1 - y_h) \quad \forall h \in V, \quad (3.15)$$

$$C_h \geq \sum_{\substack{k \in V \\ k < h}} b_k + \sum_{\substack{k \in V \\ k < h}} g_k - M * y_h \quad \forall h \in V, \quad (3.16)$$

$$\sum_{i \in \{0\} \cup V} \sum_{\substack{j \in Q \\ i \neq j}} x_{i,j,h} - y_h = 0 \quad \forall h \in V, \quad (3.17)$$

$$C_h \geq 0 \quad \forall h \in V, \quad (3.18)$$

$$e_h \geq 0 \quad \forall h \in V, \quad (3.19)$$

$$b_h \geq 0 \quad \forall h \in V, \quad (3.20)$$

$$g_h \geq 0 \quad \forall h \in V, \quad (3.21)$$

$$x_{i,j,h} \in \{0, 1\} \quad \forall i \in \{0\} \cup V, \forall j \in \{0\} \cup V, \forall h \in V, \quad (3.22)$$

$$y_h \in \{0, 1\} \quad \forall h \in V, \quad (3.23)$$

La restricción (3.4) indica que cada actividad debe ser predecesora de otra una vez, esto incluye al nodo cero, pero no toma en cuenta el servicio de mantenimiento $2n$, debido a que se considera como el nodo final en la secuencia. En la restricción (3.5) se expresa que cada tarea y servicio de mantenimiento debe ser realizado solo una vez, es decir, que en cada fila de la red multinivel solo será seleccionado un nodo, mientras que la restricción (3.6) muestra que en cada nivel será asignada solo una tarea o un servicio de mantenimiento. Con estas restricciones se asegura que cada una de las actividades será realizada una vez durante la secuencia.

Como fue planteado en los supuestos, la restricción (3.8) determina el nodo cero como el origen de la secuencia.

Como se mencionó anteriormente, la copia del servicio de mantenimiento $2n$ corresponde al último nodo que debe ser secuenciado, la restricción (3.7), especifica que el servicio $2n$ del nivel $2n$ en la red es el final de la secuencia.

La restricción (3.9) es la considerada como de continuidad de flujo, donde, para cada uno de los nodos intermedios de la red, es decir, sin considerar el origen ni el último nivel, el número de arcos de entrada es igual a la cantidad de arcos de salida.

Las restricciones de la (3.10) a la restricción (3.13), son auxiliares para calcular el valor del tiempo transcurrido entra la terminación de la tarea secuenciada y la terminación del servicio de mantenimiento antes realizado.

En este problema es necesario el conocer el tiempo muerto que se tiene en cada uno de los periodos, la restricción (3.15) permite obtener el valor de dicho tiempo, esto se realiza en base a la restricción (3.14) que calcula el tiempo ocupado por las actividades secuenciadas en cada uno de los periodos.

Como la función objetivo (3.3) busca minimizar la sumatoria de los tiempos de terminación de cada una de las tareas, el cálculo de dichos tiempos es realizado a través de la restricción (3.16).

En la restricción (3.17), se determina la relación de la variables y_h y $x_{i,j,h}$, para las posiciones donde sean asignadas tareas de mantenimiento. Finalmente de la restricción (3.18) a la (3.21) garantizan la no negatividad de las variables auxiliares.

3.2.1 OBTENCIÓN DE DESIGUALDADES VÁLIDAS

En este trabajo se hizo uso del software de optimización ILOG-CPLEX, el cual utiliza un algoritmo de *branch & cut* para resolver problemas de programación entera, en este algoritmo se van generando cotas duales en base a la relajación lineal del modelo planteado, es decir se omiten las restricciones de integralidad de las variables, por este motivo resulta relevante el buscar mejorar dichas cotas.

Si analizamos el Modelo A se puede observar que al resolver la relajación lineal, la función objetivo toma un valor de cero, lo cual se considera una mala cota inferior para el algoritmo.

Lo anterior es debido a que las variables y_1, y_2, \dots, y_{2n} ; al resolver la relajación lineal, estas están libres de tomar valores fraccionarios entre 0 y 1, lo que produce que las tareas de mantenimiento se distribuyan por partes en cada uno de los niveles, por lo tanto en la restricción (3.16), los tiempos de terminación de cada una de las

tareas es cero.

Tratando de mejorar el desempeño del algoritmo de CPLEX en la resolución de este modelo, se hará uso de desigualdades válidas.

En base al Teorema 3.1 podemos concluir distintas cosas, primeramente en el nivel 1, no puede ir asignado un servicio de mantenimiento, por lo tanto debe ser realizada una tarea. Con la implementación de la ecuación (3.24) se plantea esta conclusión, además con ello se garantiza que el valor de la función objetivo de la relajación lineal del modelo, tomara el valor del menor tiempo de transición del servicio de mantenimiento a una tarea.

$$y_1 = 0 \tag{3.24}$$

Por otro lado, para este problema el proceso de ramificación utilizado por el algoritmo de CPLEX, selecciona una variable de naturaleza binaria, la cual, al ser resuelta la relajación lineal haya tomado un valor fraccional; posteriormente, el problema es dividido en dos subproblemas que conservan las mismas restricciones, en uno se agrega la restricción de que dicha variable es igual a cero, mientras que en el otro la variable se iguala a uno. La siguiente desigualdad mejorara el rendimiento del modelo haciendo uso de esta ramificación.

Como se expresa en el Teorema 3.1, una vez secuenciada la última tarea, para la solución óptima del problema de minimizar el tiempo total de flujo, antes de esta no puede existir un periodo entre mantenimientos que no contenga tareas secuenciadas. En base a esto, si en el nivel 1 se asignó de manera obligatoria una tarea, eso quiere decir que el nivel 2 puede o no puede contener un servicio de mantenimiento, si lo contiene, el nivel 3 está obligado a que le sea asignada una tarea, por lo tanto, solo una variable entre y_2 y y_3 puede tomar el valor de 1, la desigualdad (3.25) muestra de manera más general esta característica. Esta propiedad tiene un límite, en la situación de que todas las tareas sean secuenciadas en un mismo periodo, es decir, que en los primeros n niveles no se asignen servicios de mantenimiento, así que en

este caso, y_{n+1} y y_{n+2} ambos pueden tomar el valor de 1.

$$y_h + y_{h+1} \leq 1 \quad \forall h \in N \quad (3.25)$$

La manera en que beneficia esta desigualdad al rendimiento del procedimiento de resolución de este modelo es la siguiente, si en alguna iteración del algoritmo ramifica sobre una variable binaria que asigne un servicio de mantenimiento en el nivel k , en otras palabras $y_k=1$, para un k menor o igual que n , esto provoca que en el nivel $k-1$ y en el $k+1$, no sea asignado ningún servicio de mantenimiento, por lo que se estarán fijando más de una variable durante la ramificación.

Las desigualdades válidas presentadas, fueron agregadas al Modelo A y sometidas a experimentación, como se muestra en el Capítulo 5, dicha formulación obtuvo buenos resultados, mejorando el porcentaje de instancias resultas a optimalidad.

3.3 MODELO B

Esta formulación matemática está basada en asignaciones. Primeramente se planteará como está conformada la estructura, esto con el objetivo de facilitar el entendimiento sobre la modelación del problema.

Para este problema de secuenciación de tareas se mencionó durante la formulación anterior, que n representaba una cota superior del total de los servicios de mantenimiento requeridos para secuenciar todas las tareas, en otras palabras se requieren como máximo n periodos. Si analizamos cada uno de los periodos y los subdividimos en posiciones, definiendo una posición como el lugar que puede ocupar una tarea o un servicio de mantenimiento en la secuencia dentro de un periodo, podemos decir que, en el caso de que todas las tareas sean secuenciadas en un mismo periodo, se han de requerir n posiciones, más una extra para que sea realizado el servicio de mantenimiento.

La Figura 3.4, muestra las definiciones anteriores más claramente, en ella se observa la estructura de todas las posibles posiciones que puede tomar una tarea o un servicio de mantenimiento durante la secuencia. El índice k hace referencia al periodo, que como se puede ver y se dijo anteriormente, puede existir como máximo n periodos. Por otro lado el índice h corresponde a la posición que puede ocupar una actividad dentro de un periodo, y en base a la conclusión antes mencionada cada periodo se subdivide en $n+1$ posiciones.

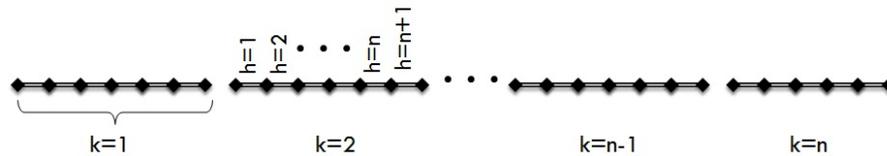


Figura 3.4: Estructura gráfica usada en el Modelo B

De igual manera al Modelo A, se considera un conjunto $N=\{1, 2, \dots, n\}$, que contiene cada una de las tareas que es necesario secuenciar en la máquina, además también se presenta el conjunto $Q=\{n+1\}$, en el cual se encuentran los servicios de mantenimiento, pero a diferencia del modelo anterior, en este caso solo se tiene un único servicio. Para facilitar la notación del modelo se hace uso de un conjunto $O=\{0\}$, el cual representa el origen de la secuencia.

En esta formulación se hace uso de la siguiente notación.

j índice para las tareas.

i índice para las actividades previas de las tareas.

h índice para la posición en que fue secuenciada una tarea en cierto periodo.

k índice para el periodo en que fue secuenciada una tarea.

p_{n+1} tiempo de procesamiento de un servicio de mantenimiento.

p_j tiempo de procesamiento de la tarea j .

$s_{i,j}$ tiempo de preparación dependiente de la secuencia para la tarea j , dado que la tarea i se realizó antes de esta.

$c_{i,j}$ tiempo total requerido para procesar la tarea j , dado que la tarea i se realizó antes de esta.

T tiempo que existe entre el término de un servicio de mantenimiento y el término del siguiente servicio de mantenimiento.

Para este modelo se define la siguiente variable binaria:

$$x_{i,j}^{h,k} = \begin{cases} 1 & \text{si la tarea } j \text{ es secuenciada después de la tarea } i, \text{ en la} \\ & \text{posición } h, \text{ en el periodo } k, \\ 0 & \text{en otro caso.} \end{cases}$$

Además se presenta la siguiente variable auxiliar:

$w_{k,h}$: variable de tiempo acumulado en la posición h , en el periodo k .

A continuación se plantea el Modelo B

Función objetivo:

$$\min z = \sum_{k \in N} \sum_{h \in N} \sum_{\substack{i \in O \cup N \\ i \neq j}} \sum_{\substack{j \in N \\ i \neq j}} k * T * x_{i,j}^{h,k} + \sum_{k \in N} \sum_{h \in N} w_{k,h} \quad (3.26)$$

Sujeto a:

$$\sum_{k \in N} \sum_{h \in N \cup Q} \sum_{\substack{i \in O \cup N \\ i \neq j}} x_{i,j}^{h,k} = 1 \quad \forall j \in N, \quad (3.27)$$

$$\sum_{k \in N} \sum_{h \in N \cup Q} \sum_{\substack{j \in N \cup Q \\ i \neq j}} x_{i,j}^{h,k} = 1 \quad \forall i \in N, \quad (3.28)$$

$$\sum_{i \in O \cup N} \sum_{\substack{j \in N \cup Q \\ i \neq j}} x_{i,j}^{h,k} \leq 1 \quad \forall h \in N \cup Q, \forall k \in N, \quad (3.29)$$

$$\sum_{j \in N \cup Q} x_{0,j}^{1,k} = 1 \quad \forall k \in N, \quad (3.30)$$

$$\sum_{h \in NUQ} \sum_{i \in OUN} x_{i,n+1}^{h,k} = 1 \quad \forall k \in N, \quad (3.31)$$

$$\sum_{h \in NUQ} \sum_{i \in OUN} x_{i,0}^{h,k} = 0 \quad \forall k \in N, \quad (3.32)$$

$$\sum_{h \in NUQ} \sum_{i \in OUN} \sum_{\substack{j \in NUQ \\ i \neq j}} x_{i,j}^{h,k} * c_{i,j} \leq T \quad \forall k \in N, \quad (3.33)$$

$$\sum_{h \in NUQ} \sum_{\substack{j \in NUQ \\ j \neq j_e}} x_{j,j_e}^{h,k} - \sum_{h \in NUQ} \sum_{\substack{i \in OUN \\ i \neq j}} x_{i,j}^{h,k} = 0 \quad \forall k \in N, \forall j \in N, \quad (3.34)$$

$$\sum_{\substack{j \in NUQ \\ i \neq j}} x_{i,j}^{h,k} \leq \sum_{\substack{j \in OUN \\ i \neq j_e}} x_{j_e,i}^{h-1,k} \quad \forall h \in (N \setminus \{1\}) \cup Q, \\ \forall i \in N, \forall k \in N, \quad (3.35)$$

$$\sum_{j \in NUQ} x_{0,j}^{1,k} * c_{0,j} = w_{k,1} \quad \forall k \in N, \quad (3.36)$$

$$w_{k,h+1} \leq T \quad \forall k \in N, \forall h \in N \quad (3.37)$$

$$w_{k,h+1} \geq w_{k,h} + \sum_{j \in N} \sum_{\substack{i \in OUN \\ i \neq j}} x_{i,j}^{h+1,k} * c_{i,j} + T * \left(\sum_{j \in N} \sum_{i \in OUN} x_{i,j}^{h+1,k} - 1 \right) \quad \forall k \in N, \forall h \in N, \quad (3.38)$$

$$w_{k,h+1} \geq 0 \quad \forall h \in N, \forall k \in N \quad (3.39)$$

$$x_{i,j}^{h,k} \in \{0, 1\} \quad \forall i \in \{0\} \cup N, \forall j \in N \cup Q, \forall h \in N, \forall k \in N \quad (3.40)$$

La función objetivo (3.26) de esta formulación está conformada por dos tipos de factores, debido al objetivo del problema, es necesario calcular el tiempo de terminación de cada una de las tareas, para ello, el primer factor corresponde a la suma de tiempo de todos los periodos transcurridos desde el origen hasta el termino del anterior en que fue secuenciada la tarea. El segundo factor representa el tiempo transcurrido desde el inicio del periodo en que fue asignada la tarea, hasta la terminación de la misma. Este cálculo se realiza para cada una de las tareas a secuenciar, y la sumatoria de todos estos elementos corresponde al tiempo total de flujo.

Las ecuaciones (3.27) y (3.28) representan las restricciones de asignación, en las que se especifica que cada una de las tareas se debe de realizar una sola vez en

la secuencia. Con la restricción (3.29), se especifica que a lo más una tarea va ser asignada en cada posición de cada periodo.

Como se mencionó en los supuestos del problema, la secuencia inicia con una tarea que esta después de un servicio de mantenimiento, si se analiza cada uno de los periodos por separados, se puede decir que cada uno de estos toma como origen un servicio de mantenimiento. La restricción (3.30), muestra que la actividad que sea secuenciada en la primera posición en cada periodo esta después de un servicio de mantenimiento. Además, como O representa el conjunto del elemento origen de la secuencia, dicho elemento no puede ir como una actividad a secuenciar, con la restricción (3.32) se evita que suceda esto.

Por otro lado, en cada periodo se debe asignar un servicio de mantenimiento, el conjunto Q tiene el elemento $n + 1$, el cual representa este servicio, en la restricción (3.31), se puede ver que dicho elemento es asignado una vez en cada periodo.

La desigualdad (3.33), es una restricción típica de mochila, donde se menciona que la sumatoria de los tiempos de cada una de las actividades secuenciadas en un periodo, no puede ser mayor al tiempo disponible entre mantenimientos.

La restricción (3.34) estipula que si una tarea es secuenciada en un periodo, esta también será un antecesor a otra actividad en el mismo periodo. Al realizar la asignación de actividades en un periodo, la restricción (3.35) solo permite que estas sean colocadas en las primeras posición del periodo, con el objetivo de que no queden espacios solos en la secuencia.

La variable auxiliar $w_{k,h}$ ayuda a calcular el tiempo de transición desde el inicio del periodo hasta la terminación de la tarea secuenciada en la posición h . De la restricción (3.36) a la (3.37) permiten hacer el cálculo de estos tiempos. Primeramente la restricción (3.36), calcula el tiempo de la tarea secuenciada en la primera posición del periodo, la restricción (3.38) hace el cálculo para el resto de las posiciones y la restricción (3.37) acota superiormente a la variable, donde esta no puede ser mayor al tiempo disponible entre mantenimientos.

Por último se tienen las restricciones (3.39) y (3.40) que especifican la naturaleza de las variables.

3.3.1 MEJORA DE LA FORMULACIÓN POR MEDIO DE LA ELIMINACIÓN DE VARIABLES

En una primera implementación, se emplea un pre-procesamiento que permite la eliminación de variables. Este procedimiento se realiza en base a la estructura en que fue formulado el problema, para esto se presenta un modelo matemático, que plantea la asignación de tareas sobre todas sus posibles posiciones en que pueden ser secuenciadas.

Como se mencionó, una secuencia puede tener a lo más n periodos, y se considera también que cada periodo contiene a lo mas $n + 1$ posiciones, por lo tanto se tienen en total $n^2 + n$ posiciones potenciales para asignar una tarea.

La cantidad de variables binarias $x_{i,j}^{h,k}$ está en función de las $n^2 + n$ posiciones disponibles, por esta razón, este procedimiento se orienta a reducir dicho número.

Si realizamos un análisis en uno de los periodos, tenemos que $n + 1$ es el total de posiciones disponibles para asignar tareas. Pero es posible que ninguna permutación de las n tareas, junto con un servicio de mantenimiento, puedan ser secuenciada en un mismo periodo. Con respecto a esto se presenta un modelo de programación lineal, el cual proporciona una cota superior al número de posiciones necesarias para cada periodo. El modelo consiste en encontrar la máxima cantidad de tareas que pueden ser secuenciadas en un mismo periodo, todo esto sujeto a las restricciones del problema original.

En la formulación matemática se considera un conjunto $N = \{1, 2, 3, \dots, n\}$ que corresponde a todas las tareas potenciales a ser secuenciadas en el periodo. De igual manera al Modelo B, se hace uso del conjunto $O = \{0\}$ que contiene al origen y el conjunto $Q = \{n + 1\}$ que incluye un servicio de mantenimiento.

Los supuestos usados en esta formulación son los mismos empleados por el problema original, solo haciendo referencia sobre un solo periodo.

En esta formulación se hace uso de la siguiente notación.

j índice para las tareas.

i índice para las actividades previas de las tareas.

h índice para la posición en que fue secuenciada una tarea en el periodo.

p_{n+1} tiempo de procesamiento de un servicio de mantenimiento.

p_j tiempo de procesamiento de la tarea j .

$s_{i,j}$ tiempo de preparación dependiente de la secuencia para la tarea j , dado que la tarea i se realizó antes de esta.

$c_{i,j}$ tiempo total requerido para procesar la tarea j , dado que la tarea i se realizó antes de esta.

T tiempo que existe entre el término de un servicio de mantenimiento y el término del siguiente servicio de mantenimiento.

Para esta formulación lineal se define la siguiente variable:

$x_{i,j,h} = \left\{ \begin{array}{l} \text{fracción de la tarea } j \text{ que es secuenciada en la posición } h, \text{ después de la tarea } i \end{array} \right.$

A continuación se plantea la formulación matemática lineal para el pre-procesamiento:

Función objetivo:

$$\max z_h = \sum_{i \in OUN} \sum_{\substack{j \in NUQ \\ i \neq j}} \sum_{h \in NUQ} x_{i,j,h} \quad (3.41)$$

Sujeto a:

$$\sum_{h \in NUQ} \sum_{\substack{i \in OUN \\ i \neq j}} x_{i,j,h} \leq 1 \quad \forall j \in N, \quad (3.42)$$

$$\sum_{h \in NUQ} \sum_{\substack{j \in NUQ \\ i \neq j}} x_{i,j,h} \leq 1 \quad \forall i \in O \cup N, \quad (3.43)$$

$$\sum_{i \in O \cup N} \sum_{\substack{j \in NUQ \\ i \neq j}} x_{i,j,h} \leq 1 \quad \forall h \in N \cup Q, \quad (3.44)$$

$$\sum_{j \in NUQ} x_{0,j,1} = 1 \quad (3.45)$$

$$\sum_{i \in N} \sum_{h \in NUQ} x_{i,n+1,h} = 1, \quad (3.46)$$

$$\sum_{i \in O \cup N} \sum_{h \in NUQ} x_{i,0,h} = 0, \quad (3.47)$$

$$\sum_{h \in NUQ} \sum_{i \in O \cup N} \sum_{\substack{j \in NUQ \\ i \neq j}} x_{i,j,h} * c_{i,j} \leq T, \quad (3.48)$$

$$\sum_{\substack{j \in NUQ \\ i \neq j}} x_{i,j,h} \leq \sum_{\substack{j_e \in O \cup N \\ j_e \neq j}} x_{j_e,i,h-1} \quad \forall i \in N, \forall h \in Q \cup N \setminus \{1\}, \quad (3.49)$$

$$0 \leq x_{i,j,h} \leq 1 \quad \forall i \in O \cup N, \forall j \in N \cup Q, \forall h \in N \cup Q \quad (3.50)$$

En esta formulación las ecuaciones (3.42) y (3.43) son restricciones de asignación, que especifican que una tarea puede ser secuenciada en el periodo y predecesora de otra solo una vez. La restricción (3.44) solo permite que una tarea sea asignada en cada una de las posiciones del periodo.

Para respetar los supuestos del problema, las restricciones (3.45) y (3.46) hacen referencia a que un servicio de mantenimiento es definido como origen y además debe ser secuenciado uno durante el proceso. Por otro lado la restricción (3.47) evita que el elemento de origen sea asignado en alguna otra posición de la secuencia.

La desigualdad (3.48), es una restricción típica de mochila, la cual aborda, que la sumatoria de los tiempos de las tareas asignadas en la secuencia, no debe ser mayor al tiempo disponible entre mantenimientos T .

Al ser este un modelo de programación lineal, el resultado obtenido en la función objetivo (3.41) pertenece a los números reales positivos, además, dado que en el problema original como supuesto se especifica que no pueden ir secuenciadas

tareas incompletas, a dicho valor se le debe aplicar la función piso, de tal forma de mantener la parte entera de la solución (3.51).

$$H_c = \lfloor z_h \rfloor \quad (3.51)$$

A continuación se presenta un ejemplo sobre la eliminación de variables:

Tabla 3.1: Tiempos de procesamiento

Actividad	Tiempo
Mtto.	2
Tarea 1	3
Tarea 2	1
Tarea 3	2
Tarea 4	3
Tarea 5	1

Se tienen 5 tareas a procesar en una máquina simple, cada tarea conlleva un tiempo de procesamiento tal como se muestra en la Tabla 3.1, además se consideran tiempos de preparación dependientes de la secuencia entre las actividades, que se muestran en la Tabla 3.2.

La máquina cuenta con un programa de mantenimiento preventivo, el cual estipula que se le debe de realizar una inspección y servicio de limpieza, forzosamente en intervalos de 15 unidades de tiempo.

Con respecto a las Tablas 3.1 y 3.2, es posible calcular los tiempos totales de procesamiento para cada una de las tareas, los cuales se muestran en la Tabla 3.3.

En base a la estructura del Modelo B, en este ejemplo se considera un $n=5$, por lo cual obtenemos que $i \in \{0,1,2,3,4,5\}$, $j \in \{1,2,3,4,5,6\}$, $h \in \{1,2,3,4,5,6\}$ y $k \in \{1,2,3,4,5\}$, generando un total de 1,080 variables binarias.

Si se implementa el modelo matemático para la eliminación de variables (3.41)-(3.50), se consigue un valor de la función objetivo de $z_h=4.64286$ tareas, que es el

Tabla 3.2: Tiempos de preparación dependiente de la secuencia

Tarea	Mtto.	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5
Mtto.	0	5	3	3	1	1
Tarea 1	5	0	5	6	1	2
Tarea 2	2	1	0	6	3	7
Tarea 3	6	4	3	0	3	2
Tarea 4	1	5	1	1	0	5
Tarea 5	3	2	6	1	3	0

Tabla 3.3: Tiempos totales de procesamiento

Tarea	Mtto.	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5
Mtto.	0	8	4	5	4	2
Tarea 1	7	0	6	8	4	3
Tarea 2	4	4	0	8	6	8
Tarea 3	8	7	4	0	6	3
Tarea 4	3	8	2	3	0	6
Tarea 5	5	5	7	3	6	0

máximo de actividades que pueden ser secuenciadas en un periodo de 15 unidades de tiempo. Puesto que las tareas se consideran ininterrumpidas es necesario aplicar la función piso a este valor (3.51), de forma de mantener la parte entera de la solución, para así determinar que $H_c=4$ es la cantidad máxima de actividades a secuenciar en un periodo.

En conclusión, como no es posible secuenciar más de 4 tareas en un mismo periodo, podemos decir que $h \in \{1,2,3,4\}$, reduciendo a un total de 720 variables binarias. De manera mas general se puede expresar que $h \in \{1,\dots,H_c\}$.

3.3.2 IMPLEMENTACIÓN EXTRA

El algoritmo de *branch & cut* de CPLEX, empleado para resolver problemas de programación lineal entera mixta, genera cotas duales en base a la relajación lineal del modelo planteado, y cuando una solución entera es encontrada en alguna iteración del algoritmo, esta es definida como una cota primal. La solución óptima es obtenida cuando la mejor cota dual y primal son iguales.

El software de CPLEX cuenta con múltiples herramientas que permiten proporcionar información al algoritmo, con el propósito de hacer que las cotas duales y primales converjan de una manera más rápida a la solución óptima.

Cuando se resuelve un modelo de programación entera mixta, se puede proporcionar una solución factible inicial del problema, la cual será empleada como una cota primal en el algoritmo de *branch & cut* y es asignada como la solución incumbente. El contar con una solución incumbente le permite a CPLEX eliminar una porción del espacio de búsqueda.

El procedimiento más usual para la generación de una solución factible para un problema de programación entera, es mediante la implementación de métodos heurísticos, los cuales se limitan a obtener una buena solución no necesariamente la óptima (Martí, 2003). Lógicamente, el tiempo invertido por el método exacto de un problema difícil, es de un orden de magnitud muy superior al del heurístico.

En esta implementación, se hace uso de un método heurístico, para proporcionar una solución factible del problema de secuenciación.

En el algoritmo que se presenta en este subtema, corresponde a un heurístico simple, que es categorizado como un método constructivo, los cuales consisten en construir paso a paso una solución a un problema. Por lo general los métodos constructivos son considerados métodos deterministas y se basan en la mejor selección en cada iteración (Martí, 2003). Con la intención de aumentar la diversidad de las soluciones, se cambia la elección determinista por una elección aleatoria de entre los mejores elementos potenciales, generando así un procedimiento aleatorizado.

DESCRIPCIÓN DEL ALGORITMO: El Algoritmo 1 muestra el procedimiento para la generación de una solución factible, en el cual como paso inicial se ingresan los tiempos de procesamiento de las tareas, los tiempos dependientes de la secuencia y el tiempo disponible entre mantenimientos de la máquina, una vez ingresados, se generan las listas *Restantes* y *Secuencia* (línea 1), la primera contiene todos los índices de las tareas pendientes a ingresar a la solución, inicialmente en esta lista se encuentran las n tareas; la segunda lista presenta la característica de que mantiene el orden en que se van ingresando los elementos a ella y es aquí en donde se va almacenar la información de la solución; bajo el supuesto de que la secuencia inicia con un servicio de mantenimiento, el primer elemento en esta lista es el 0.

Además se declaran otros conjuntos útiles en el transcurso del algoritmo; *LC* que almacenara las tareas que es posible secuenciar en cierta iteración, esto con respecto al tiempo aun disponible en el periodo, que está en función de las tareas ya agregadas a la secuencia; *LRC* es una lista restringida de candidatos, en la que para cada iteración se guardaran las mejores tareas potenciales en base a una función de evaluación descrita más adelante. También se hará uso de variables para realizar cálculos importantes en el algoritmo, *Aux0* calculará el tiempo que ha sido usado en el periodo que se está formando durante la iteración; *Aux1* es un contador que informará la cantidad de periodos completos que han sido secuenciados; y la variable

Objetivo que realizará el cálculo de la función objetivo para la solución generada. Un elemento i informará al algoritmo la última tarea ingresada a la secuencia.

La formación de la solución se desarrolla dentro de un ciclo que tiene como condición de paro el que la lista *Restantes* este vacía (línea 9), es decir, todas las tareas fueron agregadas a la secuencia. Primeramente se agregan a *LC* todas aquellas tareas las cuales su tiempo de transición, en el caso de que estas sean agregadas al final, no sobrepase el tiempo entre mantenimientos (línea 10). Realizado esto pueden suceder dos casos, que la lista *LC* contenga por lo menos un elemento o que esta se encuentre vacía. Para el primero, durante esta iteración será agregara una tarea nueva a la secuencia; mientras que para el segundo, se agregara un servicio de mantenimiento y se procederá a formar un nuevo periodo.

Algoritmo 1 Generador de soluciones

Entrada: Instancia del problema, valor de α .**Salida:** Secuencia de tareas factible y valor de la función objetivo.

- 1: $Restantes = \{1, 2, \dots, n\}$;
 - 2: Se inicializa la *Secuencia* con un servicio de mantenimiento;
 - 3: $LC = \emptyset$ (Lista de candidatos potenciales);
 - 4: $LRC = \emptyset$ (Lista restringida de candidatos);
 - 5: $Aux0 = 0$ (Tiempo acumulado en el periodo);
 - 6: $Aux1 = 0$ (Cantidad de mantenimientos);
 - 7: $i = 0$ (Último elemento en la secuencia);
 - 8: $Objetivo = 0$;
 - 9: **mientras** ($Restantes \neq \emptyset$) **hacer**
 - 10: $LC = LC \cup \{j\} \quad \forall j \in Restantes | Aux0 + c_{i,j} + c_{j,0} \leq T$;
 - 11: **si** ($LC \neq \emptyset$) **entonces**
 - 12: Ordenar LC de menor a mayor, en base a su tiempo de procesamiento $c_{i,j}$;
 - 13: $max_t \leftarrow$ tiempo de transición de la última tarea en LC ;
 - 14: $min_t \leftarrow$ tiempo de transición de la primera tarea en LC ;
 - 15: $LRC = LRC \cup \{j\} \quad \forall j \in LC | c_{i,j} \leq min_t + \alpha * (max_t - min_t)$;
 - 16: $j \leftarrow$ seleccionar un elemento aleatoriamente de LRC ;
 - 17: la tarea j es agregada a la *Secuencia*; $Restantes = Restantes \setminus \{j\}$;
 - 18: $Aux0 = Aux0 + c_{i,j}$; $Objetivo = Objetivo + Aux0 + (Aux1 * T)$;
 - 19: $i \leftarrow j$;
 - 20: $LC = \emptyset$; $LRC = \emptyset$;
 - 21: **si no**
 - 22: $Aux0 = 0$; $Aux1 = Aux1 + 1$;
 - 23: un mantenimiento es agregado a la *Secuencia*; $i = 0$;
 - 24: **fin si**
 - 25: **fin mientras**
-

AGREGAR UNA TAREA NUEVA A LA SECUENCIA: En el primer caso se tiene un conjunto LC , que contiene todas aquellas tareas que aún es posible agregar en el periodo que se está formando durante la iteración. Antes de seleccionar una tarea para ser agregada a la secuencia, se formara un conjunto restringido de candidatos (LRC), en donde α representa el índice de calidad y restringe el conjunto de tareas potenciales a solo aquellas que al ser agregadas a la solución tienen un menor impacto en la función objetivo.

Para la formación del conjunto LRC , cada tarea de LC será evaluada en base a su tiempo total de procesamiento, todo esto basado en la conclusión del siguiente análisis:

La Figura 3.5 muestra una secuencia parcial sobre dos periodos, el periodo 1 se considera completo, mientras que para el periodo 2, la tarea i es la última actividad secuenciada y se encuentra en proceso de formación.

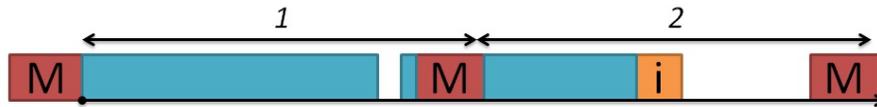


Figura 3.5: Secuencia parcial

En la Figura 3.6 se muestra el conjunto LC , el cual está conformado por solo dos tareas A y B , con tiempos de procesamiento p_A y p_B respectivamente. Cada tarea generara un tiempo total de procesamiento si esta es secuenciada al final, después de la tarea i . En este ejemplo se considera que $c_{i,A}$ es mayor que $c_{i,B}$.

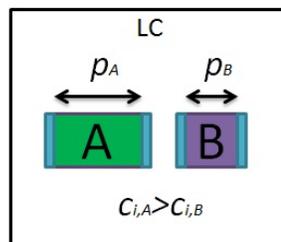


Figura 3.6: Lista de candidatos

Definiremos TTF_P como el tiempo total de flujo de la solución parcial y TU como el tiempo utilizado en el periodo 2.

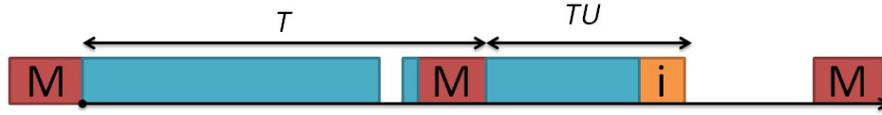


Figura 3.7: Tiempos calculados en solución parcial

Si agregamos primero la tarea A a la secuencia, obtenemos que:

$$TTF_A = TTF_P + T + TU + c_{i,A};$$

Y si agregamos la tarea B , se obtiene que:

$$TTF_B = TTF_P + T + TU + c_{i,B}.$$

En conclusión como $c_{i,A}$ es mayor que $c_{i,B}$, el tiempo total de flujo generado si se agrega la tarea A a la solución, es mayor que el obtenido si se agrega la tarea B .

Para determinar las mejores tareas candidatas, las tareas contenidas en el conjunto LC , serán ordenadas de menor a mayor, esto en base al tiempo total de procesamiento que tomarían si fueran agregadas al final de la secuencia. Por lo tanto, si se define min_t como el menor tiempo de procesamiento de las tareas candidatas, este corresponderá al tiempo de procesamiento $c_{i,j}$ de la primera tarea en LC ; y max_t como el mayor tiempo de procesamiento, este será el tiempo de la última tarea en el conjunto LC . Obtenidos estos valores se calcula un umbral de calidad (línea 15), el cual está en función de un índice α que será descrito más adelante.

Todas aquellas tareas en que su tiempo de total de procesamiento está entre el rango obtenido $[min_t, min_t + \alpha * (max_t - min_t)]$, serán agregadas a LRC . Donde finalmente, una tarea será seleccionada de manera aleatoria de este conjunto, y posteriormente agregada a la secuencia. Antes de avanzar a la siguiente iteración,

es necesario llevar a cabo la actualización de datos y conjuntos auxiliares, así como también quitar la tarea agregada del conjunto *Restantes*.

AGREGAR UN SERVICIO DE MANTENIMIENTO A LA SECUENCIA:

Para el caso en que el conjunto *Restantes* todavía contenga elementos, pero el conjunto *LC* se encuentra vacío, se puede concluir que no posible secuenciar una tarea más en el periodo que se está formando, por lo que es necesario agregar un servicio de mantenimiento a la secuencia, con fin de dar por completo el periodo y el inicio de uno nuevo.

Con el objetivo de generar múltiples soluciones, el Algoritmo 1 es ejecutado varias veces, variando el valor del parámetro de calidad. En el Algoritmo 2 se describe un proceso multi-arranque, para $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ y se resuelve $20n$ veces para cada valor de α .

Algoritmo 2 Procedimiento multi-arranque

Entrada: Parametros iniciales.

Salida: Solución incumbente.

- 1: *Incumbente* = \emptyset ;
 - 2: **para** $\forall \alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ **hacer**
 - 3: **para** $20n$ veces **hacer**
 - 4: Ejecutar **Algoritmo 1**;
 - 5: **si** (*Resultado* es mejor que el *Incumbente*) **entonces**
 - 6: *Incumbente* \leftarrow *Resultado*;
 - 7: **fin si**
 - 8: **fin para**
 - 9: **fin para**
-

El pre-procesamiento para la eliminación de variables presentado, así como el algoritmo constructivo para la obtención de una cota primal al problema, fueron implementados para el Modelo B y sometidos a experimentación. Como se muestra en el Capítulo 5, dichos procedimientos mejoran el rendimiento de la formulación.

En el Capítulo 5, se describe de manera detallada la experimentación realizada para los modelos matemáticos y las metodologías de solución aquí propuestas.

CAPÍTULO 4

PROCEDIMIENTO METAHEURÍSTICO

En la actualidad tanto en la industria como en la ciencia es posible encontrar gran variedad de problemas de optimización, de los cuales muchos pueden ser fácil de resolver, por ejemplo los problemas de programación lineal que pueden ser resueltos por medio del método simplex, mientras que otros son más difíciles de resolver, es decir, no se garantiza que se encuentre la solución óptima en un tiempo razonable (Martí, 2003).

Los procedimientos heurísticos son el resultado del trabajo de investigación realizado para resolver problemas difíciles; en Díaz (1996) los definen como:

Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.

A diferencia de los métodos exactos que encuentran la solución óptima a un problema, con los procedimientos heurísticos no se tiene esta garantía, generalmente obtienen una solución factible, no necesariamente la óptima.

Los procedimientos metaheurísticos, son técnicas superiores a los métodos heurísticos, los cuales tampoco garantizan el encontrar la solución óptima, y al igual

que los heurísticos se basan en el razonamiento lógico para encontrar soluciones, pero con la diferencia que implementan técnicas de búsqueda para salir de óptimos locales.

Los problemas de secuenciación de tareas se consideran difíciles de resolver, es por ello que en este capítulo se presenta un procedimiento metaheurístico, que permite el obtener soluciones factibles buenas, para instancias aun más grandes que las resueltas por los modelos matemáticos. La metodología de solución propuesta está compuesta principalmente por un algoritmo GRASP (Feo y Resende, 1995) y un post procesamiento.

GRASP pertenece a los llamados algoritmos multi-arranque, lo cuales están conformados principalmente por dos fases, una fase constructiva y una fase de mejora. En la primera se construye una solución factible inicial, agregando uno a uno los elementos disponibles durante la iteración, esto basado en una selección aleatoria de entre los mejores elementos disponibles, con respecto a una función de evaluación. En la fase de mejora se suelen emplear sencillos intercambios en la estructura de la solución, con el objetivo de encontrar optimalidad local sin el empleo de mucho tiempo.

A continuación se presenta paso a paso la descripción del método metaheurístico, así como también detalles de inicialización, mejora y un post procesamiento.

4.1 PARÁMETROS INICIALES

Para inicializar el algoritmo se requiere un conjunto de parámetros, de los cuales algunos fueron mencionados y descritos en el capítulo anterior. A continuación se enlistan:

n : número de tareas a procesar.

$c_{i,j}$: tiempo total requerido para procesar la tarea j , dado que la tarea i se realizó antes de esta.

T : tiempo que existe entre el término de un servicio de mantenimiento y el término del siguiente servicio de mantenimiento.

4.2 ESTRUCTURA GENERAL DEL ALGORITMO PROPUESTO

El Algoritmo 3 representa la estructura general de la metodología propuesta, la cual consiste en que una vez ingresados los parámetros iniciales, un procedimiento multi-arranque es inicializado, generando $20n$ soluciones por cada índice α , el cual representa el índice de calidad y se encarga de restringir el conjunto de tareas potenciales a solo aquellas que al ser agregadas a la solución tienen un menor impacto en la función objetivo. Una vez generada una solución, esta es sometida a un proceso de búsqueda local y diversos movimientos, finalmente se corrobora si la solución encontrada resultó mejor que la solución incumbente. Terminado el proceso multiarranque se implementa un post procesamiento, en el que se requiere que sea almacenada una población de soluciones parciales, las cuales serán combinadas para obtener una nueva solución.

4.2.1 MÉTODO CONSTRUCTIVO

Como paso inicial al procedimiento metaheurístico, en esta sección se utiliza el Algoritmo 1 explicado en el capítulo anterior, el cual se emplea para construir soluciones de una manera inteligente.

En un breve resumen, una vez proporcionados los parámetros que conforman la instancia y el valor del índice de calidad α , este algoritmo agrega un elemento a la solución de entre los mejores elementos candidatos disponibles durante cada iteración, hasta completar una solución factible completa.

Una vez completada una solución, el siguiente paso consiste en implementar un

Algoritmo 3 Algoritmo GRASP y Post procesamiento

Entrada: Parametros iniciales.

Salida: Solución incumbente.

```
1: Incumbente =  $\emptyset$ ;  
2: para  $\forall \alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$  hacer  
3:   para  $20n$  veces hacer  
4:     Algoritmo constructivo Algoritmo 1;  
5:     Fase de mejora  
6:       Búsqueda local Algoritmo 4;  
7:       Movimiento 1 Algoritmo 5;  
8:       Movimiento 2 Algoritmo 6;  
9:       Movimiento 3 Algoritmo 7;  
10:      Almacenar soluciones parciales;  
11:      si (NuevaSolución es mejor que el Incumbente) entonces  
12:        Incumbente  $\leftarrow$  NuevaSolución;  
13:      fin si  
14:    fin para  
15:  fin para  
16:  Post procesamiento sobre población de soluciones parciales Algoritmo 8;  
17:  si (NuevaSolución es mejor que el Incumbente) entonces  
18:    Incumbente  $\leftarrow$  NuevaSolución;  
19:  fin si
```

método de búsqueda local y algunos movimientos sobre la estructura de la solución, los cuales buscan acceso a mejores soluciones.

4.2.2 BÚSQUEDA LOCAL

En el Algoritmo 4 se presenta un procedimiento de búsqueda local, el cual consta de realizar intercambios entre todas las tareas que conforman la solución.

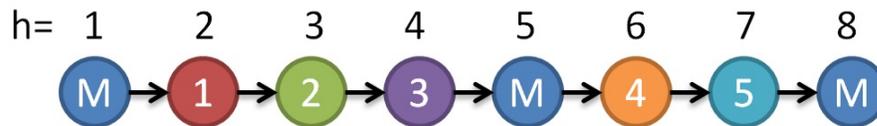


Figura 4.1: Secuencia de una solución factible

Como paso inicial se ingresa una solución factible al algoritmo. A cada tarea o servicio de mantenimiento que conforma dicha solución, se les asocia un índice h , que marcara la posición en que esta secuenciada, tal como se muestra en la Figura 4.1.

Para este algoritmo es necesario el declarar un conjunto P , que contendrá los índices de todas las posiciones en que fue secuenciada una tarea; para el ejemplo de la Figura 4.1, $P = \{2, 3, 4, 6, 7\}$, donde la posición $P_{[2]}$ corresponde a la tarea 1, $P_{[3]}$ a la tarea 2, y así sucesivamente.

Enseguida se procede a realizar intercambios de tareas, entre todas las posiciones que pertenecen al conjunto P (Figura 4.2). Una vez hecho el intercambio, en el caso de que la solución generada sea factible y el valor de su función objetivo sea mejor que el de la solución original, el intercambio se mantiene y se continúa con el siguiente (líneas 7 y 8); en caso contrario se regresan las tareas a su posición anterior (líneas 11 y 14).

Una vez realizados todos los intercambios, se verifica si la solución mejoro en el proceso de búsqueda local, en el caso de que si exista una mejora se repite el algoritmo sobre la nueva solución encontrada, en caso contrario el proceso simplemente termina

Algoritmo 4 Búsqueda local

Entrada: Secuencia factible y valor de la función objetivo.**Salida:** Secuencia factible y valor de la función objetivo.

- 1: P = posiciones ocupadas por la solución;
 - 2: V = valor de la función objetivo de la solución incumbente;
 - 3: V_m = valor de la función objetivo de la solución generada;
 - 4: **mientras** (se encuentre una solución mejor) **hacer**
 - 5: **para** $\forall h \in P$ **hacer**
 - 6: **para** $\forall k \in P | k > h$ **hacer**
 - 7: Realizar intercambio de tareas $P_{[h]} \leftrightarrow P_{[k]}$;
 - 8: **si** (se obtiene una solución factible) **entonces**
 - 9: **si** ($V > V_m$) **entonces**
 - 10: $V = V_m$;
 - 11: **si no**
 - 12: Deshacer intercambio;
 - 13: **fin si**
 - 14: **si no**
 - 15: Deshacer intercambio;
 - 16: **fin si**
 - 17: **fin para**
 - 18: **fin para**
 - 19: **fin mientras**
-

y se avanza al siguiente paso.

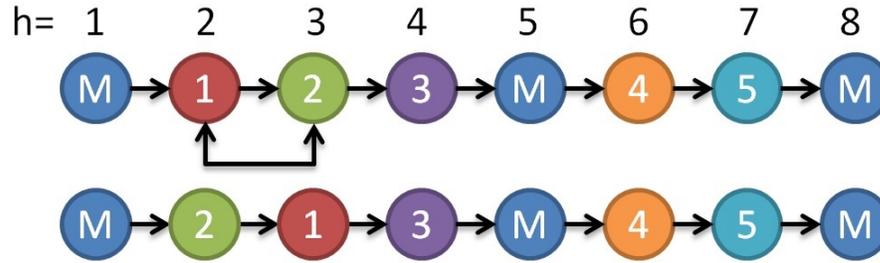


Figura 4.2: Proceso de búsqueda local

En este algoritmo en el peor de los casos se mantendrá la solución generada por el constructor, debido a que deshace los cambios si la solución no mejora o se sale de factibilidad.

4.2.3 MOVIMIENTO 1

En este primer movimiento, se trata de buscar aquella tarea que genera un mayor costo de transición, la cual será retirada de la secuencia y reinsertada en la posición donde genere el menor costo posible.

La Figura 4.3 muestra la manera de calcular el tiempo de transición, el cual está dado por la suma del tiempo de preparación de la tarea en base a la anterior, el tiempo de procesamiento de la tarea y el tiempo de preparación de la tarea siguiente.



Tiempo de transición de $b = s_{a,b} + p_b + s_{b,c}$

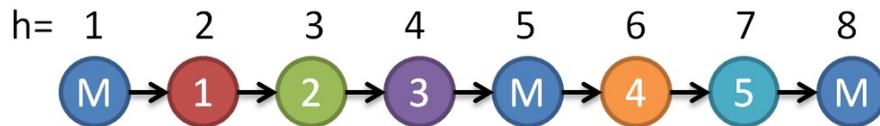
Figura 4.3: Cálculo de costos de transición

Al igual como se muestra en la Figura 4.1, presentada en el apartado de búsqueda local, para este movimiento también se asociara un índice h para cada tarea, que especifica la posición en que fue secuenciada.

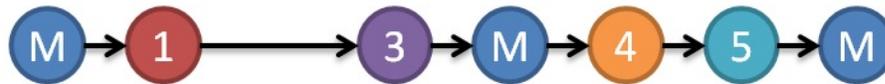
En el Algoritmo 5 se muestra que como paso inicial, es necesario hacer un recorrido por cada una de las posiciones de la solución, para localizar aquella tarea que genera el mayor tiempo de transición. Una vez identificada esta, es eliminada de la solución y es colocada en la posición donde genere el menor costo de transición posible.

Ejemplo:

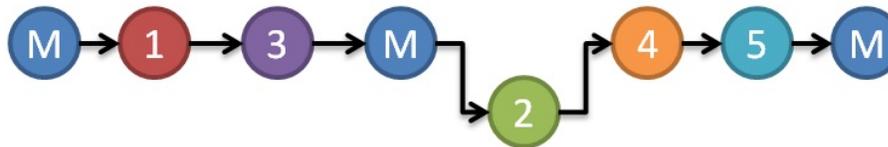
Considere que se tiene una secuencia de 5 tareas tal como se muestra a continuación.



Supongamos que son calculados los tiempos de transición y la tarea 2 es la que genera un mayor costo de tiempo, entonces esta será retirada de la secuencia.



Hasta el momento se tiene una solución infactible por el supuesto de que deben ser realizadas todas las tareas. La posición en que la tarea 2 se colocara es aquella en que genere el menor costo de transición.



Realizado el movimiento pueden suceder dos cosas, que la nueva solución sea factible o infactible; en el primer caso se implementara la búsqueda local representada en el Algoritmo 4, si la solución encontrada al final resulta que es mejor que la incumbente, el movimiento se mantiene y se actualiza la información, de lo contrario se deshace el movimiento; para el caso de que la solución resulte infactible simplemente se regresa la tarea a su posición inicial.

Algoritmo 5 Movimiento 1

Entrada: Secuencia factible y valor de la función objetivo.

Salida: Secuencia factible y valor de la función objetivo.

- 1: P = posiciones ocupadas por la solución;
 - 2: V = valor de la función objetivo de la solución incumbente;
 - 3: V_m = valor de la función objetivo de la solución generada;
 - 4: **para** $\forall h \in P$ **hacer**
 - 5: Calcular tiempo de transición $tt_h = s_{P_{[h-1]}, P_{[h]}} + p_{P_{[h]}} + s_{P_{[h]}, P_{[h+1]}}$
 - 6: **fin para**
 - 7: Identificar la tarea que genera mayor tiempo de transición;
 - 8: Borrar la tarea de mayor transición de la secuencia;
 - 9: Insertar la tarea faltante en la posición donde genere el menor tiempo de transición;
 - 10: **si** (se obtiene una solución factible) **entonces**
 - 11: Implementar el **Algoritmo 4** de búsqueda local;
 - 12: **si** ($V > V_m$) **entonces**
 - 13: $V = V_m$;
 - 14: **si no**
 - 15: Deshacer el movimiento;
 - 16: **fin si**
 - 17: **si no**
 - 18: Deshacer movimiento;
 - 19: **fin si**
-

4.2.4 MOVIMIENTO 2

Como se puede apreciar en el Algoritmo 6, la única diferencia entre este movimiento y el Movimiento 1, es que este movimiento no se enfoca en seleccionar la tarea que genera un mayor costo de transición, si no el de llevar a cabo una selección aleatoria de alguna tarea en la secuencia, la cual será removida y reinsertada en la posición donde genere un menor tiempo de transición.

Algoritmo 6 Movimiento 2

Entrada: Secuencia factible y valor de la función objetivo.

Salida: Secuencia factible y valor de la función objetivo.

- 1: V =valor de la función objetivo de la solución incumbente;
 - 2: V_m =valor de la función objetivo de la solución generada;
 - 3: **para** (n veces) **hacer**
 - 4: Seleccionar una tarea aleatoria y remover de la secuencia;
 - 5: Insertar la tarea faltante en la posición donde genere el menor tiempo de transición;
 - 6: **si** (se obtiene una solución factible) **entonces**
 - 7: Implementar el **Algoritmo 4** de búsqueda local;
 - 8: **si** ($V > V_m$) **entonces**
 - 9: $V = V_m$;
 - 10: **si no**
 - 11: Deshacer el movimiento;
 - 12: **fin si**
 - 13: **si no**
 - 14: Deshacer movimiento;
 - 15: **fin si**
 - 16: **fin para**
-

Este procedimiento puede ser considerado semiciego, debido al modo de selección de la tarea; y es por esta característica de aleatoriedad que es posible ejecutar el algoritmo varias veces y poder llegar a obtener soluciones diferentes.

4.2.5 MOVIMIENTO 3

El Algoritmo 7 muestra un proceso de selección e inserción totalmente ciego, en este movimiento una tarea es seleccionada aleatoriamente de entre la secuencia, para posteriormente ser removida, y colocada en alguna otra posición aleatoria.

Del mismo modo que el Movimiento 2, se presenta un ciclo para llevar a cabo varias veces la ejecución del algoritmo, y nos permita de esta manera obtener soluciones diferentes.

Algoritmo 7 Movimiento 3

Entrada: Secuencia factible y valor de la función objetivo.

Salida: Secuencia factible y valor de la función objetivo.

- 1: V =valor de la función objetivo de la solución incumbente;
 - 2: V_m =valor de la función objetivo de la solución generada;
 - 3: **para** (n veces) **hacer**
 - 4: Seleccionar una tarea aleatoria y remover de la secuencia;
 - 5: Insertar la tarea faltante en una posición aleatoria en la secuencia;
 - 6: **si** (se obtiene una solución factible) **entonces**
 - 7: Implementar el **Algoritmo 4** de búsqueda local;
 - 8: **si** ($V > V_m$) **entonces**
 - 9: $V = V_m$;
 - 10: **si no**
 - 11: Deshacer el movimiento;
 - 12: **fin si**
 - 13: **si no**
 - 14: Deshacer movimiento;
 - 15: **fin si**
 - 16: **fin para**
-

4.2.6 POST PROCESAMIENTO

Terminado el proceso de construcción y de mejora de soluciones, se implementa un nuevo método constructivo, el cual hace uso de una población de soluciones parciales.

Como introducción a esta metodología, definiremos una solución parcial como la secuencia de tareas que existe en un mismo periodo, esto es que comienza y termina con un servicio de mantenimiento. Como ejemplo en la Figura 4.4 inciso a) se muestra una solución completa, y en el inciso b) se representa la misma solución subdividida en sus soluciones parciales.

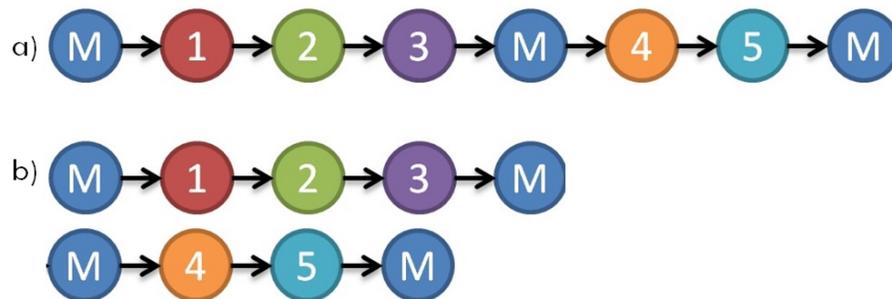


Figura 4.4: Representación de soluciones parciales

En este procedimiento se hace uso del siguiente teorema, el cual fue propuesto por Chen (2006) para un problema de secuenciación de tareas en una máquina simple, considerando etapas fijas de mantenimiento preventivo, con el objetivo de minimizar el tiempo total de flujo.

Teorema 4.1 *Considere dos periodos, ambos compuestos por diferente número de trabajos. Si el periodo con el menor número de trabajos es secuenciado después, entonces un tiempo total de flujo más pequeño es obtenido.*

Prueba 4.1 *Sea una secuencia parcial $S=(M_k, J_p, M_l, J_q, J_r)$, donde J_p está en el periodo B_i , y J_q está en el periodo B_j . Sea S' generada por el intercambio en S de*

B_i con B_j . Ahora, se obtiene $S'=(M_k, J_q, J_r, M_l, J_p)$. Entonces el tiempo total de flujo en S es:

$$F(S)=CM_k+p_p+CM_l+p_q+CM_l+p_q+p_r$$

El tiempo total de flujo en S' es:

$$F(S')=CM_k+p_q+CM_k+p_q+p_r+CM_l+p_p$$

Es claro que CM_k es más pequeño que CM_l . Así que $F(S')<F(S)$.

En el Algoritmo 3 se especifica que cada solución generada en el GRASP, será subdividida y almacenada, para de esta forma construir una población de soluciones parciales (línea 9), la cual será utilizada como dato de entrada para el post procesamiento.

El Algoritmo 8 corresponde al post procesamiento, el cual inicia una vez que es proporcionada una población de soluciones parciales, y es asignada como incumbente la mejor solución obtenida por el GRASP (línea 1). La formación de una nueva solución se lleva a cabo dentro de un ciclo iterativo, el cual realiza como paso inicial, la búsqueda de la solución parcial dentro de la población que contenga mayor número de tareas secuenciadas, dicha cantidad de tareas es definida como la cardinalidad de la secuencia (línea 6). Una vez localizada la máxima cardinalidad, se crea un subconjunto que incluya todas las soluciones parciales que contengan la misma cantidad de tareas secuenciadas que esta (línea 7), posteriormente se selecciona una solución parcial de manera aleatoria de este subconjunto, la cual es agregada a la secuencia de la *NuevaSolución* (línea 8 y 9). Finalmente, se elimina de la población toda solución parcial que contenga algún elemento en común con la *NuevaSolución* (línea 10), y se repite nuevamente el proceso. Esta metodología de construcción termina una vez que la población de soluciones parciales se encuentre vacía.

Realizado este proceso pueden suceder dos cosas, que la *NuevaSolución* sea una solución completa o que esta sea una solución infactible, esto debido a la falta de alguna tarea en la secuencia (línea 12). En el caso de infactibilidad, es necesario

Algoritmo 8 Algoritmo Post procesamiento

Entrada: Población de soluciones parciales.**Salida:** Solución incumbente.

- 1: *Incumbente* \leftarrow Mejor solución obtenida por el GRASP;
 - 2: **para** $10n$ veces **hacer**
 - 3: Reiniciar población de soluciones parciales;
 - 4: *NuevaSolución* $= \emptyset$;
 - 5: **mientras** Población contenga soluciones parciales **hacer**
 - 6: $MC \leftarrow$ Mayor cardinalidad entre las soluciones parciales en la población;
 - 7: $Sub \leftarrow$ Subconjunto de la población, que contiene las soluciones parciales de cardinalidad MC ;
 - 8: $D \leftarrow$ Seleccionar una solución parcial aleatoriamente de Sub ;
 - 9: Agregar D a *NuevaSolución*;
 - 10: Eliminar de la población toda solución que contenga un elemento de D ;
 - 11: **fin mientras**
 - 12: **si** (*NuevaSolución* es infactible) **entonces**
 - 13: Implementar **Algoritmo 9** con $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ a partir de *NuevaSolución*;
 - 14: **fin si**
 - 15: Implementar el **Algoritmo 4** de búsqueda local;
 - 16: **si** (*NuevaSolución* es mejor que el *Incumbente*) **entonces**
 - 17: $Incumbente \leftarrow NuevaSolución$;
 - 18: **fin si**
 - 19: **fin para**
-

la implementación del Algoritmo 9, el cual se encarga de complementar la solución infactible agregando una tarea de las faltantes en cada paso, el procedimiento utilizado en este algoritmo es similar al utilizado en el Algoritmo 1, en cual se forma una lista restringida de candidatos de entre las mejores opciones posibles, y se selecciona un elemento aleatorio de esta para ser agregado a la secuencia. Una vez obtenida una solución completa se realiza un procedimiento de búsqueda local, el mismo que es utilizado en el GRASP en la etapa de mejora. Por último se verifica si la solución obtenida es mejor que el incumbente, en caso afirmativo, la *NuevaSolución* es asignada como incumbente.

El algoritmo propuesto en esta etapa, tiene un criterio de selección aleatoria, por lo cual se propone la generación de múltiples soluciones dentro de un ciclo iterativo, el cual especifica que el algoritmo proporcionara $10n$ nuevas soluciones.

En el Capítulo5, se describe de manera detallada la experimentación realizada para el método metaheurístico aquí propuesto; y se realiza la comparación contra las mejores soluciones conocidas hasta el momento.

Algoritmo 9 Complementar soluciones parciales

Entrada: Instancia del problema, Secuencia parcial, valor de α .**Salida:** Secuencia de tareas factible y valor de la función objetivo.

- 1: $Restantes \leftarrow$ Tareas pendientes en la secuencia parcial;
 - 2: Se inicializa la *Secuencia* con la secuencia parcial;
 - 3: $LC = \emptyset$ (Lista de candidatos potenciales);
 - 4: $LRC = \emptyset$ (Lista restringida de candidatos);
 - 5: $Aux0$ (Tiempo utilizado en el último periodo de la secuencia parcial);
 - 6: $Aux1$ (Cantidad de periodos completos en la secuencia parcial);
 - 7: i (Última tarea en la secuencia parcial);
 - 8: $Objetivo \leftarrow$ Calcular función objetivo de la secuencia parcial;
 - 9: **mientras** ($Restantes \neq \emptyset$) **hacer**
 - 10: $LC = LC \cup \{j\} \quad \forall j \in Restantes | Aux0 + c_{i,j} + c_{j,0} \leq T$;
 - 11: **si** ($LC \neq \emptyset$) **entonces**
 - 12: Ordenar LC de menor a mayor, en base a su tiempo de procesamiento $c_{i,j}$;
 - 13: $max_t \leftarrow$ tiempo de transición de la última tarea en LC ;
 - 14: $min_t \leftarrow$ tiempo de transición de la primera tarea en LC ;
 - 15: $LRC = LRC \cup \{j\} \quad \forall j \in LC | c_{i,j} \leq min_t + \alpha * (max_t - min_t)$;
 - 16: $j \leftarrow$ seleccionar un elemento aleatoriamente de LRC ;
 - 17: la tarea j es agregada a la *Secuencia*; $Restantes = Restantes \setminus \{j\}$;
 - 18: $Aux0 = Aux0 + c_{i,j}$; $Objetivo = Objetivo + Aux0 + (Aux1 * T)$;
 - 19: $i \leftarrow j$;
 - 20: $LC = \emptyset$; $LRC = \emptyset$;
 - 21: **si no**
 - 22: $Aux0 = 0$; $Aux1 = Aux1 + 1$;
 - 23: un mantenimiento es agregado a la *Secuencia*; $i = 0$;
 - 24: **fin si**
 - 25: **fin mientras**
-

CAPÍTULO 5

EXPERIMENTACIÓN COMPUTACIONAL

En este capítulo se presentan los resultados obtenidos al realizar la experimentación de las diferentes metodologías planteadas en los capítulos 3 y 4. Dicha experimentación fue programada en lenguaje c++ y desarrollada en Visual Studio 2012, se hizo uso de una computadora con Procesador Intel ®Core™i7 CPU @ 3.40Ghz; RAM 8.00GB, Windows 7 Professional.

5.1 INSTANCIAS

Basado en la estructura del problema propuesto, y bajo el conocimiento que no existen muchos casos de investigación acerca de problemas de secuenciación que presenten las características y supuestos aquí planteados, se hará uso de las instancias desarrolladas por Ángel-Bello et al. (2011a), quienes presentan siete tamaños diferentes de instancias: 10, 12, 15, 20, 30, 40 y 50 tareas. Los valores de los $c_{i,j}$ fueron generados usando una distribución uniforme en tres distintos intervalos. Las instancias desde Prx01 a Prx05 fueron generadas en un intervalo de (2,8); las instancias de Prx06 a Prx10 en un intervalo de (4,12) y las instancias de Prx11 a la Prx15 en un intervalo de (5,20).

Para determinar el tiempo entre mantenimientos T , se aplica un proceso similar al propuesto por Li et al. (1992) para determinar una restricción de distancia en un

problema de ruteo de vehículos. Para este proceso se calcula un valor dm , el cual es obtenido como $dm = \max_i \{(c_{0,i} + c_{i,0})/2\}$. Para asegurar la factibilidad, la restricción de tiempos entre mantenimientos se genera en función del dm ($T \geq 2dm$).

Con el objetivo de tener diferentes niveles de estrechez de tiempos entre mantenimientos, T tomara los valores: $2.25dm$, $2.5dm$, $3dm$ y $4dm$. Como resultado se obtienen cuatro instancias por cada una de las instancias originales. Se probaron un total de 720 instancias.

5.2 EXPERIMENTACIÓN DE LOS MODELOS MATEMÁTICOS

Las formulaciones fueron implementadas en lenguaje c++ haciendo uso de las herramientas proporcionadas por CPLEX 12.5, ILOG Concert Technology. Durante la ejecución se estableció un tiempo límite de 7200 segundos, superado este tiempo el algoritmo de solución del software es interrumpido y se denomina como instancia no resuelta.

En la Tabla 5.1 se agruparon las instancias de acuerdo al número de tareas, y se muestran los porcentajes de instancias que fueron resueltas a optimalidad para cada una de las formulaciones; la columna uno corresponde al tamaño de las instancias, la columna dos indica la cantidad total de instancias para cada diferente conjunto, la columna tres representa el porcentaje de tareas resueltas a optimalidad por el Modelo A, la columna cuatro (Mod.A+Imp.) es el porcentaje resuelto por el Modelo A adicionando las desigualdades válidas para el mejoramiento de la formulación, la columna cinco muestra el porcentaje de instancias resueltas por al Modelo B y la columna seis representa el Modelo B con pre-procesamiento para la eliminación de variables (Mod.B+Imp.) y además se hace uso de la solución encontrada por el método heurístico, asignándola como solución incumbente inicial.

Tabla 5.1: Porcentaje de instancias resueltas

Eficiencia de los modelos matemáticos					
n	Cant.	Modelo A	Mod.A+Imp.	Modelo B	Mod.B+Imp.
10	120	100 %	100 %	100 %	100 %
12	120	99.2 %	100 %	100 %	100 %
15	120	0.8 %	96.7 %	97.5 %	100 %
20	120	0 %	4.2 %	1.7 %	65 %
30	60	0 %	0 %	0 %	0 %
40	60	0 %	0 %	0 %	0 %
50	120	0 %	0 %	0 %	0 %

En base a la tabla anterior se puede observar que los distintas metodologías empleadas, fueron capaces de resolver a optimalidad el conjunto de instancias con 10 tareas a secuenciar; el Modelo B haciendo uso de procedimientos de mejora (pre-procesamiento y proporcionar una solución factible inicial), resultó ser el método con mayor rendimiento, encontrando la solución óptima en todas las instancias con 10, 12 y 15 tareas a secuenciar, y además, se pudo resolver un 65 % de las instancias con 20 tareas.

Pese a la implementación de mejoras en ambas formulaciones, ninguna fue capaz de resolver instancias mayores a 20 tareas dentro de los tiempos límites establecidos.

Las Tablas 5.2 y 5.3 consisten en el análisis de tiempos de ejecución, tomando en cuenta solamente las instancias que fueron resueltas a optimalidad para las distintas metodologías. La primera columna muestra el tamaño de las instancias; la columna dos contiene el tiempo promedio de ejecución y la columna tres la desviación estándar de resolver simplemente el modelo matemático; de manera similar, la columna cuatro y cinco contienen el tiempo promedio y la desviación estándar de resolver el modelo matemático haciendo uso de diversas implementaciones de mejora.

Tabla 5.2: Tiempos de ejecución del Modelo A

n	Modelo A		Modelo A + D.V.	
	Tiempo Promedio	Desviación Estandar	Tiempo Promedio	Desviación Estandar
10	121.84	± 58.61	44.75	± 16.72
12	1062.12	± 998.12	206.58	± 104.68
15	5647.33	± 0	1863.04	± 1420.14
20	-	-	5133.15	± 1244.35

Tabla 5.3: Tiempos de ejecución del Modelo B

n	Modelo B		Modelo B + Imp.	
	Tiempo Promedio	Desviación Estandar	Tiempo Promedio	Desviación Estandar
10	13.01	± 8.86	3.53	± 2.71
12	82.46	± 131.26	13.42	± 9.56
15	1421.4	± 1213.16	174.71	± 507.33
20	5108.82	± 2625.28	1882.35	± 1834.78

Mediante lo presentado en las tablas se puede concluir que el Modelo B, tanto con implementaciones de mejora como sin ellas, presenta un mejor rendimiento que el Modelo A, esto es debido a como fue mencionado en capítulos anteriores, el algoritmo de *branch & cut* utilizado por CPLEX va generando cotas duales en base a la relación lineal del modelo planteado, en este caso para el Modelo A sin el uso de desigualdades válidas, la relajación lineal inicial siempre es 0, lo cual se considera una muy mala cota inferior, esto provoca que el algoritmo converja de una manera más lenta a la solución óptima.

En el apéndice A se muestran los resultados individuales para cada una de las instancias.

5.3 EXPERIMENTACIÓN DEL PROCEDIMIENTO METAHEURÍSTICO

El procedimiento metaheurístico fue implementado en lenguaje c++ y desarrollado en Visual Studio 2012.

Debido a que durante la implementación de los modelos matemáticos no todas las instancias se resolvieron a optimalidad, la experimentación se realizará de dos maneras, la primera consiste en la comparación entre el valor de la solución incumbente obtenida por el procedimiento metaheurístico y el valor objetivo de la solución óptima; mientras que en aquellos casos en que se desconoce el óptimo, se realizará una comparación con el algoritmo del vecino más cercano. Se realizaron 10 réplicas del algoritmo para cada una de las instancias

5.3.1 COMPARACIÓN CONTRA LA SOLUCIÓN ÓPTIMA

En la Tabla 5.4 se muestran los resultados generales de la experimentación del método metaheurístico en comparación con los resultados óptimos conocidos. La primera columna indica el tamaño de las instancias, la columna dos muestra la

cantidad de instancias en que se conoce la solución óptima, es de mencionarse que en las instancias con 20 tareas, solo en 82 se conoce la solución óptima; la columna tres muestra el porcentaje de instancias en que el algoritmo propuesto encontró la solución óptima, la columna cuatro corresponde al coeficiente de variabilidad para la función objetivo, obtenido de realizar 10 réplicas del algoritmo para cada instancia; finalmente la columna cinco contiene el GAP promedio y la columna cinco el tiempo promedio de ejecución.

Tabla 5.4: Experimentación 1 del método metaheurístico

n	Cant.	Porcentaje de optimalidad	Coficiente de variabilidad	GAP promedio	Tiempo promedio seg.
10	120	97.5 %	0.19 %	0.03 %	2.686
12	120	91.6 %	0.29 %	0.31 %	5.189
15	120	82.5 %	0.31 %	0.35 %	12.211
20	82	30.5 %	0.63 %	0.35 %	15.216

Los GAP presentados en la tabla corresponden a los promedios de los GAP´s obtenidos por la metaheurística y son calculados como:

$$GAP = 100 * \frac{SoluciónIncumbente - SoluciónÓptima}{SoluciónÓptima} \quad (5.1)$$

donde la solución incumbente corresponde a la mejor solución obtenida por el procedimiento metaheurístico.

Como se puede observar en los resultados, era de esperarse que los tiempos de procesamiento fueran mucho menores para el proceso metaheurístico en comparación con los reportados por los modelos matemáticos, además a pesar de que la metodología propuesta no alcanza en su totalidad a encontrar las soluciones óptimas, esta presenta un GAP cercano a cero.

En el apéndice B se muestran los resultados individuales para cada una de las instancias.

5.3.2 COMPARACIÓN CONTRA EL VECINO MÁS CERCANO

Antes de continuar se presenta el algoritmo del vecino más cercano, el cual se considera en un método totalmente ciego, como se puede ver en el Algoritmo 10 este procedimiento consiste en ir formando una solución paso a paso, en el cual durante cada iteración es agregada a la secuencia aquella tarea que genera el menor costo en la función objetivo, es decir se agrega la tarea que tenga el menor tiempo total de procesamiento $(c_{i,j})$ en base a la última tarea en la secuencia.

Algoritmo 10 Algoritmo del vecino más cercano

Entrada: Instancia del problema.

Salida: Secuencia de tareas factible y valor de la función objetivo.

$Restantes = \{1, 2, \dots, n\};$

Se inicializa la *Secuencia* con un servicio de mantenimiento;

$LC = \emptyset$ (Lista de candidatos potenciales);

$Aux0 = 0$ (Tiempo acumulado en el periodo);

$Aux1 = 0$ (Cantidad de mantenimientos);

$i = 0$ (Último elemento en la secuencia);

$Objetivo = 0;$

mientras ($Restantes \neq \emptyset$) **hacer**

$LC = LC \cup \{j\} \quad \forall j \in Restantes \mid Aux0 + c_{i,j} + c_{j,0} \leq T;$

si ($LC \neq \emptyset$) **entonces**

$j \leftarrow$ tarea con menor tiempo total de procesamiento $c_{i,j}, \forall j \in LC;$

la tarea j es agregada a la *Secuencia*; $Restantes = Restantes \setminus \{j\};$

$Aux0 = Aux0 + c_{i,j};$ $Objetivo = Objetivo + Aux0 + (Aux1 * T);$

$i \leftarrow j;$

$LC = \emptyset;$

si no

$Aux0 = 0; Aux1 = Aux1 + 1;$

un mantenimiento es agregado a la *Secuencia*; $i = 0;$

fin si

fin mientras

En la Tabla 5.5 se muestran los resultados para la metaheurística propuesta en aquellas instancias en que los modelos matemáticos no lograron encontrar la solución óptima, para esto se realiza la comparación con el algoritmo del vecino más cercano. La primera columna indica el tamaño de las instancias, la columna dos muestra la cantidad de instancias en que se desconoce la solución óptima, la columna tres corresponde al coeficiente de variabilidad para la función objetivo, obtenido de realizar 10 réplicas del algoritmo para cada instancia; la columna cuatro contiene el GAP promedio y la cinco representa los tiempos promedio de ejecución.

Los GAP presentados en esta tabla corresponden a los promedios de los GAP 's obtenidos por la metaheurística y son calculados como:

$$GAP = 100 * \frac{SoluciónIncumbente - SoluciónVC}{SoluciónVC} \quad (5.2)$$

donde la *SoluciónVC* corresponde a la solución obtenida por el algoritmo del vecino más cercano.

Tabla 5.5: Experimentación 2 del método metaheurístico

n	Cant.	Coeficiente de variabilidad	GAP promedio	Tiempo promedio seg.
20	38	0.63 %	-10.73 %	17.467
30	60	0.44 %	-11.54 %	38.797
40	60	0.36 %	-11.16 %	73.970
50	120	1.03 %	-28.10 %	154.394

Si tomamos en cuenta, el que para estas instancias el tratar de resolverlas por medio de un método exacto, como lo es una formulación matemática usando un *software* de optimización conlleva más de 7200 seg., que era el tiempo límite de ejecución establecido, es más que claro que el tiempo de computo de este algoritmo es de una magnitud mucho menor. La calidad de las soluciones en comparación con el algoritmo del vecino más cercano, era de esperarse que fuera mucho mejor, debido

a los criterios de selección aleatorizados y a las múltiples soluciones que se generan durante el proceso.

En el apéndice C se muestran los resultados individuales para cada una de las instancias.

CAPÍTULO 6

CONCLUSIONES GENERALES

Para una empresa resulta relevante el realizar una buena planeación de la producción, esto con el fin de cumplir los objetivos establecidos. Durante dicha planeación, el considerar un programa de mantenimiento preventivo beneficia en gran medida a la organización, pero ocasiona paros directamente en las líneas de producción. Otro factor a considerar, son los tiempos de preparación dependientes de la secuencia, los cuales son cruciales al generar las diferentes órdenes de trabajo, debido a la gran relevancia que estos tienen en los procesos de producción. Este problema de secuenciación de tareas, aumenta significativamente su complejidad al considerar ambos factores, es por ello que se plantea en una sola máquina, tratando de presentar una estructura sencilla para el análisis de datos.

El problema de secuenciación de tareas presentado en esta tesis, hasta donde se conoce, aún no ha sido estudiado en la literatura. Por lo cual, este corresponde a una nueva variante a los problemas de secuenciación.

En esta nueva variante se maneja como función objetivo el de minimizar el tiempo total de flujo, y se plantean dos formulaciones matemáticas, las cuales fueron sometidas a experimentación para determinar el alcance de cada una. Posteriormente se desarrollaron metodologías que mejoran las formulaciones propuestas, dichas metodologías fueron realizadas en base a la estructura de la formulación y al algoritmo empleado por el *software* de optimización; una vez implementadas las mejoras se puede observar que el *software* tiene un mejor desempeño en su funcionamiento y es capaz de resolver mayor número de instancias.

Pese a las implementaciones de mejora a las formulaciones matemáticas, estas no fueron capaces de resolver en su totalidad las instancias durante la experimentación, en base a esto se propuso un algoritmo metaheurístico compuesto principalmente por un GRASP y un post procesamiento. Dicho algoritmo fue sometido a experimentación para la totalidad de las instancias, reportando resultados de buena calidad en comparación con las soluciones óptimas conocidas, además los tiempos de cómputo requeridos son en magnitud mucho menores a los registrados por las formulaciones matemáticas.

Como trabajo futuro sobre este problema se planea lo siguiente:

1.- Basado en los teoremas presentados a lo largo de la tesis, se planea la creación de nuevas metodologías, que en unión con el procedimiento metaheurístico proporcionen mejores soluciones.

2.- Conforme a la estructura del problema es posible el desarrollo de un algoritmo de programación dinámica, el cual encuentre soluciones óptimas mediante la obtención de soluciones parciales, tanto factibles como infactibles.

3.- Se pretende realizar modificaciones en los modelos matemáticos, de manera que sean capaces de resolver problemas relacionados, y con esto llevar a cabo las comparaciones correspondientes.

4.- Entre la literatura se mencionan trabajos que presentan servicios de mantenimiento no necesariamente fijos. Es por ello que se planea el considerar una distribución de probabilidad para que sean requeridos.

5.- Como ya fue mencionado este problema se plantea sobre una sola máquina, es por ello que se busca evolucionar el problema para múltiples máquinas en paralelo, respetando los mismos supuestos aquí planteados. Para este caso se contemplan modificaciones drásticas en las formulaciones matemáticas.

APÉNDICE A

APÉNDICE A

A.1 RESULTADOS DE LAS FORMULACIONES MATEMÁTICAS

En esta sección se muestran resultados individuales para cada una de las instancias en que se probaron las diferentes formulaciones matemáticas y sus procedimientos de mejora. Esta experimentación abarca tamaños de instancias de 10, 12, 15 y 20 tareas, debido a que fueron estas en se obtuvieron soluciones óptimas. Los resultados están clasificados por tablas en base a su número de tareas a secuenciar (n) y al factor utilizado para calcular el tiempo entre mantenimientos (Xdm). La primera columna representa el nombre de la instancia; las columnas dos, tres, cuatro y cinco corresponden a los tiempos de ejecución en segundos para cada una de las metodologías propuestas. Se utiliza un guion (-) para indicar que el tiempo de ejecución es superior al tiempo límite establecido de 7200 segundos.

Finalmente la columna seis representa el valor óptimo de la instancia. Se utiliza un guion (-) para indicar que ninguna metodología fue capaz de solucionar la instancia a optimalidad.

Tabla A.1: Instancias de $n = 10$ y $T = 2.25dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr101	77.347	33.368	3.276	3.108	138
Pr102	129.691	62.385	16.63	1.273	161
Pr103	95.724	41.933	5.117	3.949	148
Pr104	99.825	46.082	3.863	3.744	138
Pr105	117.846	37.721	10.359	2.597	148
Pr106	235.671	67.548	39.812	8.531	415
Pr107	317.047	57.097	36.147	7.843	426
Pr108	193.737	50.888	17.067	1.835	329
Pr109	226.939	46.644	15.553	3.533	301
Pr110	130.387	75.068	8.509	1.808	319
Pr111	130.277	51.792	19.66	7.807	435
Pr112	124.988	60.451	18.751	2.017	475
Pr113	99.702	42.916	11.14	3.774	379
Pr114	117.844	55.926	4.681	3.022	411
Pr115	147.858	44.819	17.154	2.965	434
Pr101t	144.041	46.301	15.303	2.322	159
Pr102t	286.637	50.341	29.578	0.58	189
Pr103t	98.374	53.899	9.208	2.787	157
Pr104t	127.599	51.715	15.959	3.266	166
Pr105t	127.172	52.042	16.348	3.374	166
Pr106t	275.373	96.44	22.901	8.335	413
Pr107t	233.305	68.282	33.992	6.71	426
Pr108t	270.194	61.215	29.967	2.972	328
Pr109t	251.477	81.136	36.661	7.497	377
Pr110t	245.634	89.513	17.285	2.906	350
Pr111t	192.326	69.202	62.35	6.744	530
Pr112t	94.46	65.443	26.103	2.954	475
Pr113t	176.932	57.782	28.24	2.411	494
Pr114t	156.625	59.842	21.578	4.582	470
Pr115t	126.02	57.611	21.075	3.286	495

Tabla A.2: Instancias de $n = 10$ y $T = 2.5dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr101	86.493	53.072	6.786	1.602	142
Pr102	84.01	37.175	7.863	1.04	147
Pr103	76.077	36.161	12.717	2.342	141
Pr104	99.978	37.331	10.786	2.882	141
Pr105	89.263	32.698	4.929	1.652	139
Pr106	185.942	44.71	16.942	1.03	342
Pr107	207.123	47.846	23.666	2.647	366
Pr108	233.69	65.099	21.965	5.725	332
Pr109	163.021	41.465	14.913	4.71	297
Pr110	160.072	55.989	19.089	0.8	313
Pr111	95.07	59.265	10.202	3.468	396
Pr112	114.165	41.028	8.003	3.549	433
Pr113	78.377	43.072	12.246	2.759	392
Pr114	84.038	70.965	5.538	2.626	375
Pr115	121.078	49.889	12.933	1.98	401
Pr101t	128.514	37.424	17.318	17.582	162
Pr102t	153.676	47.05	15.416	2.14	165
Pr103t	128.794	41.543	6.341	3.299	148
Pr104t	96.971	42.791	12.099	3.606	148
Pr105t	96.811	43.259	12.216	3.435	148
Pr106t	141.703	41.793	23.759	1.11	341
Pr107t	152.382	82.025	21.2	2.211	366
Pr108t	162.672	64.335	19.891	4.484	338
Pr109t	186.517	66.378	14.974	1.89	294
Pr110t	320.196	44.367	14.57	1.13	315
Pr111t	107.197	71.246	18.268	3.067	424
Pr112t	86.457	25.444	3.776	2.297	413
Pr113t	156.017	66.612	17.851	0.77	420
Pr114t	166.578	83.585	8.224	7.002	442
Pr115t	185.313	55.318	17.724	3.001	457

Tabla A.3: Instancias de $n = 10$ y $T = 3dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr101	65.77	20.092	10.874	4.008	134
Pr102	75.832	35.616	5.544	2.283	142
Pr103	80.705	26.925	3.293	1.908	128
Pr104	69.67	48.985	16.117	2.586	138
Pr105	76.979	16.676	9.484	8.905	138
Pr106	161.009	64.584	14.719	3.83	301
Pr107	94.537	33.306	15.99	0.632	315
Pr108	153.803	28.158	11.606	5.039	285
Pr109	89.279	23.978	13.37	8.683	281
Pr110	79.115	49.718	5.821	0.9	290
Pr111	86.393	42.183	10.296	10.381	376
Pr112	125.757	45.63	2.855	3.066	404
Pr113	86.214	43.743	4.389	3.323	374
Pr114	77.735	40.981	8.393	2.034	354
Pr115	76.924	15.257	4.477	1.261	375
Pr101t	72.446	43.477	12.886	2.792	139
Pr102t	92.219	37.518	6.037	1.43	147
Pr103t	88.351	29.422	9.818	2.48	130
Pr104t	98.037	19.999	11.388	7.322	141
Pr105t	98.233	20.327	11.513	7.367	141
Pr106t	203.49	42.463	9.307	1.858	319
Pr107t	85.755	47.783	13.187	0.791	315
Pr108t	140.357	53.18	17.195	2.242	308
Pr109t	116.143	39.921	10.795	1.24	261
Pr110t	109.347	59.608	6.271	2.373	294
Pr111t	86.163	24.82	11.491	3.154	382
Pr112t	78.63	23.276	2.964	2.433	404
Pr113t	127.312	62.073	16.603	1.11	412
Pr114t	162.413	45.038	5.214	3.501	383
Pr115t	124.634	46.301	12.449	2.55	401

Tabla A.4: Instancias de $n = 10$ y $T = 4dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr101	62.291	39.437	4.141	3.917	132
Pr102	54.21	30.326	9.251	3.281	132
Pr103	83.585	17.036	3.76	3.05	124
Pr104	70.248	33.213	3.002	3.254	128
Pr105	74.558	13.4	14.992	4.518	128
Pr106	89.763	38.611	11.42	2.624	290
Pr107	76.201	36.504	3.011	3.715	280
Pr108	86.816	48.579	14.274	7.42	276
Pr109	149.909	26.661	6.805	4.114	258
Pr110	70.856	52.354	3.809	1.857	271
Pr111	61.613	24.071	4.761	17.733	331
Pr112	64.446	50.576	4.635	1.881	391
Pr113	56.437	33.306	7.319	3.418	359
Pr114	64.886	27.004	7.145	2.987	329
Pr115	99.545	32.12	10.577	1.69	372
Pr101t	91.26	19.422	4.541	2.539	136
Pr102t	64.575	25.647	8.799	1.58	136
Pr103t	76.144	29.812	4.867	3.001	124
Pr104t	74.911	29.219	9.532	3.175	130
Pr105t	74.522	29.359	9.345	3.186	130
Pr106t	83.996	44.585	8.551	4.425	297
Pr107t	61.768	45.427	3.761	2.381	280
Pr108t	70.735	41.028	15.366	4.83	283
Pr109t	89.729	40.342	8.91	2.138	260
Pr110t	76.506	21.653	10.369	2.919	265
Pr111t	83.897	16.771	13.697	4.595	361
Pr112t	56.863	20.218	9.782	1.642	391
Pr113t	133.869	28.252	7.269	1.572	384
Pr114t	67.361	34.164	6.057	2.128	346
Pr115t	67.478	46.659	11.825	2.546	376

Tabla A.5: Instancias de $n = 12$ y $T = 2.25dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr201	1441.76	312.72	486.598	11.823	241
Pr202	642.49	168.45	47.097	14.589	206
Pr203	406.601	166.079	72.442	7.657	220
Pr204	1076.92	186.047	88.14	10.822	260
Pr205	423.491	98.436	83.516	15.292	195
Pr206	687.527	215.313	43.056	6.819	424
Pr207	850.848	274.016	32.012	9.784	425
Pr208	704.251	240.99	46.371	12.378	437
Pr209	775.934	520.95	42.265	4.424	411
Pr210	3769.03	435.212	57.267	2.77	447
Pr211	657.904	392.139	71.433	1.973	604
Pr212	4449.17	227.044	55.587	3.288	540
Pr213	2238.34	293.531	93.319	16.326	609
Pr214	577.688	403.028	123.46	10.221	651
Pr215	774.484	187.623	83.491	9.271	582
Pr201t	3105.98	262.55	249.961	18.665	241
Pr202t	2698.61	372.141	1355.03	17.916	264
Pr203t	3877.19	353.202	97.859	9.262	245
Pr204t	3738.2	294.218	167.28	13.651	260
Pr205t	1716.42	222.021	100.792	2.908	228
Pr206t	3005.12	523.773	250.725	23.76	550
Pr207t	1274.41	416.82	88.222	6.663	468
Pr208t	1686.71	319.256	110.246	6.397	457
Pr209t	2152.98	419.549	183.605	11.755	513
Pr210t	1594.63	375.76	151.383	0.94	507
Pr211t	4156.59	355.495	45.555	1.221	620
Pr212t	1077.11	295.31	91.347	12.13	616
Pr213t	5084.84	199.947	142.195	19.935	688
Pr214t	794.624	323.312	84.719	12.397	652
Pr215t	2798.74	280.115	99.857	6.441	684

Tabla A.6: Instancias de $n = 12$ y $T = 2.5dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr201	586.194	208.168	54.316	5.744	212
Pr202	401.88	127.453	83.112	22.319	200
Pr203	655.86	113.958	52.947	9.46	210
Pr204	560.338	158.949	56.581	10.456	232
Pr205	358.459	83.71	44.229	48.503	191
Pr206	623.958	182.849	78.724	8.959	413
Pr207	778.626	198.34	38.736	3.764	386
Pr208	471.981	190.992	72.665	7.357	413
Pr209	633.629	328.507	64.654	7.096	410
Pr210	921.404	188.854	46.146	3.541	410
Pr211	895.633	174.893	48.803	2.843	558
Pr212	621.554	161.929	38.159	4.616	516
Pr213	680.975	161.165	40.315	2.844	535
Pr214	911.84	193.208	104.866	7.505	601
Pr215	533.993	194.674	26.85	19.254	549
Pr201t	969.159	182.038	51.249	9.84	212
Pr202t	997.844	227.777	185.049	13.724	228
Pr203t	1051.35	271.13	60.668	9.811	220
Pr204t	540.64	196.951	118.285	9.424	230
Pr205t	1027.78	216.763	52.993	2.755	220
Pr206t	1381.16	251.411	53.125	2.978	453
Pr207t	1324.06	209.103	62.104	1.801	456
Pr208t	952.729	301.597	89.989	14.553	437
Pr209t	1337.01	339.942	79.872	6.617	453
Pr210t	1557.9	475.881	51.808	9.139	460
Pr211t	3258.88	232.145	73.149	8.333	655
Pr212t	2281.97	200.383	36.062	1.696	540
Pr213t	-	189.323	94.895	9.398	593
Pr214t	1528.36	192.302	65.854	6.712	614
Pr215t	1167.42	261.145	98.871	17.751	659

Tabla A.7: Instancias de $n = 12$ y $T = 3dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr201	499.64	130.261	44.617	3.295	191
Pr202	578.624	106.845	41.531	8.91	185
Pr203	378.167	145.284	40.047	12.941	196
Pr204	804.626	121.415	41.125	12.07	214
Pr205	261.177	85.458	43.245	24.436	185
Pr206	1401.4	617.218	60.123	13.249	402
Pr207	323.562	236.497	49.297	7.206	379
Pr208	746.071	179.978	121.581	16.549	413
Pr209	768.025	200.602	52.312	14.232	386
Pr210	299.008	149.636	54.506	4.084	391
Pr211	624.41	228.526	30.123	12.852	546
Pr212	515.724	136.563	16.882	15.198	467
Pr213	316.488	195.017	21.755	16.429	499
Pr214	1004.23	165.408	42.308	17.681	536
Pr215	492.012	147.343	56.8	21.319	526
Pr201t	3194.88	162.382	42.682	2.821	191
Pr202t	1978.86	152.257	100.34	11.51	202
Pr203t	518.224	126.22	78.22	17.218	209
Pr204t	451.238	233.799	40.342	14.598	214
Pr205t	1159.15	137.094	66.55	11.589	200
Pr206t	623.318	189.479	53.15	8.75	432
Pr207t	527.082	216.248	38.173	8.292	386
Pr208t	1283.73	135.035	47.814	8.403	402
Pr209t	955.88	157.795	261.707	6.598	410
Pr210t	3027.87	181.351	74.381	3.124	405
Pr211t	941.619	249.149	45.116	1.921	565
Pr212t	587.266	160.323	34.634	13.248	495
Pr213t	791.751	168.839	51.247	16.097	550
Pr214t	601.372	169.854	59.125	22.206	570
Pr215t	624.175	182.428	99.466	6.821	582

Tabla A.8: Instancias de $n = 12$ y $T = 4dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr201	328.354	116.268	39.312	14.398	181
Pr202	342.251	105.316	136.969	21.761	182
Pr203	379.441	118.046	15.584	12.7	191
Pr204	273.985	90.403	30.56	32.61	199
Pr205	227.858	78.78	77.813	21.512	182
Pr206	319.316	109.809	50.466	20.292	386
Pr207	475.871	113.522	30.841	12.249	370
Pr208	395.868	105.129	25.928	37.582	374
Pr209	313.297	93.179	38.302	23.71	366
Pr210	393.539	123.818	34.524	14.114	374
Pr211	405.68	66.675	12.131	11.715	495
Pr212	537.337	149.527	54.567	19.618	460
Pr213	523.868	86.269	33.628	8.185	435
Pr214	418.551	174.472	36.086	18.367	512
Pr215	498.548	127.156	41.39	52.516	504
Pr201t	460.238	113.6	137.749	20.663	181
Pr202t	488.961	133.474	27.478	9.223	181
Pr203t	582.663	171.43	68.687	18.526	194
Pr204t	394.628	91.603	36.392	28.314	197
Pr205t	655.467	131.306	50.42	28.687	191
Pr206t	508.773	360.659	89.679	35.326	399
Pr207t	644.877	160.962	26.103	27.008	369
Pr208t	389.037	119.856	22.542	48.91	374
Pr209t	419.268	162.803	22.278	26.222	385
Pr210t	378.459	112.602	63.15	17.44	377
Pr211t	339.349	69.015	27.555	11.57	498
Pr212t	485.82	121.322	33.371	22.599	463
Pr213t	433.574	98.795	43.119	11.592	479
Pr214t	418.133	138.186	50.404	18.989	508
Pr215t	428.894	127.188	59.14	20.422	526

Tabla A.9: Instancias de $n = 15$ y $T = 2.25dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr301	-	1698.31	1542.19	41.302	315
Pr302	-	665.22	483.198	42.068	274
Pr303	5647.33	1925.18	277.105	28.088	308
Pr304	-	1078.09	1013.11	34.082	292
Pr305	-	1399.03	254.282	24.572	318
Pr306	-	2177.57	1737.09	11.576	646
Pr307	-	1938.89	165.377	17.973	599
Pr308	-	-	3600.98	27.175	671
Pr309	-	3729.03	2334.55	11.689	688
Pr310	-	1773.87	3639.52	9.852	658
Pr311	-	1987.89	619.044	26.051	847
Pr312	-	3785.9	1283.44	45.913	814
Pr313	-	2735.15	456.553	40.932	789
Pr314	-	2111.44	1286.61	32.428	834
Pr315	-	2417.44	1378.78	70.408	858
Pr301t	-	5430.08	2935.97	26.512	353
Pr302t	-	2412.32	185.937	14.427	315
Pr303t	-	4451.83	801.252	9.223	408
Pr304t	-	-	414.735	12.272	355
Pr305t	-	3483.63	1808.43	31.181	354
Pr306t	-	4777.09	-	2.08	738
Pr307t	-	5859.04	3406.93	48.249	689
Pr308t	-	-	1877.23	23.854	731
Pr309t	-	2563.96	4257.85	10.658	688
Pr310t	-	4022.35	3439.51	17.042	658
Pr311t	-	2304.62	672.848	12.2	881
Pr312t	-	7078.45	5823.41	45.394	1012
Pr313t	-	3696.94	561.214	21.588	853
Pr314t	-	6395.43	1979.38	12.769	951
Pr315t	-	-	-	46.467	1036

Tabla A.10: Instancias de $n = 15$ y $T = 2.5dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr301	-	996.316	1054.69	77.032	301
Pr302	-	810.113	507.503	71.076	264
Pr303	-	3905.83	303.353	41.681	300
Pr304	-	1291.35	959.832	155.899	287
Pr305	-	1255.26	1416.15	66.641	313
Pr306	-	1437.97	429.72	5.626	584
Pr307	-	1494.08	969.219	9.611	601
Pr308	-	2217.27	2010.67	13.434	644
Pr309	-	2096.08	1672.64	15.387	636
Pr310	-	1683.64	1852.89	38.369	618
Pr311	-	1383.29	380.08	32.608	796
Pr312	-	1221.47	944.134	34.761	765
Pr313	-	999.763	1194.59	39.919	746
Pr314	-	1041.4	1192.29	39.262	782
Pr315	-	1742.98	2027.84	229.53	831
Pr301t	-	1812.5	528.219	31.557	314
Pr302t	-	1669.27	718.01	46.699	301
Pr303t	-	3456.81	496.029	34.264	362
Pr304t	-	3090.36	320.945	17.547	327
Pr305t	-	1868.64	478.63	50.982	317
Pr306t	-	2492.08	3321.36	13.496	646
Pr307t	-	6463.31	158.223	3.659	632
Pr308t	-	5206.91	614.597	33.462	670
Pr309t	-	3765.72	2230.08	15.355	636
Pr310t	-	2874.26	3630.29	38.571	622
Pr311t	-	2176.17	592.992	40.728	847
Pr312t	-	2656.09	1866.54	20.794	861
Pr313t	-	1930.2	268.478	17.181	782
Pr314t	-	3503.28	176.796	11.351	850
Pr315t	-	4431.55	-	91.594	971

Tabla A.11: Instancias de $n = 15$ y $T = 3dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr301	-	860.517	1035.37	188.034	288
Pr302	-	573.225	474.929	108.382	252
Pr303	-	1260.85	1040.56	849.859	278
Pr304	-	632.412	626.734	683.193	281
Pr305	-	946.615	1681.35	282.381	306
Pr306	-	895.04	1347.39	35.28	576
Pr307	-	698.385	1233.03	25.897	545
Pr308	-	1052.21	436.371	37.545	592
Pr309	-	1145.75	2441.5	46.334	595
Pr310	-	1164.45	2254.42	40.831	585
Pr311	-	800.535	685.833	114.428	770
Pr312	-	569.591	822.282	99.426	712
Pr313	-	745.638	941.357	221.844	716
Pr314	-	1015.13	373.025	136.286	778
Pr315	-	1895.6	1255.29	231.734	777
Pr301t	-	1368.64	1177.7	57.491	300
Pr302t	-	1111.29	219.572	35.207	271
Pr303t	-	1608.89	473.252	30.156	302
Pr304t	-	3467.81	1358.18	64.532	310
Pr305t	-	1028.9	2125.08	33.83	309
Pr306t	-	1508.31	4125.35	13.987	610
Pr307t	-	1686.98	2992.79	14.88	601
Pr308t	-	1696.4	436.179	18.08	606
Pr309t	-	962.338	1826.69	52.317	595
Pr310t	-	2222.8	2034.77	169.126	595
Pr311t	-	1444.4	509.546	54.34	767
Pr312t	-	2187.76	1704.73	46.799	813
Pr313t	-	1186.84	1431.56	40.313	759
Pr314t	-	1304.7	2916.99	31.621	836
Pr315t	-	2635.11	2841.11	167.838	911

Tabla A.12: Instancias de $n = 15$ y $T = 4dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr301	-	566.97	690.258	219.686	278
Pr302	-	547.408	1541.93	57.179	256
Pr303	-	856.57	696.389	210.812	262
Pr304	-	678.137	1148.56	502.548	264
Pr305	-	414.573	2085.01	316.019	288
Pr306	-	735.436	1218.66	37.007	562
Pr307	-	573.896	455.351	1824.23	513
Pr308	-	773.109	3082.19	212.135	574
Pr309	-	480.53	3277.07	5013.53	563
Pr310	-	748.088	1174.66	525.92	545
Pr311	-	997.595	524.132	42.757	704
Pr312	-	737.791	217.035	446.834	679
Pr313	-	647.123	588.807	63.493	681
Pr314	-	597.718	211.66	99.024	749
Pr315	-	728.182	957.597	230.359	757
Pr301t	-	737.168	640.108	522.736	280
Pr302t	-	662.318	619.589	176	252
Pr303t	-	703.097	731.239	1226.37	278
Pr304t	-	1361.06	1584.14	648.028	286
Pr305t	-	688.713	729.088	376.134	288
Pr306t	-	840.908	5168.91	60.669	583
Pr307t	-	971.512	552.072	38.86	527
Pr308t	-	954.539	708.665	80.559	572
Pr309t	-	1168.51	4215.9	281.379	563
Pr310t	-	740.396	5724.57	499.344	560
Pr311t	-	1053.69	796.619	503.431	746
Pr312t	-	817.102	951.169	177.17	706
Pr313t	-	843.934	633.536	238.119	700
Pr314t	-	872.201	1075.79	183.878	758
Pr315t	-	1065.3	927.865	83.164	783

Tabla A.13: Instancias de $n = 20$ y $T = 2.25dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr401	-	-	-	3786.66	473
Pr402	-	-	-	3337.27	498
Pr403	-	-	-	1737.8	516
Pr404	-	-	-	58.828	460
Pr405	-	-	6965.18	67.206	459
Pr406	-	-	-	175.33	1088
Pr407	-	-	-	2065.2	1254
Pr408	-	-	3252.47	34.991	1075
Pr409	-	-	-	136.486	1083
Pr410	-	-	-	807.18	1202
Pr411	-	-	-	2287.33	1530
Pr412	-	-	-	1275.59	1305
Pr413	-	-	-	4309.92	1478
Pr414	-	-	-	5262.51	1313
Pr415	-	-	-	1494.8	1456
Pr401t	-	-	-	106.221	539
Pr402t	-	-	-	2103.71	641
Pr403t	-	-	-	530.107	592
Pr404t	-	-	-	27.02	547
Pr405t	-	-	-	26.941	553
Pr406t	-	-	-	727.885	1223
Pr407t	-	-	-	-	-
Pr408t	-	-	-	1958.11	1173
Pr409t	-	-	-	5942.78	1136
Pr410t	-	-	-	1059.23	1494
Pr411t	-	-	-	1126.7	1781
Pr412t	-	-	-	3272.4	1462
Pr413t	-	-	-	5492.81	1628
Pr414t	-	-	-	-	-
Pr415t	-	-	-	-	-

Tabla A.14: Instancias de $n = 20$ y $T = 2.5dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr401	-	-	-	5957.07	461
Pr402	-	5912.33	-	5889.21	479
Pr403	-	-	-	6188.92	506
Pr404	-	-	-	1311	457
Pr405	-	-	-	1053.51	453
Pr406	-	-	-	774.826	1003
Pr407	-	-	-	50.435	1128
Pr408	-	-	-	778.086	1010
Pr409	-	-	-	1015.99	1022
Pr410	-	-	-	834.56	1099
Pr411	-	-	-	3194.34	1441
Pr412	-	-	-	2552.79	1275
Pr413	-	-	-	3903.43	1439
Pr414	-	-	-	-	-
Pr415	-	-	-	620.635	1382
Pr401t	-	-	-	900.079	514
Pr402t	-	-	-	420.766	566
Pr403t	-	-	-	1577.03	549
Pr404t	-	-	-	41.917	488
Pr405t	-	-	-	24.056	488
Pr406t	-	-	-	209.447	1088
Pr407t	-	-	-	-	-
Pr408t	-	-	-	574.255	1064
Pr409t	-	-	-	710.912	1083
Pr410t	-	-	-	2120.23	1242
Pr411t	-	-	-	1111.23	1589
Pr412t	-	-	-	3053.58	1385
Pr413t	-	-	-	-	-
Pr414t	-	-	-	-	-
Pr415t	-	-	-	2022.46	1556

Tabla A.15: Instancias de $n = 20$ y $T = 3dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr401	-	-	-	-	-
Pr402	-	-	-	5468.97	464
Pr403	-	-	-	-	-
Pr404	-	-	-	1301.27	446
Pr405	-	-	-	857.084	445
Pr406	-	-	-	283.095	997
Pr407	-	-	-	3566.56	1038
Pr408	-	-	-	-	-
Pr409	-	-	-	2755.54	951
Pr410	-	-	-	391.703	1041
Pr411	-	-	-	3696.16	1346
Pr412	-	-	-	-	-
Pr413	-	-	-	-	-
Pr414	-	-	-	-	-
Pr415	-	-	-	5212.76	1353
Pr401t	-	-	-	980.248	463
Pr402t	-	-	-	818.678	521
Pr403t	-	-	-	625.767	514
Pr404t	-	-	-	36.411	493
Pr405t	-	-	-	21.278	495
Pr406t	-	-	-	465.695	1007
Pr407t	-	-	-	2116.84	1099
Pr408t	-	-	-	289.023	1010
Pr409t	-	-	-	505.021	971
Pr410t	-	-	-	64.116	1097
Pr411t	-	-	-	4309.51	1434
Pr412t	-	-	-	4240.53	1326
Pr413t	-	-	-	5765.8	1399
Pr414t	-	-	-	1454.15	1370
Pr415t	-	-	-	1971.87	1439

Tabla A.16: Instancias de $n = 20$ y $T = 4dm$

Instancia	Modelo A Tiempo seg.	Mod.A+D.V. Tiempo seg.	Modelo B Tiempo seg.	Mod.B+Imp. Tiempo seg.	Valor Óptimo
Pr401	-	4427.32	-	-	434
Pr402	-	-	-	-	-
Pr403	-	-	-	-	-
Pr404	-	3930.04	-	-	434
Pr405	-	4472.75	-	-	433
Pr406	-	-	-	3523.88	921
Pr407	-	-	-	-	-
Pr408	-	-	-	-	-
Pr409	-	-	-	-	-
Pr410	-	-	-	-	-
Pr411	-	-	-	-	-
Pr412	-	-	-	-	-
Pr413	-	-	-	-	-
Pr414	-	-	-	-	-
Pr415	-	-	-	-	-
Pr401t	-	6923.34	-	-	441
Pr402t	-	-	-	-	-
Pr403t	-	-	-	-	-
Pr404t	-	-	-	-	-
Pr405t	-	-	-	-	-
Pr406t	-	-	-	-	-
Pr407t	-	-	-	-	-
Pr408t	-	-	-	-	-
Pr409t	-	-	-	-	-
Pr410t	-	-	-	-	-
Pr411t	-	-	-	-	-
Pr412t	-	-	-	-	-
Pr413t	-	-	-	-	-
Pr414t	-	-	-	-	-
Pr415t	-	-	-	-	-

APÉNDICE B

APÉNDICE B

B.1 RESULTADOS DEL PROCEDIMIENTO METAHEURÍSTICO EN COMPARACIÓN CON LA SOLUCIÓN ÓPTIMA

En esta sección se muestran los resultados obtenidos por el procedimiento metaheurístico propuesto, para cada una de las instancias en que se conoce la solución óptima. Esta experimentación abarca tamaños de instancias de 10, 12, 15 y 20 tareas. Los resultados están clasificados por tablas en base a su número de tareas a secuenciar (n) y al factor utilizado para calcular el tiempo entre mantenimientos (Xdm). La primera columna corresponde al nombre de la instancia; la columna dos representa el valor de la función objetivo de la solución obtenida por el algoritmo del vecino más cercano; la columna tres muestra las soluciones óptimas de las instancias; la columna cuatro contiene el valor de la función objetivo para la solución encontrada por el procedimiento metaheurístico (GRASP+P.P.), en la columna cinco se muestra la desviación estándar de las soluciones encontradas al realizar 10 réplicas del algoritmo para cada instancia; la columna seis indica el GAP de la solución encontrada con respecto a la solución óptima; y por último la columna siete corresponde a los tiempos promedio de ejecución en segundos para la metodología propuesta.

Tabla B.1: Instancias de $n = 10$ y $T = 2.25dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr101	156	138	138	± 0	0%	0.9326
Pr102	209	161	161	± 0	0%	0.8513
Pr103	232	148	148	± 0	0%	0.8574
Pr104	169	138	138	± 0	0%	0.7962
Pr105	178	148	148	± 0	0%	0.818
Pr106	462	415	417	± 2.213	0.48%	0.921
Pr107	555	426	426	± 6.279	0%	0.8148
Pr108	349	329	329	± 0	0%	0.8099
Pr109	325	301	301	± 0	0%	0.7792
Pr110	341	319	319	± 0	0%	0.835
Pr111	487	435	442	± 0	1.61%	0.7972
Pr112	683	475	475	± 8.966	0%	0.7889
Pr113	457	379	379	± 0	0%	0.7957
Pr114	582	411	411	± 0	0%	0.8246
Pr115	538	434	434	± 0	0%	0.8191
Pr101t	197	159	159	± 0	0%	0.7738
Pr102t	234	189	189	± 2.415	0%	0.9463
Pr103t	244	157	157	± 0	0%	0.8373
Pr104t	215	166	166	± 0	0%	0.8297
Pr105t	215	166	166	± 0	0%	0.8526
Pr106t	557	413	413	± 0.527	0%	0.9094
Pr107t	555	426	426	± 4.110	0%	0.8108
Pr108t	362	328	328	± 0	0%	0.8502
Pr109t	623	377	377	± 0	0%	0.8
Pr110t	368	350	350	± 0	0%	0.8318
Pr111t	581	530	530	± 7.589	0%	0.9098
Pr112t	541	475	475	± 0	0%	0.7586
Pr113t	601	494	494	± 0	0%	0.8719
Pr114t	590	470	470	± 0	0%	0.7982
Pr115t	539	495	495	± 0	0%	0.8234

Tabla B.2: Instancias de $n = 10$ y $T = 2.5dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr101	154	142	142	± 0	0%	3.4138
Pr102	172	147	147	± 0	0%	3.126
Pr103	195	141	141	± 0	0%	3.306
Pr104	144	141	141	± 0	0%	3.5024
Pr105	188	139	139	± 0	0%	3.3002
Pr106	382	342	342	± 0	0%	2.937
Pr107	513	366	366	± 0	0%	2.8586
Pr108	361	332	332	± 0	0%	3.0548
Pr109	351	297	297	± 0	0%	3.1674
Pr110	333	313	313	± 0	0%	3.0158
Pr111	522	396	396	± 1.032	0%	3.1914
Pr112	595	433	433	± 0	0%	2.9972
Pr113	472	392	392	± 0	0%	3.329
Pr114	442	375	375	± 21.447	0%	3.2394
Pr115	446	401	401	± 0	0%	3.053
Pr101t	174	162	162	± 0	0%	3.1006
Pr102t	196	165	165	± 0	0%	2.9578
Pr103t	231	148	148	± 0	0%	3.1196
Pr104t	150	148	148	± 0	0%	3.172
Pr105t	150	148	148	± 0	0%	3.191
Pr106t	372	341	341	± 0	0%	3.0146
Pr107t	513	366	366	± 5.481	0%	2.9606
Pr108t	355	338	338	± 0	0%	2.9462
Pr109t	317	294	294	± 0	0%	3.0216
Pr110t	315	315	315	± 0	0%	2.9028
Pr111t	620	424	424	± 0	0%	2.9876
Pr112t	519	413	413	± 0	0%	3.1842
Pr113t	623	420	420	± 0	0%	3.0294
Pr114t	442	442	442	± 0	0%	3.1398
Pr115t	461	457	457	± 0	0%	2.907

Tabla B.3: Instancias de $n = 10$ y $T = 3dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr101	144	134	134	± 2.529	0%	3.6324
Pr102	157	142	142	± 0	0%	3.3824
Pr103	143	128	128	± 0	0%	3.5128
Pr104	140	138	138	± 0	0%	3.5122
Pr105	158	138	138	± 0	0%	3.5886
Pr106	332	301	301	± 7.375	0%	3.329
Pr107	346	315	315	± 0	0%	3.2774
Pr108	335	285	285	± 0	0%	3.2468
Pr109	332	281	281	± 0	0%	3.2962
Pr110	310	290	290	± 0	0%	3.2468
Pr111	424	376	376	± 4.638	0%	3.4028
Pr112	491	404	404	± 0	0%	3.3498
Pr113	411	374	374	± 9.811	0%	3.4878
Pr114	455	354	354	± 0	0%	3.412
Pr115	428	375	375	± 0	0%	3.3408
Pr101t	146	139	139	± 0	0%	3.4904
Pr102t	160	147	147	± 0	0%	3.3216
Pr103t	157	130	130	± 0	0%	3.394
Pr104t	149	141	141	± 0	0%	3.4374
Pr105t	149	141	141	± 0	0%	3.4052
Pr106t	320	319	319	± 0	0%	3.333
Pr107t	346	315	315	± 0	0%	3.3082
Pr108t	359	308	308	± 0	0%	3.2776
Pr109t	335	261	261	± 0	0%	3.3126
Pr110t	298	294	294	± 0	0%	3.24
Pr111t	496	382	382	± 0	0%	3.3868
Pr112t	491	404	404	± 0	0%	3.5238
Pr113t	443	412	412	± 0	0%	3.2596
Pr114t	448	383	383	± 0	0%	3.3344
Pr115t	446	401	401	± 0	0%	3.1854

Tabla B.4: Instancias de $n = 10$ y $T = 4dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr101	150	132	132	± 0	0%	3.5785
Pr102	151	132	132	± 0	0%	3.4783
Pr103	128	124	124	± 0	0%	3.5866
Pr104	128	128	128	± 0	0%	3.4818
Pr105	157	128	128	± 0	0%	3.5419
Pr106	304	290	290	± 0	0%	3.5387
Pr107	321	280	280	± 0	0%	3.5232
Pr108	302	276	276	± 0	0%	3.4781
Pr109	297	258	261	± 0	1.16%	3.4781
Pr110	282	271	271	± 0	0%	3.4359
Pr111	379	331	331	± 0	0%	3.4332
Pr112	414	391	391	± 0	0%	3.4692
Pr113	421	359	359	± 0	0%	3.4672
Pr114	329	329	329	± 0	0%	3.4809
Pr115	377	372	372	± 0	0%	3.4091
Pr101t	148	136	136	± 0	0%	3.3822
Pr102t	139	136	136	± 0	0%	3.3273
Pr103t	138	124	124	± 0	0%	3.4018
Pr104t	149	130	130	± 0	0%	3.4005
Pr105t	149	130	130	± 0	0%	3.3942
Pr106t	298	297	297	± 0	0%	3.3616
Pr107t	321	280	280	± 0	0%	3.3966
Pr108t	318	283	283	± 0	0%	3.3463
Pr109t	300	260	260	± 0	0%	3.3649
Pr110t	276	265	265	± 0	0%	3.3553
Pr111t	378	361	361	± 0	0%	3.3926
Pr112t	414	391	391	± 0	0%	3.4074
Pr113t	433	384	384	± 0	0%	3.3533
Pr114t	350	346	346	± 1.264	0%	3.4013
Pr115t	385	376	376	± 0	0%	3.3588

Tabla B.5: Instancias de $n = 12$ y $T = 2.25dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr201	316	241	241	± 0	0%	5.0416
Pr202	282	206	215	± 0	4.37%	6.5211
Pr203	274	220	220	± 0	0%	5.1931
Pr204	325	260	279	± 4.131	7.31%	4.74
Pr205	236	195	195	± 0	0%	5.4317
Pr206	490	424	426	± 0	0.47%	4.757
Pr207	532	425	425	± 0	0%	4.9496
Pr208	530	437	437	± 0	0%	4.9135
Pr209	591	411	411	± 0	0%	4.7287
Pr210	557	447	447	± 6.279	0%	4.6216
Pr211	720	604	604	± 0	0%	4.9055
Pr212	765	540	540	± 0	0%	4.8916
Pr213	635	609	609	± 0	0%	4.9497
Pr214	765	651	651	± 0	0%	4.8932
Pr215	726	582	582	± 6.324	0%	5.2929
Pr201t	286	241	241	± 0	0%	4.5569
Pr202t	279	264	264	± 0	0%	4.508
Pr203t	290	245	245	± 0	0%	4.8139
Pr204t	343	260	260	± 7.766	0%	4.6629
Pr205t	280	228	228	± 0	0%	4.561
Pr206t	631	550	550	± 0	0%	4.362
Pr207t	504	468	468	± 0	0%	4.4944
Pr208t	595	457	457	± 0	0%	4.5838
Pr209t	640	513	513	± 0	0%	4.377
Pr210t	564	507	507	± 0	0%	4.4701
Pr211t	782	620	620	± 0	0%	4.8411
Pr212t	700	616	616	± 0	0%	4.6251
Pr213t	780	688	688	± 0.483	0%	4.8463
Pr214t	748	652	652	± 0	0%	4.8353
Pr215t	865	684	684	± 0	0%	4.543

Tabla B.6: Instancias de $n = 12$ y $T = 2.5dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr201	219	212	212	± 0	0%	5.0735
Pr202	219	200	200	± 0	0%	5.3058
Pr203	239	210	210	± 0	0%	5.3578
Pr204	313	232	255	± 1.337	9.91%	4.9318
Pr205	220	191	192	± 1.955	0.52%	5.4211
Pr206	529	413	413	± 8.854	0%	4.8785
Pr207	470	386	386	± 5.059	0%	5.0754
Pr208	483	413	413	± 0	0%	4.874
Pr209	429	410	410	± 0	0%	4.8073
Pr210	520	410	410	± 0	0%	4.7889
Pr211	729	558	558	± 0	0%	5.0581
Pr212	649	516	516	± 1.897	0%	5.05
Pr213	639	535	535	± 0	0%	4.8862
Pr214	753	601	601	± 2.529	0%	4.9865
Pr215	649	549	549	± 11.067	0%	5.1981
Pr201t	219	212	212	± 0	0%	4.7374
Pr202t	283	228	228	± 1.581	0%	4.532
Pr203t	274	220	220	± 0	0%	4.8385
Pr204t	278	230	246	± 5.87	6.96%	4.8243
Pr205t	300	220	220	± 0	0%	4.491
Pr206t	512	453	453	± 5.059	0%	4.4018
Pr207t	521	456	456	± 3.027	0%	4.6751
Pr208t	530	437	437	± 0	0%	4.7376
Pr209t	467	453	453	± 7.378	0%	4.5221
Pr210t	538	460	460	± 0.699	0%	4.4894
Pr211t	854	655	655	± 0	0%	4.8695
Pr212t	665	540	540	± 0	0%	4.9038
Pr213t	612	593	593	± 6.762	0%	4.8329
Pr214t	740	614	614	± 0	0%	4.8613
Pr215t	817	659	659	± 0	0%	4.7771

Tabla B.7: Instancias de $n = 12$ y $T = 3dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr201	207	191	191	± 4.261	0%	5.3809
Pr202	251	185	185	± 0.843	0%	5.5867
Pr203	242	196	196	± 0	0%	5.473
Pr204	286	214	214	± 0	0%	5.3766
Pr205	222	185	185	± 1.054	0%	5.6995
Pr206	473	402	408	± 1.581	1.49%	5.3256
Pr207	446	379	379	± 0	0%	5.3119
Pr208	445	413	413	± 0	0%	5.1822
Pr209	436	386	386	± 0	0%	5.2172
Pr210	446	391	391	± 0.316	0%	5.2292
Pr211	601	546	546	± 0	0%	5.3251
Pr212	628	467	467	± 1.897	0%	5.2858
Pr213	625	499	499	± 6.279	0%	5.1733
Pr214	637	536	536	± 0	0%	5.2809
Pr215	564	526	539	± 0	2.47%	5.4689
Pr201t	207	191	191	± 0	0%	5.0989
Pr202t	240	202	202	± 0	0%	4.9794
Pr203t	217	209	209	± 2.108	0%	5.1744
Pr204t	284	214	214	± 2.898	0%	5.256
Pr205t	246	200	200	± 0	0%	5.0321
Pr206t	499	432	432	± 1.032	0%	4.7777
Pr207t	470	386	386	± 0	0%	5.1354
Pr208t	499	402	402	± 0	0%	5.0921
Pr209t	429	410	410	± 0	0%	4.9282
Pr210t	489	405	405	± 0	0%	4.9126
Pr211t	647	565	565	± 0	0%	5.211
Pr212t	588	495	495	± 0	0%	5.3503
Pr213t	626	550	550	± 0	0%	5.2046
Pr214t	621	570	570	± 1.449	0%	5.2672
Pr215t	632	582	582	± 0	0%	5.142

Tabla B.8: Instancias de $n = 12$ y $T = 4dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr201	207	181	185	± 1.64	2.21%	5.9721
Pr202	238	182	185	± 0	1.65%	6.0711
Pr203	196	191	191	± 0	0%	6.0442
Pr204	245	199	199	± 2.635	0%	5.9518
Pr205	212	182	182	± 0.948	0%	5.96
Pr206	475	386	386	± 0	0%	5.82
Pr207	385	370	370	± 0	0%	5.9677
Pr208	409	374	374	± 0.52	0%	5.9686
Pr209	409	366	366	± 0	0%	5.9868
Pr210	424	374	374	± 1.032	0%	5.7969
Pr211	563	495	495	± 0	0%	5.8861
Pr212	482	460	460	± 0	0%	5.8597
Pr213	630	435	435	± 0	0%	5.9406
Pr214	525	512	512	± 0	0%	5.9895
Pr215	541	504	504	± 0	0%	5.9705
Pr201t	199	181	181	± 0	0%	5.7201
Pr202t	230	181	181	± 0	0%	5.4771
Pr203t	198	194	194	± 1.26	0%	5.8494
Pr204t	243	197	197	± 0.96	0%	5.9587
Pr205t	220	191	191	± 0	0%	5.6094
Pr206t	462	399	399	± 0.948	0%	5.3999
Pr207t	415	369	369	± 0	0%	5.6679
Pr208t	430	374	374	± 0.316	0%	5.6062
Pr209t	407	385	385	± 1.686	0%	5.4444
Pr210t	394	377	377	± 0	0%	5.5836
Pr211t	578	498	498	± 0	0%	5.8238
Pr212t	508	463	463	± 0	0%	5.8782
Pr213t	540	479	479	± 0	0%	5.8259
Pr214t	517	508	508	± 0	0%	5.8298
Pr215t	560	526	526	± 0	0%	5.8904

Tabla B.9: Instancias de $n = 15$ y $T = 2.25dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr301	518	315	315	±0	0%	7.9077
Pr302	305	274	274	±0	0%	8.6825
Pr303	406	308	335	±0.875	8.77%	8.3112
Pr304	373	292	302	±1.264	3.42%	8.1776
Pr305	428	318	322	±10.637	1.26%	7.671
Pr306	728	646	646	±0	0%	7.3545
Pr307	617	599	599	±6.746	0%	7.6905
Pr308	793	671	673	±0	0.30%	7.8096
Pr309	796	688	688	±0	0%	7.418
Pr310	831	658	658	±0	0%	7.108
Pr311	1026	847	847	±12.175	0%	8.0323
Pr312	886	814	814	±0.421	0%	8.0937
Pr313	958	789	789	±11.157	0%	8.0927
Pr314	871	834	834	±0	0%	8.0363
Pr315	1143	858	859	±11.384	0.12%	7.9847
Pr301t	440	353	353	±0	0%	7.1066
Pr302t	347	315	315	±0	0%	7.7861
Pr303t	468	408	408	±0	0%	7.2348
Pr304t	472	355	355	±0	0%	7.101
Pr305t	404	354	354	±0	0%	7.3542
Pr306t	840	738	738	±0.966	0%	7.0183
Pr307t	868	689	689	±0.483	0%	6.8019
Pr308t	852	731	731	±0	0%	7.1643
Pr309t	755	688	688	±0	0%	7.475
Pr310t	759	658	658	±0	0%	7.2206
Pr311t	983	881	881	±0	0%	7.7095
Pr312t	1090	1012	1012	±2.529	0%	7.0316
Pr313t	958	853	853	±0	0%	7.3803
Pr314t	1068	951	951	±0	0%	7.4071
Pr315t	1279	1036	1036	±0	0%	6.909

Tabla B.10: Instancias de $n = 15$ y $T = 2.5dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr301	408	301	301	±0	0%	8.3294
Pr302	297	264	264	±4.647	0%	9.1884
Pr303	342	300	304	±2.951	1.33%	8.8726
Pr304	389	287	287	±5.493	0%	8.6315
Pr305	398	313	321	±0.632	2.56%	8.0879
Pr306	740	584	584	±0	0%	7.6094
Pr307	626	601	601	±0.948	0%	7.9481
Pr308	793	644	644	±1.349	0%	7.9093
Pr309	684	636	636	±0	0%	7.852
Pr310	679	618	618	±0	0%	7.6297
Pr311	1090	796	796	±1.897	0%	8.452
Pr312	810	765	765	±1.264	0%	8.5074
Pr313	824	746	778	±0.632	4.29%	8.4931
Pr314	992	782	782	±0	0%	8.3277
Pr315	975	831	860	±0	3.49%	8.3683
Pr301t	459	314	314	±0	0%	7.7062
Pr302t	355	301	301	±0	0%	8.3375
Pr303t	449	362	362	±0	0%	7.8374
Pr304t	393	327	327	±0.843	0%	7.3488
Pr305t	401	317	317	±0	0%	8.132
Pr306t	728	646	646	±0	0%	7.3136
Pr307t	802	632	632	±0	0%	7.2419
Pr308t	793	670	670	±1.449	0%	7.4389
Pr309t	681	636	636	±0	0%	8.145
Pr310t	738	622	622	±0	0%	7.5719
Pr311t	997	847	847	±2.529	0%	8.1331
Pr312t	945	861	861	±8.221	0%	7.4892
Pr313t	944	782	782	±0	0%	7.7299
Pr314t	906	850	850	±0	0%	7.9579
Pr315t	1062	971	976	±0.483	0.51%	7.4949

Tabla B.11: Instancias de $n = 15$ y $T = 3dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr301	383	288	292	± 0.316	1.39 %	9.3458
Pr302	292	252	252	± 0	0 %	9.8316
Pr303	309	278	278	± 1.264	0 %	9.4594
Pr304	322	281	283	± 3.062	0.71 %	9.2824
Pr305	402	306	306	± 1.563	0 %	9.0776
Pr306	674	576	576	± 0	0 %	8.4876
Pr307	622	545	545	± 0.316	0 %	8.8552
Pr308	764	592	599	± 16.844	1.18 %	8.6786
Pr309	672	595	595	± 0.483	0 %	8.6658
Pr310	638	585	585	± 0.632	0 %	8.5246
Pr311	869	770	789	± 0.483	2.47 %	9.2642
Pr312	741	712	712	± 4.131	0 %	9.3162
Pr313	821	716	724	± 7.905	1.12 %	9.2736
Pr314	860	778	786	± 2.685	1.03 %	9.2334
Pr315	888	777	806	± 0.674	3.73 %	9.1584
Pr301t	389	300	300	± 0	0 %	8.5016
Pr302t	299	271	271	± 0	0 %	9.0956
Pr303t	375	302	302	± 4.19	0 %	8.5816
Pr304t	314	310	310	± 0.63	0 %	8.3698
Pr305t	368	309	309	± 0	0 %	8.5186
Pr306t	639	610	610	± 0	0 %	8.114
Pr307t	626	601	601	± 0.948	0 %	8.0748
Pr308t	734	606	606	± 0	0 %	8.2576
Pr309t	672	595	595	± 0.421	0 %	8.7848
Pr310t	702	595	595	± 0.567	0 %	8.6626
Pr311t	867	767	767	± 0	0 %	9.199
Pr312t	886	813	813	± 0.567	0 %	8.3098
Pr313t	900	759	759	± 0	0 %	8.9174
Pr314t	944	836	836	± 0	0 %	8.6842
Pr315t	936	911	913	± 1.663	0.22 %	7.9666

Tabla B.12: Instancias de $n = 15$ y $T = 4dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	Tiempo de ejecución seg.	GAP
Pr301	316	278	284	± 1.032	25.4733	2.16 %
Pr302	278	256	256	± 0	26.3622	0 %
Pr303	283	262	262	± 0	25.8813	0 %
Pr304	313	264	264	± 3.852	25.7244	0 %
Pr305	357	288	288	± 1.1	25.3556	0 %
Pr306	628	562	562	± 0	23.7137	0 %
Pr307	574	513	513	± 0	24.1502	0 %
Pr308	720	574	576	± 1.475	23.854	0.35 %
Pr309	604	563	563	± 2.108	23.7582	0 %
Pr310	601	545	546	± 5.287	23.3083	0.18 %
Pr311	828	704	704	± 0	25.0533	0 %
Pr312	794	679	679	± 1.054	25.6787	0 %
Pr313	781	681	681	± 5.692	25.547	0 %
Pr314	757	749	749	± 0.421	25.0093	0 %
Pr315	816	757	757	± 4.377	25.8868	0 %
Pr301t	343	280	280	± 1.264	23.5971	0 %
Pr302t	292	252	252	± 0	24.7501	0 %
Pr303t	298	278	278	± 0	24.1087	0 %
Pr304t	348	286	286	± 2.25	23.0343	0 %
Pr305t	360	288	288	± 2.097	24.6725	0 %
Pr306t	635	583	583	± 0	23.4823	0 %
Pr307t	585	527	527	± 0	23.6187	0 %
Pr308t	686	572	572	± 2.213	23.6175	0 %
Pr309t	604	563	568	± 0	24.8827	0.89 %
Pr310t	615	560	560	± 0.421	24.0738	0 %
Pr311t	788	746	746	± 3.496	24.9563	0 %
Pr312t	734	706	706	± 7.728	23.595	0 %
Pr313t	753	700	700	± 6.324	23.8852	0 %
Pr314t	800	758	758	± 1.264	23.5815	0 %
Pr315t	868	783	783	± 0	23.0296	0 %

Tabla B.13: Instancias de $n = 20$ y $T = 2.25dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr401	509	473	473	± 4.685	0 %	16.0238
Pr402	630	498	498	± 0.843	0 %	14.9101
Pr403	603	516	516	± 6.325	0 %	15.6091
Pr404	578	460	466	± 3.098	1.30 %	15.3779
Pr405	583	459	465	± 2.529	1.31 %	15.6124
Pr406	1287	1088	1108	± 7.857	1.84 %	13.9247
Pr407	1467	1254	1254	± 1.449	0 %	13.3639
Pr408	1367	1075	1075	± 3.478	0 %	14.2367
Pr409	1170	1083	1083	± 2.538	0 %	14.1425
Pr410	1471	1202	1206	± 16.531	0.33 %	12.6436
Pr411	2001	1530	1550	± 19.368	1.31 %	13.5857
Pr412	1721	1305	1345	± 9.568	3.07 %	14.2758
Pr413	1842	1478	1478	± 0.316	0 %	13.7016
Pr414	1620	1313	1313	± 48.02	0 %	15.2885
Pr415	1731	1456	1456	± 33.166	0 %	15.2504
Pr401t	600	539	539	± 1.686	0 %	13.7927
Pr402t	797	641	641	± 3.794	0 %	12.0058
Pr403t	807	592	592	± 0	0 %	13.3576
Pr404t	589	547	547	± 0	0 %	13.4508
Pr405t	647	553	553	± 1.449	0 %	12.9591
Pr406t	1424	1223	1232	± 14.619	0.74 %	12.8451
Pr408t	1560	1173	1190	± 2.282	1.45 %	12.7215
Pr409t	1312	1136	1136	± 0.567	0 %	13.3075
Pr410t	1599	1494	1494	± 3.496	0 %	11.38
Pr411t	2091	1781	1840	± 31.826	3.31 %	12.3648
Pr412t	1639	1462	1462	± 0	0 %	13.3773
Pr413t	2475	1628	1628	± 3.098	0 %	12.5014

Tabla B.14: Instancias de $n = 20$ y $T = 2.5dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr401	510	461	463	± 1.523	0.43 %	17.1716
Pr402	599	479	479	± 3.33	0 %	16.1685
Pr403	578	506	506	± 3.247	0 %	16.6592
Pr404	586	457	462	± 2.366	1.09 %	16.5574
Pr405	545	453	453	± 2.078	0 %	16.9004
Pr406	1255	1003	1022	± 0.527	1.89 %	14.7983
Pr407	1203	1128	1128	± 0	0 %	14.3227
Pr408	1185	1010	1011	± 1.897	0.10 %	15.2825
Pr409	1208	1022	1022	± 1.897	0 %	15.3202
Pr410	1365	1099	1099	± 4.498	0 %	13.9537
Pr411	1871	1441	1441	± 24.87	0 %	14.9818
Pr412	1580	1275	1304	± 28.639	2.27 %	15.7429
Pr413	1688	1439	1457	± 11.229	1.25 %	14.7741
Pr415	1644	1382	1382	± 14.719	0 %	15.6253
Pr401t	557	514	514	± 2.668	0 %	15.2342
Pr402t	724	566	566	± 2.108	0 %	13.2282
Pr403t	616	549	549	± 0.816	0 %	14.3468
Pr404t	565	488	488	± 0	0 %	14.3019
Pr405t	512	488	488	± 1.932	0 %	14.375
Pr406t	1300	1088	1088	± 8.654	0 %	14.1744
Pr408t	1373	1064	1064	± 0	0 %	13.877
Pr409t	1170	1083	1083	± 1.264	0 %	14.6005
Pr410t	1403	1242	1247	± 12.995	0.40 %	12.4739
Pr411t	1889	1589	1589	± 1.032	0 %	13.7124
Pr412t	1654	1385	1385	± 26.134	0 %	14.6053
Pr415t	1640	1556	1556	± 0	0 %	13.875

Tabla B.15: Instancias de $n = 20$ y $T = 3dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr402	580	464	464	± 9.72	0 %	17.6337
Pr404	506	446	446	± 5.375	0 %	18.2555
Pr405	499	445	445	± 0.632	0 %	18.2988
Pr406	1034	997	997	± 7.125	0 %	16.4952
Pr407	1144	1038	1044	± 4.709	0.58 %	16.1016
Pr409	1038	951	952	± 7.199	0.11 %	16.9266
Pr410	1232	1041	1041	± 2.633	0 %	15.579
Pr411	1468	1346	1363	± 8.263	1.26 %	16.9397
Pr415	1538	1353	1394	± 11.133	3.03 %	17.2199
Pr401t	516	463	463	± 4.989	0 %	16.7953
Pr402t	637	521	521	± 1.229	0 %	15.064
Pr403t	589	514	514	± 9.727	0 %	15.9138
Pr404t	513	493	493	± 1.286	0 %	16.0683
Pr405t	510	495	495	± 0.421	0 %	16.0698
Pr406t	1109	1007	1007	± 6.48	0 %	16.1834
Pr407t	1292	1099	1099	± 2.869	0 %	15.2202
Pr408t	1183	1010	1010	± 7.23	0 %	15.6237
Pr409t	1059	971	971	± 11.239	0 %	16.227
Pr410t	1389	1097	1097	± 7.355	0 %	14.2847
Pr411t	1790	1434	1434	± 2.716	0 %	15.4832
Pr412t	1549	1326	1327	± 6.696	0.08 %	16.4302
Pr413t	1486	1399	1402	± 2.496	0.21 %	16.2394
Pr414t	1513	1370	1370	± 8.854	0 %	14.8733
Pr415t	1624	1439	1440	± 7.978	0.07 %	15.892

Tabla B.16: Instancias de $n = 20$ y $T = 4dm$

Instancia	V.C.	Óptimo	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr401	477	434	438	± 2.17	0.92 %	20.4855
Pr404	511	434	437	± 4.939	0.69 %	20.1244
Pr405	456	433	433	± 1.852	0 %	20.3191
Pr406	988	921	921	± 3.689	0 %	18.9373
Pr401t	468	441	441	± 3.301	0 %	18.9598

APÉNDICE C

C.1 RESULTADOS DEL PROCEDIMIENTO METAHEURÍSTICO EN COMPARACIÓN CON EL VECINO MÁS CERCANO

En esta sección se muestran los resultados obtenidos por el procedimiento metaheurístico propuesto, para cada una de las instancias en las que no se conoce la solución óptima. Esta experimentación abarca tamaños de instancias de 20, 30, 40 y 50 tareas. Los resultados están clasificados por tablas en base a su número de tareas a secuenciar (n) y al factor utilizado para calcular el tiempo entre mantenimientos (Xdm). La primera columna corresponde al nombre de la instancia; la columna dos representa el valor de la función objetivo de la solución obtenida por el algoritmo del vecino más cercano; la columna tres contiene el valor de la función objetivo para la solución encontrada por el procedimiento metaheurístico (GRASP+P.P.), en la columna cuatro se muestra la desviación estándar de las soluciones encontradas al realizar 10 réplicas del algoritmo para cada instancia; la columna cinco muestra el GAP de la solución encontrada con respecto a la obtenida por el algoritmo del vecino más cercano; y por último la columna seis corresponde a los tiempos promedio de ejecución en segundos para la metodología propuesta.

Tabla C.1: Instancias de $n = 20$ y $T = 2.25dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr407t.txt	1614	1376	± 0.674	-14.75 %	11.9779
Pr414t.txt	1847	1692	± 0	-8.39 %	12.3893
Pr415t.txt	1949	1695	± 2.394	-13.03 %	12.9165

Tabla C.2: Instancias de $n = 20$ y $T = 2.5dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr414.txt	1726	1312	± 12.363	-23.99 %	16.2146
Pr407t.txt	1465	1254	± 1.686	-14.40 %	13.7897
Pr413t.txt	1791	1541	± 10.583	-13.96 %	13.8989
Pr414t.txt	1799	1552	± 6.957	-13.73 %	13.3735

Tabla C.3: Instancias de $n = 20$ y $T = 3dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr401.txt	455	442	± 4.576	-2.86 %	18.7619
Pr403.txt	550	486	± 5.642	-11.64 %	18.0408
Pr408.txt	1183	951	± 2.869	-19.61 %	16.8289
Pr412.txt	1536	1292	± 8.969	-15.89 %	17.4221
Pr413.txt	1599	1349	± 6.929	-15.63 %	16.6367
Pr414.txt	1347	1272	± 6.924	-5.57 %	17.6221

Tabla C.4: Instancias de $n = 20$ y $T = 4dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr402.txt	563	447	± 7.8	-20.60 %	19.8613
Pr403.txt	502	467	± 3.314	-6.97 %	20.015
Pr407.txt	1046	979	± 7.789	-6.41 %	18.6014
Pr408.txt	996	899	± 5.425	-9.74 %	19.1407
Pr409.txt	1072	921	± 10.989	-14.09 %	19.0436
Pr410.txt	1036	965	± 2.547	-6.85 %	17.8254
Pr411.txt	1385	1277	± 12.251	-7.80 %	19.0852
Pr412.txt	1305	1219	± 14.796	-6.59 %	19.6016
Pr413.txt	1421	1279	± 9.237	-9.99 %	19.0704
Pr414.txt	1279	1195	± 14.377	-6.57 %	19.4411
Pr415.txt	1457	1291	± 16.138	-11.39 %	19.2101
Pr402t.txt	591	479	± 3.552	-18.95 %	17.3174
Pr403t.txt	550	485	± 2.16	-11.82 %	18.3056
Pr404t.txt	470	455	± 1.074	-3.19 %	18.3479
Pr405t.txt	476	449	± 0	-5.67 %	18.5667
Pr406t.txt	988	952	± 0.516	-3.64 %	18.9011
Pr407t.txt	1040	1001	± 0	-3.75 %	17.6005
Pr408t.txt	1106	926	± 2.366	-16.27 %	18.1315
Pr409t.txt	1052	933	± 11.508	-11.31 %	18.5626
Pr410t.txt	1076	1020	± 3.084	-5.20 %	16.9463
Pr411t.txt	1408	1345	± 3.949	-4.47 %	18.203
Pr412t.txt	1425	1247	± 5.211	-12.49 %	18.5789
Pr413t.txt	1464	1268	± 16.196	-13.39 %	18.4859
Pr414t.txt	1387	1258	± 17.373	-9.30 %	17.1616
Pr415t.txt	1433	1321	± 8.934	-7.82 %	17.8736

Tabla C.5: Instancias de $n = 30$ y $T = 2.25dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr601t	1317	1199	± 6.254	-8.96 %	34.7265
Pr602t	1498	1162	± 4.472	-22.43 %	32.5927
Pr603t	1374	1194	± 8.54	-13.10 %	35.0782
Pr604t	1539	1248	± 3.259	-18.91 %	31.1758
Pr605t	1351	1219	± 4.033	-9.77 %	33.105
Pr606t	2687	2399	± 15.237	-10.72 %	30.7629
Pr607t	2637	2451	± 7.208	-7.05 %	32.0205
Pr608t	3063	2590	± 15.652	-15.44 %	30.6872
Pr609t	3092	2514	± 8.002	-18.69 %	31.2321
Pr610t	3300	2785	± 0	-15.61 %	29.7112
Pr611t	3849	3571	± 14.88	-7.22 %	31.5345
Pr612t	4114	3374	± 7.118	-17.99 %	30.969
Pr613t	4576	3792	± 44.33	-17.13 %	30.7258
Pr614t	4438	3694	± 22.842	-16.76 %	30.8499
Pr615t	3731	3219	± 28.897	-13.72 %	33.1255

Tabla C.6: Instancias de $n = 30$ y $T = 2.5dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr601t	1292	1136	± 7.379	-12.07 %	39.1214
Pr602t	1461	1184	± 4.618	-18.96 %	34.7242
Pr603t	1315	1135	± 6.186	-13.69 %	38.3065
Pr604t	1380	1136	± 3.979	-17.68 %	35.0428
Pr605t	1315	1144	± 2.668	-13.00 %	37.2337
Pr606t	2736	2565	± 6.39	-6.25 %	32.751
Pr607t	2780	2573	± 9.663	-7.45 %	33.1735
Pr608t	2700	2332	± 9.243	-13.63 %	35.0502
Pr609t	2816	2559	± 7.177	-9.13 %	32.199
Pr610t	2969	2510	± 2.394	-15.46 %	33.5144
Pr611t	3459	3223	± 3.359	-6.82 %	34.7487
Pr612t	3622	3180	± 4.877	-12.20 %	34.4137
Pr613t	3904	3384	± 0	-13.32 %	34.5097
Pr614t	3981	3403	± 6.19	-14.52 %	34.4827
Pr615t	3612	3061	± 5.777	-15.25 %	36.4635

Tabla C.7: Instancias de $n = 30$ y $T = 3dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr601t	1151	1046	± 4.808	-9.12 %	44.2276
Pr602t	1279	1048	± 8.175	-18.06 %	41.1679
Pr603t	1116	1052	± 4.575	-5.73 %	43.5819
Pr604t	1285	1119	± 1.567	-12.92 %	40.9392
Pr605t	1105	1053	± 3.794	-4.71 %	43.1954
Pr606t	2524	2228	± 1.646	-11.73 %	37.9827
Pr607t	2644	2168	± 9.338	-18.00 %	38.3834
Pr608t	2413	2263	± 7.264	-6.22 %	39.1152
Pr609t	2434	2163	± 9.787	-11.13 %	37.8184
Pr610t	2548	2288	± 17.531	-10.20 %	38.3049
Pr611t	3311	2975	± 14.563	-10.15 %	40.5412
Pr612t	3247	2898	± 5.05	-10.75 %	39.5183
Pr613t	3290	3042	± 23.383	-7.54 %	39.4339
Pr614t	3422	3001	± 22.225	-12.30 %	39.0443
Pr615t	3100	2801	± 2.99	-9.65 %	42.598

Tabla C.8: Instancias de $n = 30$ y $T = 4dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr601t	1057	1003	± 9.094	-5.11 %	51.088
Pr602t	1172	995	± 5.578	-15.10 %	49.741
Pr603t	1082	992	± 7.691	-8.32 %	51.5922
Pr604t	1176	1008	± 4.195	-14.29 %	47.732
Pr605t	1094	999	± 4.55	-8.68 %	51.716
Pr606t	2187	2118	± 5.399	-3.16 %	45.419
Pr607t	2215	2124	± 4.332	-4.11 %	45.5538
Pr608t	2268	2046	± 7.663	-9.79 %	45.9692
Pr609t	2335	2121	± 1.686	-9.16 %	46.6878
Pr610t	2448	2126	± 12.804	-13.15 %	45.6138
Pr611t	2901	2762	± 25.148	-4.79 %	47.8562
Pr612t	2918	2745	± 38.33	-5.93 %	46.3784
Pr613t	3148	2829	± 27.414	-10.13 %	46.8649
Pr614t	3430	2857	± 17.442	-16.71 %	45.811
Pr615t	2736	2652	± 9.854	-3.07 %	49.9233

Tabla C.9: Instancias de $n = 40$ y $T = 2.25dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr701t	2725	2260	± 12.937	-17.06 %	60.4536
Pr702t	2553	2216	± 6.111	-13.20 %	60.8043
Pr703t	2432	2084	± 16.868	-14.31 %	61.8321
Pr704t	2824	2284	± 11.296	-19.12 %	58.3732
Pr705t	2567	2239	± 7.936	-12.78 %	60.0527
Pr706t	5511	4600	± 3.627	-16.53 %	57.6031
Pr707t	5247	4651	± 23.354	-11.36 %	57.0349
Pr708t	5005	4377	± 29.207	-12.55 %	58.3085
Pr709t	5026	4354	± 19.374	-13.37 %	59.0341
Pr710t	5066	4242	± 18.656	-16.27 %	57.7973
Pr711t	7700	6017	± 15.613	-21.86 %	60.1014
Pr712t	5958	5608	± 22.349	-5.87 %	63.0075
Pr713t	7727	5988	± 34.511	-22.51 %	57.0686
Pr714t	7226	6125	± 28.371	-15.24 %	59.2767
Pr715t	6890	5992	± 31.201	-13.03 %	60.1397

Tabla C.10: Instancias de $n = 40$ y $T = 2.5dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr701t	2657	2186	± 5.229	-17.73 %	63.5074
Pr702t	2402	1994	± 4.945	-16.99 %	63.5749
Pr703t	2437	2122	± 10.154	-12.93 %	64.1827
Pr704t	2527	2128	± 1.828	-15.79 %	62.3455
Pr705t	2491	1976	± 11.954	-20.67 %	63.1357
Pr706t	4955	4447	± 8.031	-10.25 %	60.1045
Pr707t	4688	4418	± 16.877	-5.76 %	59.9064
Pr708t	5177	4591	± 13.306	-11.32 %	60.8507
Pr709t	4829	4589	± 2.368	-4.97 %	62.2375
Pr710t	4717	4404	± 12.395	-6.64 %	60.7177
Pr711t	6268	5619	± 1.251	-10.35 %	65.5013
Pr712t	6009	5241	± 18.573	-12.78 %	69.4381
Pr713t	7019	5412	± 41.117	-22.89 %	61.3184
Pr714t	6904	5567	± 51.368	-19.37 %	61.243
Pr715t	6225	5465	± 32.852	-12.21 %	65.7016

Tabla C.11: Instancias de $n = 40$ y $T = 3dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr701t	2181	1976	± 5.274	-9.40 %	79.244
Pr702t	2122	1983	± 4.917	-6.55 %	80.7256
Pr703t	2153	1875	± 11.539	-12.91 %	81.4648
Pr704t	2200	2005	± 5.072	-8.86 %	80.4253
Pr705t	2062	1991	± 3.591	-3.44 %	82.487
Pr706t	4410	4298	± 3.56	-2.54 %	74.4512
Pr707t	4390	4286	± 4.175	-2.37 %	73.7904
Pr708t	4393	3857	± 4.715	-12.20 %	77.0589
Pr709t	4090	3946	± 3.119	-3.52 %	76.7795
Pr710t	4413	4280	± 2.806	-3.01 %	74.3562
Pr711t	5839	5156	± 24.801	-11.70 %	79.844
Pr712t	5434	4869	± 10.211	-10.40 %	88.223
Pr713t	5563	5259	± 5.417	-5.46 %	79.53
Pr714t	5936	5243	± 17.256	-11.67 %	78.6823
Pr715t	5539	5140	± 24.22	-7.20 %	81.3931

Tabla C.12: Instancias de $n = 40$ y $T = 4dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr701t	2145	1826	± 7.809	-14.87 %	96.9072
Pr702t	2003	1787	± 6.081	-10.78 %	96.1494
Pr703t	1947	1781	± 8.143	-8.53 %	99.9749
Pr704t	1928	1806	± 3.198	-6.33 %	96.2618
Pr705t	1880	1792	± 3.414	-4.68 %	95.6478
Pr706t	4177	3648	± 11.605	-12.66 %	87.7759
Pr707t	4018	3678	± 11.075	-8.46 %	86.9201
Pr708t	3996	3792	± 10.84	-5.11 %	91.1106
Pr709t	3874	3788	± 2.282	-2.22 %	92.3921
Pr710t	4095	3631	± 14.715	-11.33 %	88.4799
Pr711t	5122	4852	± 15.998	-5.27 %	97.1337
Pr712t	4946	4612	± 51.837	-6.75 %	104.2337
Pr713t	5391	4758	± 34.054	-11.74 %	92.2171
Pr714t	5423	4776	± 29.907	-11.93 %	92.4912
Pr715t	5481	4829	± 23.542	-11.90 %	97.422

Tabla C.13: Instancias de $n = 50$ y $T = 2.25dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr501	3305	3108	±6.36	-5.96 %	131.2045
Pr502	3300	2855	±12.311	-13.48 %	144.9885
Pr503	3451	2975	±9.045	-13.79 %	127.1066
Pr504	3305	2997	±9.442	-9.32 %	131.0027
Pr505	3250	2950	±10.027	-9.23 %	142.8331
Pr506	3279	1395	±9.721	-57.46 %	155.7371
Pr507	2538	1219	±0.516	-51.97 %	182.4319
Pr508	3309	1325	±22.005	-59.96 %	178.951
Pr509	4594	1332	±12.594	-71.01 %	179.1633
Pr510	4180	1252	±8.407	-70.05 %	169.2936
Pr511	9297	7878	±77.671	-15.26 %	124.5201
Pr512	8944	7885	±48.602	-11.84 %	135.1952
Pr513	8777	7580	±26.391	-13.64 %	129.5793
Pr514	8865	7855	±60.657	-11.39 %	131.0523
Pr515	8429	7542	±87.73	-10.52 %	140.5522
Pr501t	3610	3225	±15.273	-10.66 %	100.7364
Pr502t	3507	3102	±11.951	-11.55 %	114.493
Pr503t	4295	3661	±7.261	-14.76 %	98.6258
Pr504t	4148	3479	±12.138	-16.13 %	106.7933
Pr505t	3701	3351	±15.4	-9.46 %	114.5585
Pr506t	7463	2359	±90.691	-68.39 %	115.1436
Pr507t	7454	2510	±78.203	-66.33 %	115.2192
Pr508t	7459	2454	±79.023	-67.10 %	113.3711
Pr509t	7852	2385	±68.791	-69.63 %	121.9465
Pr510t	7465	2666	±93.249	-64.29 %	113.1413
Pr511t	10628	9091	±26.411	-14.46 %	103.7324
Pr512t	11569	9299	±72.8	-19.62 %	101.0528
Pr513t	10043	9189	±38.762	-8.50 %	101.9892
Pr514t	11406	9402	±49.333	-17.57 %	100.9848
Pr515t	10435	8864	±35.115	-15.06 %	107.2852

Tabla C.14: Instancias de $n = 50$ y $T = 2.5dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr501	3179	3025	±4.557	-4.84 %	139.7549
Pr502	3237	2827	±10.466	-12.67 %	154.545
Pr503	3359	2898	±15.557	-13.72 %	137.993
Pr504	3131	2904	±11.908	-7.25 %	144.6833
Pr505	3262	2885	±13.356	-11.56 %	153.0566
Pr506	4860	1339	±7.828	-72.45 %	172.8901
Pr507	2513	1158	±5.165	-53.92 %	200.4084
Pr508	3533	1249	±11.454	-64.65 %	193.2939
Pr509	3404	1241	±20.156	-63.54 %	197.8843
Pr510	3899	1196	±39.821	-69.33 %	186.6242
Pr511	8603	7750	±27.595	-9.92 %	134.9504
Pr512	8775	7661	±101.847	-12.70 %	145.3485
Pr513	8161	7495	±21.432	-8.16 %	140.9733
Pr514	8270	7732	±37.426	-6.51 %	141.0176
Pr515	8072	7452	±84.495	-7.68 %	151.4447
Pr501t	3735	3423	±5.116	-8.35 %	104.8198
Pr502t	3565	3258	±6.019	-8.61 %	116.2678
Pr503t	3754	3421	±1.663	-8.87 %	103.6435
Pr504t	3691	3353	±4.433	-9.16 %	108.4357
Pr505t	3585	3174	±4.137	-11.46 %	127.564
Pr506t	6274	2086	±82.7	-66.75 %	128.4833
Pr507t	6351	2198	±60.681	-65.39 %	127.5037
Pr508t	6469	2180	±65.23	-66.30 %	129.376
Pr509t	6695	2057	±60.784	-69.28 %	135.8206
Pr510t	6421	2252	±57.37	-64.93 %	126.7587
Pr511t	9811	8492	±50.745	-13.44 %	111.6804
Pr512t	10507	8588	±28.813	-18.26 %	109.8721
Pr513t	9703	8236	±48.717	-15.12 %	109.7826
Pr514t	10217	8595	±56.584	-15.88 %	106.7832
Pr515t	9638	8073	±54.617	-16.24 %	114.733

Tabla C.15: Instancias de $n = 50$ y $T = 3dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr501	3087	2896	± 4.137	-6.19 %	163.4717
Pr502	3046	2750	± 7.484	-9.72 %	176.7992
Pr503	2986	2813	± 18.078	-5.79 %	162.888
Pr504	3098	2813	± 15.293	-9.20 %	167.5248
Pr505	2948	2779	± 22.07	-5.73 %	176.3142
Pr506	3791	1289	± 6.944	-66.00 %	206.6496
Pr507	2769	1088	± 2.668	-60.71 %	224.2102
Pr508	3238	1165	± 5.189	-64.02 %	222.2147
Pr509	3487	1232	± 15.191	-64.67 %	225.4284
Pr510	3576	1168	± 5.374	-67.34 %	220.0427
Pr511	8085	7602	± 96.347	-5.97 %	155.3726
Pr512	7901	7466	± 82.126	-5.51 %	169.4557
Pr513	7796	7479	± 69.946	-4.07 %	160.3691
Pr514	7784	7495	± 75.238	-3.71 %	162.2704
Pr515	8034	7441	± 53.252	-7.38 %	172.1405
Pr501t	3371	3233	± 4.299	-4.09 %	119.1758
Pr502t	3211	2888	± 7.788	-10.06 %	139.079
Pr503t	3437	3154	± 6.828	-8.23 %	128.0072
Pr504t	3229	3111	± 6.232	-3.65 %	131.5983
Pr505t	3199	2931	± 7.837	-8.38 %	150.4748
Pr506t	6016	1881	± 90.867	-68.73 %	154.8264
Pr507t	6058	1930	± 57.252	-68.14 %	157.008
Pr508t	6166	1915	± 84.836	-68.94 %	154.3926
Pr509t	6314	1936	± 59.058	-69.34 %	163.6064
Pr510t	6166	2048	± 65.232	-66.79 %	151.4744
Pr511t	8335	7865	± 12.615	-5.64 %	129.4729
Pr512t	9241	8069	± 26.626	-12.68 %	134.5759
Pr513t	8560	7823	± 27.089	-8.61 %	125.3021
Pr514t	8573	7941	± 30.873	-7.37 %	124.2612
Pr515t	9093	7835	± 11.499	-13.83 %	126.7477

Tabla C.16: Instancias de $n = 50$ y $T = 4dm$

Instancia	V.C.	GRASP+P.P.	Desv. Est.	GAP	Tiempo de ejecución seg.
Pr501	2872	2702	± 6.359	-5.92 %	194.8465
Pr502	2881	2673	± 8.124	-7.22 %	210.5699
Pr503	2903	2702	± 7.394	-6.92 %	197.258
Pr504	2903	2702	± 5.296	-6.92 %	202.5122
Pr505	2841	2691	± 7.211	-5.28 %	211.2419
Pr506	3774	1221	± 2.44	-67.65 %	233.4103
Pr507	2842	1084	± 4.089	-61.86 %	242.1633
Pr508	2611	1114	± 5.884	-57.33 %	240.7468
Pr509	2860	1161	± 7.688	-59.41 %	243.5788
Pr510	3302	1104	± 18.56	-66.57 %	240.8999
Pr511	7672	7356	± 49.796	-4.12 %	186.7894
Pr512	7560	7201	± 72.132	-4.75 %	198.4449
Pr513	7382	7233	± 29.032	-2.02 %	190.5374
Pr514	7741	7419	± 44.138	-4.16 %	190.593
Pr515	7190	7166	± 47.096	-0.33 %	201.0343
Pr501t	2975	2825	± 8.045	-5.04 %	147.8165
Pr502t	3047	2766	± 3.922	-9.22 %	171.2989
Pr503t	3110	2839	± 11.333	-8.71 %	155.0388
Pr504t	3149	2834	± 11.363	-10.00 %	158.1502
Pr505t	2951	2763	± 6.073	-6.37 %	178.5233
Pr506t	5720	1728	± 60.911	-69.79 %	184.1018
Pr507t	5736	1566	± 138.28	-72.70 %	184.1632
Pr508t	5968	1718	± 88.32	-71.21 %	182.5236
Pr509t	6033	1626	± 81.136	-73.05 %	198.6752
Pr510t	5913	1841	± 62.907	-68.87 %	183.088
Pr511t	8076	7574	± 21.893	-6.22 %	156.6499
Pr512t	8030	7569	± 35.54	-5.74 %	154.7668
Pr513t	7701	7220	± 52.019	-6.25 %	148.1167
Pr514t	8148	7578	± 29.234	-7.00 %	146.0913
Pr515t	7640	7227	± 26.248	-5.41 %	149.452

BIBLIOGRAFÍA

- Adiri, I., Frostig, E., y Kan, A. (1991). Scheduling on a single machine with a single breakdown to minimize stochastically the number of tardy jobs. *Naval Research Logistics (NRL)*, 38(2):261–271.
- Akturk, M. S., Ghosh, J. B., y Gunes, E. D. (2003). Scheduling with tool changes to minimize total completion time: a study of heuristics and their performance. *Naval Research Logistics (NRL)*, 50(1):15–30.
- Akturk, M. S., Ghosh, J. B., y Gunes, E. D. (2004). Scheduling with tool changes to minimize total completion time: basic results and spt performance. *European Journal of Operational Research*, 157(3):784–790.
- Allahverdi, A. y Soroush, H. (2008). The significance of reducing setup times/setup costs. *European Journal of Operational Research*, 187(3):978–984.
- Ángel-Bello, F., Álvarez, A., Pacheco, J., y Martínez, I. (2011a). A single machine scheduling problem with availability constraints and sequence-dependent setup costs. *Applied Mathematical Modelling*, 35(4):2041 – 2050.
- Ángel-Bello, F., Álvarez, A., Pacheco, J., y Martínez, I. (2011b). A heuristic approach for a scheduling problem with periodic maintenance and sequence-dependent setup times. *Computers & Mathematics with Applications*, 61(4):797–808.
- Anghinolfi, D. y Paolucci, M. (2009). A new ant colony optimization approach for the single machine total weighted tardiness scheduling problem.

- Baker, K. R. (1974). Introduction to sequencing and scheduling. *John Wiley and Sons*.
- Birge, J., Frenk, J., Mittenthal, J., y Kan, A. (1990). Single-machine scheduling subject to stochastic breakdowns. *Naval Research Logistics (NRL)*, 37(5):661–677.
- Burkard, R. E., Deineko, V. G., van Dal, R., van, J. A. A., Veen, d., y Woeginger, G. J. (1998). Well-solvable special cases of the traveling salesman problem: A survey. *SIAM Rev.*, 40(3):496–546.
- Chen, J.-S. (2008). Optimization models for the tool change scheduling problem. *Omega*, 36(5):888–894.
- Chen, W. (2006). Minimizing total flow time in the single-machine scheduling problem with periodic maintenance. *Journal of the Operational Research Society*, 57(4):410–415.
- Chen, W. (2009a). Scheduling with dependent setups and maintenance in a textile company. *Computers & Industrial Engineering*, 57(3):867–873.
- Chen, W.-J. (2009b). Minimizing number of tardy jobs on a single machine subject to periodic maintenance. *Omega*, 37(3):591–599.
- Choi, Y.-C. y Kim, Y.-D. (2001). Tool replacement policies for a machining centre producing multiple types of products with distinct due dates. *International Journal of Production Research*, 39(5):907–921.
- Choobineh, F. F., Mohebbi, E., y Khoo, H. (2006). A multi-objective tabu search for a single-machine scheduling problem with sequence-dependent setup times. *European Journal of Operational Research*, 175(1):318–337.
- Conner, G. (2009). 10 questions. *Manufacturing Engineering Magazine*, pages 093–099.

- Cruzan, R. (2009). *Manager's Guide to Preventive Building Maintenance*. The Fairmont Press, Inc.
- Díaz, A., G. F. G. H. G. J. L. M. M. P. y. T. F. (1996). *Optimización Heurística y Redes Neuronales*. Paraninfo, Madrid.
- Dudek, R., Smith, M., y Panwalkar, S. (1974). Use of a case study in sequencing/scheduling research. *Omega*, 2(2):253 – 261.
- Ebrahimi Zade, A. y Fakhrzad, M. B. (2013). A dynamic genetic algorithm for solving a single machine scheduling problem with periodic maintenance. *ISRNI Industrial Engineering*, 2013.
- Ecker, K. H. y Gupta, J. N. (2005). Scheduling tasks on a flexible manufacturing machine to minimize tool change delays. *European Journal of Operational Research*, 164(3):627–638.
- Feo, T. A. y Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., y Kan, A. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete Mathematics*, 5:287–326.
- Guo, Q. y Tang, L. (2015). An improved scatter search algorithm for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times. *Applied Soft Computing*, (0):–.
- Gupta, S. R. y Smith, J. S. (2006). Algorithms for single machine total tardiness scheduling with sequence dependent setups. *European Journal of Operational Research*, 175(2):722–739.
- Jackson, J. R. (1955). Scheduling a production line to minimize maximum tardiness. Technical report, DTIC Document.

- Ji, M., He, Y., y Cheng, T. E. (2007). Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & operations research*, 34(6):1764–1770.
- Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R., Thatcher, J., y Bohlinger, J., editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US.
- Kirlik, G. y Oguz, C. (2012). A variable neighborhood search for minimizing total weighted tardiness with sequence dependent setup times on a single machine. *Computers & Operations Research*, 39(7):1506–1520.
- Lee, J.-Y. y Kim, Y.-D. (2012). Minimizing the number of tardy jobs in a single-machine scheduling problem with periodic maintenance. *Computers & Operations Research*, 39(9):2196–2205.
- Lee, S. M. y Asllani, A. A. (2004). Job scheduling with dual criteria and sequence-dependent setups: mathematical versus genetic programming. *Omega*, 32(2):145–153.
- Li, C.-L., Simchi-Levi, D., y Desrochers, M. (1992). On the distance constrained vehicle routing problem. *Operations research*, 40(4):790–799.
- Liao, C. y Chen, W. (2003). Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers & Operations Research*, 30(9):1335 – 1347.
- Liao, C.-J. y Juan, H.-C. (2007). An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. *Computers & Operations Research*, 34(7):1899–1909.
- Lin, S.-W. y Ying, K.-C. (2007). Solving single-machine total weighted tardiness problems with sequence-dependent setup times by meta-heuristics. *The International Journal of Advanced Manufacturing Technology*, 34(11-12):1183–1190.
- Ma, Y., Chu, C., y Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, 58(2):199–211.

- Marti, R. (2003). Procedimientos metaheurísticos en optimización combinatoria. *Matemáticas, Universidad de Valencia*, 1(1):3–62.
- Morton, T. (1993). *Heuristic scheduling systems: with applications to production systems and project management*, volume 3. John Wiley & Sons.
- Pacheco, J., Ángel-Bello, F., y Álvarez, A. (2013). A multi-start tabu search method for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. *Journal of Scheduling*, 16(6):661–673.
- Picard, J.-C. y Queyranne, M. (1978). The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110.
- Pinedo, M. y Rammouz, E. (1988). A note on stochastic scheduling on a single machine subject to breakdown and repair. *Probability in the Engineering and Informational Sciences*, 2(01):41–49.
- Qi, X., Chen, T., y Tu, F. (1999). Scheduling the maintenance on a single machine. *Journal of the Operational Research Society*, pages 1071–1078.
- Salazar-Hornig, E. y Schriels-Abreu, G. (2014). Estrategias evolutivas para la minimización del makespan en una máquina con tiempos de preparación dependientes de la secuencia. *Ingeniería, investigación y tecnología*, 15(1):1–10.
- Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66.
- Tanaka, S. y Araki, M. (2013). An exact algorithm for the single-machine total weighted tardiness problem with sequence-dependent setup times. *Computers & Operations Research*, 40(1):344–352.
- Tasgetiren, M. F., Pan, Q.-K., y Liang, Y.-C. (2009). A discrete differential evolution algorithm for the single machine total weighted tardiness problem with sequence dependent setup times. *Computers & Operations Research*, 36(6):1900–1915.

-
- Valente, J. y Alves, R. A. (2008). Beam search algorithms for the single machine total weighted tardiness scheduling problem with sequence-dependent setups. *Computers & Operations Research*, 35(7):2388–2405.
- Wang, J.-B. (2008). Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect. *Computers & Industrial Engineering*, 55(3):584–591.
- Winston, W. L. (2006). *Investigación de Operaciones Aplicaciones y Algoritmos*. THOMPSON, E.U.A., Indiana, 4 edition.

FICHA AUTOBIOGRÁFICA

Pedro Ines Loera Martínez

Candidato para el grado de Maestría en Ciencias en
Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

SECUENCIACIÓN DE TAREAS EN UNA MÁQUINA CON TIEMPOS DE PREPARACIÓN DEPENDIENTES DE LA SECUENCIA Y ETAPAS FIJAS DE MANTENIMIENTO PREVENTIVO

Nací el 18 de abril de 1989, en la ciudad de San Nicolás de los Garza, Nuevo León, siendo el segundo hijo de Pedro Loera Rosas y Santos Guadalupe Martínez Camacho. Gran parte de mi vida he vivido en el municipio de Juárez, Nuevo León. Asistí a la escuela primaria Profesor Lauro Aguirre Espinoza y realicé los estudios secundarios en la Escuela Secundaria Técnica N°68, ambas cercanas a mi localidad. En el año 2004 ingresé a la Preparatoria 8, posteriormente curse la licenciatura en la Facultad de Ingeniería Mecánica y Eléctrica, donde me titulé como Ingeniero Mecánico Administrador. Realicé prácticas profesionales en la empresa Hammond Power Solutions, donde desarrollé habilidades enfocadas a la administración del mantenimiento;

terminado el periodo de prácticas me desempeñe como diseñador en ingeniería eléctrica en la empresa Grupo Subind, donde trabajé por dos años. Posteriormente en el año 2013 ingresé a realizar mis estudios de maestría en el Posgrado de Ingeniería de Sistemas, donde tuve la oportunidad de enriquecer mis conocimientos y desarrollar habilidades en áreas de mi interés.