

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

ZONA METROPOLITANA DE LA CIUDAD DE MÉXICO

PROGRAMA DE GRADUADOS EN INGENIERÍA Y CIENCIAS

TESIS

IMPLEMENTACIÓN DE UN ALGORITMO DE RECOCIDO SIMULADO PARA EL PROBLEMA DE PLANIFICACIÓN DE TAREAS EN UNA EMPRESA DE MANUFACTURA DE COSMÉTICOS.

Aspirante: MAYRA GEORGINA MIRANDA FLORES

Asesor: Dr. EDUARDO DÍAZ SANTILLÁN

Comité de Tesis: Dr. JAIME MORA VARGAS

Dr. ALEJANDRO SANDOVAL CORREA

Dr. EDUARDO DÍAZ SANTILLÁN

Campus Estado de México

Matrícula: 1163333

Especialidad y plan: MII

Firma del aspirante:

Atizapán de Zaragoza, Edo. Méx., noviembre de 2009.

Índice General

Resumen	5
Agradecimientos	6
1. Introducción	8
1.1 Motivación para la investigación	8
1.2 Descripción del problema	10
1.3 Objetivos	12
1.4 Hipótesis	13
1.5 Justificación	14
1.6 Contribución	15
1.7 Alcances	16
2. Marco Teórico	17
2.1 La planeación de los recursos y la administración de la capacidad	17
2.2 La Planificación de Tareas.	18
2.2.1 Historia y Estado del Arte de la teoría de Planificación de tareas	19
2.3 Calidad de la solución	23
2.4 Métodos de resolución para los Problemas de Planificación de Tareas.	23
2.4.1 Métodos Exactos	24
2.4.1.1 Enumeración Exhaustiva	24
2.4.1.2 Técnicas de Programación Matemática	25
2.4.1.3 Método de Ramificación y Acotamiento.	26
2.4.2 Métodos Heurísticos.	27
2.4.2.1 Reglas de despacho	27

2.4.2.2	Heurísticas basadas en cuellos de botella	28
2.4.2.3	Métodos Metaheurísticos de Búsqueda Local	28
2.4.2.4	Recocido Simulado	30
2.4.2.5	Búsqueda Tabú	33
2.4.2.6	Cómputo Evolutivo	34
2.4.2.7	Algoritmos Genéticos	36
2.5	La complejidad computacional del Job Shop Scheduling Problem.	37
3.	Formulación del problema y descripción de la metodología.	41
3.1	Panorama general de la Industria de Cosméticos en México.	41
3.2	Situación de la compañía	43
3.3	Formulación del problema	46
3.4	Distribución de planta.	49
3.5	Descripción de la metodología	50
3.5.1	Uso del algoritmo de recocido simulado en el método propuesto.	51
3.5.1.1	Creación de la solución inicial.	51
3.5.1.2	Creación de soluciones “vecinas”	52
3.5.1.3	Función de evaluación	53
3.5.1.4	Diagrama de Flujo	54
3.5.2	Elección de factores y niveles.	55
3.5.3	Variable de respuesta.	56
3.6	Instancias	57
4.	Discusión de resultados.	58
4.1	Resultados obtenidos	58
4.2	Pruebas de hipótesis	64
4.2.1	Prueba de hipótesis de tiempo total de ejecución del método.	64

4.2.2 Prueba de hipótesis de tiempo total de proceso obtenido.	65
4.3 Análisis de capacidad del método propuesto	67
5. Conclusiones y trabajo futuro.	68
5.1 Conclusiones	68
5.2 Trabajo futuro.	69
Anexo 1	70
Bibliografía	74

Resumen.

La creciente competencia que enfrenta la industria mexicana de cosméticos frente a las empresas trasnacionales hace necesaria una adecuada administración de los recursos a fin de mejorar la competitividad y responder con mayor eficiencia a las condiciones del mercado y los requerimientos de los clientes.

La Planificación de tareas es una herramienta de apoyo en la administración de los recursos y facilita la toma de decisiones dentro de un ambiente de manufactura mediante la generación de secuencias de producción factibles que deriven en un calendario de producción en el que los diferentes recursos disponibles sean asignados con el objetivo de satisfacer la demanda del cliente en tiempo y forma.

Una de las características principales del problema de planificación de tareas radica en la complejidad para su resolución óptima que en instancias grandes presenta. Debido a esto, se han desarrollado múltiples metodologías para aproximarse a una solución, entre las que destacan los métodos heurísticos y metaheurísticos, que si bien no obtienen la solución óptima si obtienen una muy buena en un tiempo relativamente corto.

Dentro de los métodos metaheurísticos se encuentra el recocido simulado, el cuál, inspirado en principios de la física y la ciencia de los materiales ha demostrado ser efectivo en la resolución de problemas de planificación de tareas, teniendo entre sus ventajas la de poder escapar de óptimos locales así como la sencillez en su programación.

El presente trabajo aborda el problema de planificación de tareas mediante un método que utiliza un algoritmo de recocido simulado que nos permite obtener secuencias factibles de producción que minimicen el tiempo total de ejecución de todos los trabajos dentro del ambiente de un taller de manufactura de cosméticos.

También se planteó como objetivo reducir el tiempo de obtención de estas secuencias de producción, optimizando con esto el proceso de toma de decisiones en el área de planeación de la producción así como el uso de los recursos disponibles en la planta; todo esto sin necesidad de aumentar la capacidad mediante inversiones en maquinaria, equipo o programas de cómputo comerciales.

Agradecimientos

A mis abuelos porque con ellos comenzó esta historia. Especialmente a mi abuelita Cristina Hernández Reyes por toda una vida de amor y trabajo; este triunfo también es fruto de su esfuerzo.

A mis papás Georgina Flores y Liborio Miranda cuyo inconmensurable amor y grandes esfuerzos nunca terminaré de agradecer.

A mis hermanos Elizabeth, Carlos y Daniela por estar siempre conmigo y quererme incondicionalmente.

A Aquilio por el amor, el dolor y la enseñanza. Por todo este camino recorrido juntos, pero sobre todo por el que falta.

A mis queridas tías Juana y Sara, así como a Sol, Chucho, Lalo, Tania y Lety por todo el cariño y por esperar siempre lo mejor de mí.

A todos los amigos, presentes y pasados que mediante buenas y malas experiencias me han enseñado acerca de la naturaleza humana. A Gaby Blanco y a Nancy Rodríguez por enseñarme tantas cosas sobre la amistad, la lealtad y el perdón. A Lupita Rodríguez, con admiración por su fortaleza de espíritu. A Gabriel Zúñiga por años de hermosa amistad. A Fabián Ávila por el inmenso cariño. A Daniel López con mucha gratitud, por la amistad recién encontrada, el apoyo y la paciencia.

A todos los profesores en mi vida por todo lo que han sembrado en mí, especialmente a Lucy Valencia por el ejemplo y el impulso cuya inercia aún continúa.

Al Instituto Tecnológico y de Estudios Superiores de Monterrey por darme la oportunidad de pertenecer a su programa de maestría en Ingeniería Industrial y por el apoyo financiero otorgado. A Eduardo Díaz, mi asesor de tesis por sus enseñanzas, consejos, ideales y buen humor así como su orientación a lo largo de este trabajo de investigación.

*Un guerrero de la luz nunca olvida la gratitud.
Durante la lucha, fue ayudado por los ángeles; las fuerzas
celestiales colocaron cada cosa en su lugar, y permitieron
que él pudiera dar lo mejor de sí.*

*Los compañeros comentan: "¡Qué suerte tiene!" Y el guerrero a
veces consigue mucho más de lo que su capacidad permite.
Por eso, cuando el sol se pone, se arrodilla y agradece el
Manto Protector que lo rodea.*

*Su gratitud, no obstante, no se limita al mundo espiritual;
él jamás olvida a sus amigos, porque la sangre de ellos se
mezcló con la suya en el campo de batalla.*

*Un guerrero no necesita que nadie le recuerde la ayuda de
los otros, él se acuerda solo, y reparte con ellos la
recompensa."*

Extracto del "Manual del Guerrero de la luz". P.Coelho.

Capítulo 1

Introducción

1.1 Motivación para la investigación

La apertura económica provocada por la globalización ha creado en las empresas la necesidad de adaptación a nuevas reglas de comercio internacional y nuevos estándares de calidad por lo que una rivalidad en el sector industrial ha crecido dramáticamente en los últimos años [51].

En el mercado mexicano se ha incrementado el consumo de productos cosméticos y de cuidado personal [10] debido a un cambio de actitud donde los consumidores han dejado de considerarlos suntuarios, viéndolos ahora como elementos indispensables y de uso diario. El consumidor mexicano se ha vuelto también más exigente, por lo que las tendencias se dirigen hacia el desarrollo de productos más sofisticados, lo que incrementa la presión entre las empresas por desarrollar nuevas formulaciones y colocar nuevos productos en el mercado con mayor rapidez que la competencia.

Sin embargo, el mercado mexicano de cosméticos tiene algunas debilidades [10] como son los bajos niveles de integración de la cadena productiva, el desarrollo limitado de nuevos productos (la mayoría de las innovaciones provienen del extranjero), la maquinaria es obsoleta en muchos de los casos, los sistemas de calidad no están adecuadamente estandarizados para este segmento, así como las dificultades existentes para el acceso a distintos instrumentos financieros, lo que dificulta a las empresas mexicanas competir con las grandes compañías transnacionales de cosméticos.

Aunado a esto, en la industria manufacturera de cosméticos la demanda es muy variable ya que obedece a modas [10], lo que dificulta elaborar pronósticos de fabricación de estos productos y por tanto, complica las decisiones de abastecimiento de insumos para poder reaccionar a tiempo ante la demanda futura o una demanda imprevista. Para una empresa maquiladora de cosméticos, con esquema de fabricación bajo pedido, esta variabilidad de la demanda le dificulta entregar en

corto tiempo los pedidos urgentes ya que al no tener previamente inventario para la fabricación, la elaboración del producto pasa por todas las etapas de la cadena de suministro.

Debido a estos nuevos niveles de competencia, a la variabilidad e incertidumbre de la demanda, a la capacidad limitada y a la crisis económica que ha ocasionado un aumento en los costos de producción, las empresas manufactureras deben responder con mayor eficiencia a las condiciones del mercado y a los requerimientos de los clientes mediante el uso de nuevos métodos de administración de la producción con el objetivo de reducir costos, apearse a los estándares de calidad solicitados por el cliente y respetar tiempos y cantidades de entrega comprometidos con el cliente. Uno de los aspectos diferenciadores clave para los clientes lo constituye el tiempo total de la orden desde que esta es colocada hasta que se entrega así como la entrega a tiempo del producto o servicio.

La secuenciación de trabajos o planificación de tareas es una herramienta clave para maximizar el desempeño del uso de los recursos en un taller de manufactura y minimizar plazos de entrega del producto sin necesidad de invertir en infraestructura. La planificación de tareas en el ambiente de un taller de manufactura es también conocida como *Job Shop Scheduling Problem* (JSSP) y debido a que este tipo de problemas no son fáciles de resolver [19], muchos investigadores han desarrollado distintos algoritmos y metodologías con el objetivo de obtener buenas soluciones; entre estos métodos se encuentran los métodos exactos y los métodos heurísticos.

Debido a la complejidad que en instancias grandes presenta el problema de planificación de tareas en un taller de manufactura o *Job Shop Scheduling Problem* (JSSP), no se cuenta con un método exacto para resolverlo. La enumeración de todas las posibles soluciones resulta impráctica porque el número total de posibles soluciones es $(n!)^m$ [15], siendo n el número de trabajos y m el número de máquinas y si, por ejemplo, se tiene un problema de 24 operaciones y una máquina, el espacio de búsqueda para ese problema es de $(24!) = 6.20 \times 10^{23}$ posibles soluciones, lo que llevaría aproximadamente 20 mil millones de años en resolver si una computadora pudiera procesar una solución cada μseg [17]. Una alternativa ampliamente utilizada desde hace algunos años ha sido la aplicación de métodos heurísticos y metaheurísticos para solucionar el problema de planificación de tareas. Estos métodos, si bien no garantizan la mejor solución, proporcionan una buena solución en una menor cantidad de tiempo que los

métodos exactos, obteniendo resultados en un tiempo que va desde segundos a horas dependiendo el tamaño del problema.

La complejidad del problema de planificación de tareas (en este trabajo se aborda un problema NP-difícil) y a la vez la gran utilidad de su aplicación (en líneas de producción industrial, en la elaboración de los horarios escolares, en la planificación de vuelos en los aeropuertos, etc.) [17], es lo que motivó el desarrollo de este trabajo de tesis, en el que se plantea el uso de la metaheurística conocida como Recocido Simulado para la elaboración de programas de producción en un taller de manufactura.

1.2 Descripción del problema

El problema general de planificación de tareas en la descripción de Jain y Meeran [26] consiste en un conjunto finito de trabajos $\{J_i\}_{i=1}^n$ para ser procesados en un conjunto finito M de m máquinas $\{M_k\}_{k=1}^m$.

Cada trabajo J_i consiste de una cadena de m_i operaciones $O_{i1}, O_{i2}, \dots, O_{im}$ que tienen que ser programadas en un predeterminado orden. Este orden constituye las restricciones de precedencia. La cantidad total de operaciones N está dada por $N = \sum_{i=1}^n m_i$.

Los trabajos entre sí, no tienen ninguna restricción de precedencia.

O_{ik} es la operación del trabajo J_i que debe ser procesada en la máquina M_k para un tiempo ininterrumpido de proceso τ_{ik} . No se permite interrumpir una operación sin que haya finalizado su proceso en la máquina.

Las restricciones del **Job Shop Scheduling Problem** clásico son las siguientes:

- Los trabajos son unidireccionales.
- La máquina que inicia no lo hace con el primer trabajo, cualquier máquina puede empezar o terminar.

- Cada trabajo tiene su propio patrón de flujo o ruta de trabajo a través de las máquinas, el cuál es independiente de los demás trabajos.
- No está permitido que dos operaciones del mismo trabajo se procesen simultáneamente.
- Cada máquina puede procesar solamente un trabajo a la vez así como cada trabajo puede ser procesado por solamente una máquina a la vez.
- Ninguna operación tiene prioridad sobre las demás.
- Cada trabajo es procesado hasta concluirse, aún cuando haya que esperar y retardarse entre las operaciones procesadas.
- Un trabajo puede iniciarse en cualquier momento, siempre y cuando esté disponible la máquina y no se haya especificado un tiempo de inicio para éste.
- Los trabajos tienen que esperar a que la siguiente máquina esté disponible para continuar su proceso
- Los tiempos de configuración y cambio de máquina son independientes del orden de procesamiento y están incluidos en los tiempos de procesamiento.
- Hay sólo un tipo de máquina.
- Es posible que las máquinas estén ociosas en algún momento del plan de trabajo.
- Se requiere que para cada trabajo, la secuencia de operaciones contenga exactamente una operación para ser procesada por cada una de las máquinas del problema.

Las operaciones a ser procesadas en una máquina forman la *secuencia de operaciones* para ésta. Una instancia del JSSP se define por medio de una matriz que contiene el orden de procesamiento de cada una de las operaciones. Un plan de trabajo consta de n secuencias de operaciones para cada máquina y es aquel donde cada una de las operaciones que conforman los trabajos han sido asignadas a las máquinas de tal forma que cumplan las restricciones del problema. Para visualizar el plan de trabajo se utilizan generalmente gráficas de Gantt (Fig. 1.) aunque también pueden utilizarse otras representaciones como las de cadena. En la figura 1 se visualiza un problema con 3 máquinas (m_0 , m_1 y m_2) y tres trabajos (j_0 , j_1 , y j_2) con 3 operaciones cada uno, cuyo tiempo total de terminación es de 12 unidades de tiempo.

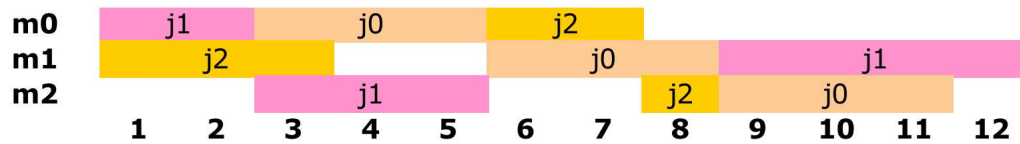


Fig1. Representación del problema en una gráfica de Gantt

El período de tiempo en el que todas las operaciones son completadas es conocido como *Makespan* [44], que significa el tiempo de terminación máximo de la última tarea en el sistema. El objetivo más comúnmente planteado en el *Job Shop Scheduling* es el de minimizar el tiempo requerido para terminar todos los trabajos (Makespan) $C_{\max}^* = \min(C_{\max})$ a la vez que satisfacer todas las restricciones. Este objetivo es equivalente a minimizar los tiempos muertos, o a maximizar la utilización de las máquinas [56]. Un Job Shop Scheduling Problem cuyo objetivo es minimizar el Makespan puede ser considerado análogo al Problema del Agente Viajero (TSP) [40].

1.3 Objetivos

El objetivo principal de la investigación es apoyar al proceso de toma de decisiones en la planeación de la producción de un taller de manufactura mediante el diseño de una metodología de apoyo para la elaboración de secuencias de operaciones.

Se pretende que las secuencias de operaciones tengan las siguientes características:

1. Ayuden a reducir las horas-hombre dedicadas a la elaboración manual de estas secuencias (entendiéndose como manual la elaboración de las secuencias sin la ayuda de software alguno), que actualmente va desde 5 minutos hasta incluso horas dependiendo del tamaño de máquinas y operaciones que deben ser programadas.
2. Indiquen cuál es el orden más apropiado en que deben ser ejecutadas las operaciones pertenecientes a los trabajos, con el fin de que el tiempo de ejecución total (*makespan*) sea el menor posible.

3. Mediante la reducción del *makespan* apoyar en el cumplimiento a la fecha de entrega de los productos acordada con el cliente.

La metodología hará uso de un algoritmo de recocido simulado para generar secuencias factibles de producción para diferentes instancias del problema de Planificación de Tareas (*Job Shop Scheduling Problem*). Se evaluarán diferentes instancias del Problema de Planificación de Tareas.

Los objetivos específicos de este trabajo de investigación son los siguientes:

- Encontrar soluciones factibles que superen el desempeño del método manual actualmente utilizado para el Problema de Planificación de Tareas.
- Reducir en la práctica el tiempo de elaboración de programas de producción mediante el uso de esta metodología.
- Encontrar la secuencia que minimice el tiempo total de procesamiento (*Makespan*) de todos los trabajos demandados para un periodo de tiempo dado, y con esto, apoyar al cumplimiento de las fechas de entrega acordadas con el cliente.
- Reducir conflictos en los recursos al momento de elaborar programas de producción.
- Indicar el orden en que las operaciones de todos los trabajos serán programadas en las máquinas y con esto establecer prioridades en la liberación de órdenes de trabajo.
- Indicar el momento en que cada una de las operaciones debe iniciarse y debe terminar.
- Ayudar a identificar períodos de tiempo disponibles para otras actividades como el mantenimiento preventivo.

1.4 Hipótesis

La aplicación de la metodología para la Planificación de Tareas en un taller de manufactura, con base en el algoritmo metaheurístico de recocido simulado, permitirá obtener secuencias factibles de producción, reducir el tiempo de generación de estas secuencias así como el tiempo total de completamiento (*makespan*) dentro del proceso de planeación de la producción .

Las preguntas de investigación son las siguientes:

1. ¿Es posible que esta metodología pueda encontrar secuencias de producción factibles, que puedan ser aplicables en el ambiente de un taller de manufactura?
2. ¿Es significativa la reducción del tiempo dedicado a elaborar programas de producción al utilizar esta metodología, comparada con la programación manual?
3. ¿Cuál es el orden más conveniente en que deben programarse las operaciones pertenecientes a todos los trabajos a fin de que éstos se finalicen al menos en la fecha acordada con el cliente?

1.5 Justificación

Para el presente caso de estudio, la Planificación de Tareas constituye una opción para obtener mejoras en el proceso de planeación de la producción sin tener que hacer una inversión significativa en equipo o infraestructura.

La importancia de este trabajo radica en que la Planificación de Tareas es un proceso vital dentro de la administración de recursos en cualquier empresa manufacturera por lo que es necesario llevarla a cabo de manera eficiente, cuidando tanto la calidad de los resultados arrojados como el tiempo dedicado a esta.

La Planificación de Tareas es una herramienta que permite administrar el uso de los recursos mediante la calendarización de eventos o tareas asociados a éstos de manera que el plan de producción obtenido se apege a los objetivos previamente establecidos que tengan que ver con cantidad producida, tiempo de entrega, presupuesto, entre otros; y, de ser posible, se busca un ahorro respecto a un estado inicial [17].

Ha sido ampliamente demostrado que el problema de Planificación de Tareas es sumamente difícil de resolver de manea óptima y que los métodos metaheurísticos constituyen una alternativa aceptable para abordar este tipo de problemas [27]. Es sabido también que el algoritmo de recocido simulado es sencillo de programar y que basta una sola corrida del programa para encontrar una buena solución que resulte similar a la arrojada por métodos más sofisticados pero

que entrañan una mayor complejidad en su programación e implementación [4]. Otra ventaja es que el recocido simulado se puede adaptar fácilmente a los nuevos problemas y, debido a su capacidad para evitar óptimos locales pobres, es apto en la obtención de muy buenos resultados [29].

Debido a lo anterior se escogió este algoritmo para utilizarlo en la metodología propuesta, fortaleciéndose así la idea de que lo declarado en la hipótesis es correcto, motivando esto a la realización del trabajo de investigación.

1.6 Contribución

El aporte del presente trabajo consiste en el diseño de una metodología de uso libre que mediante la aplicación del algoritmo de recocido simulado resuelva una amplia variedad de instancias del problema de Planificación de Tareas, reduciendo el tiempo dedicado a la obtención de secuencias válidas de producción, indicando el orden más conveniente en que deben ser procesadas las diferentes operaciones en cada una de las máquinas, favoreciendo con esto la toma de decisiones dentro del ambiente de un taller de manufactura.

La herramienta que se obtendrá como resultado de la investigación, podrá ser aplicada en las empresas de manufactura, ofreciendo la posibilidad de incrementar su productividad al reducir tanto el tiempo dedicado a elaborar programas de producción como el tiempo de ejecución de éstos, de manera que conduzca a la mejora en el uso de los recursos, permitiendo una mayor competitividad sin necesidad de inversiones en infraestructura.

La intención de esto es incentivar el uso de estas técnicas en la optimización de procesos dentro de la industria manufacturera mexicana, ya que actualmente su uso no está extendido y es una buena opción en lugar de invertir en programas de cómputo comerciales que pueden ser económicamente inaccesibles para muchas empresas mexicanas.

Esto significa una solución innovadora a un problema común en la industria manufacturera mexicana, tomando en cuenta las restricciones económicas que la mayoría de estas empresas enfrentan dada la situación del país.

1.7 Alcances

Los alcances del trabajo de investigación son:

- El desarrollo de una metodología para elaborar programas de producción en un taller de manufactura utilizando un algoritmo de recocido simulado.
- La investigación y generación de diferentes instancias del problema para probar la efectividad del modelo propuesto.

Capítulo 2

Marco Teórico

2.1 La planeación de los recursos y la administración de la capacidad

La planificación de los recursos es una actividad muy importante ya que tiene aplicación en una amplia gama de actividades. Para una empresa productiva, la adecuada utilización de los recursos está directamente relacionada con la generación de ganancias por lo que se requiere que dicha planificación sea lo más eficiente posible [17].

Existen tres fases fundamentales en el Proceso de Planeación: La fase estratégica, la fase táctica y la fase operativa. Dentro de la fase estratégica se encuentra la Planeación de la Capacidad, el Plan Maestro de Producción (MPS) constituye la fase táctica y en la fase operativa se encuentra la planeación de requerimiento de materiales (MRP) así como el control de la producción que se da dentro del “día a día” [5].

La Capacidad dentro del contexto de la cadena de suministro, se refiere a lo que puede lograr empleando todos los recursos del sistema de producción: los centros de trabajo, centros de almacenamiento, recursos humanos, equipo, etc. Cuando se habla de tener o no la Capacidad, se refiere a tener los recursos necesarios para lograr la producción requerida dentro del tiempo permitido. El factor tiempo es muy importante, ya que no sólo se trata de “lo que se puede hacer”, sino “lo que se puede hacer dentro del plazo”. Aquí es donde entra la Administración de la Capacidad, junto con la planeación de la Capacidad y el control de la Capacidad.

Los límites competitivos de la empresa son definidos por la capacidad del sistema de producción [5], debido a que indican su capacidad de respuesta a los requerimientos del mercado, su estructura de costos, la composición de su mano de obra y su estrategia general de manejo de inventarios. Cuando la capacidad de una empresa no es adecuada, esta puede perder clientes porque su servicio será lento y permitirá que la competencia entre al mercado. Por otro lado, cuando la capacidad es excesiva, se tienen que hacer cosas como reducir los precios para

estimular la demanda, el personal es sub-utilizado, hay inventario en exceso o se tiene que ampliar el portafolio de productos aunque sean menos rentables con tal de seguir en actividad .

La fase operativa del proceso de planeación es la que ocupará nuestra atención, ya que en ella se realiza el control de la producción durante el “día a día”, lo que también se conoce como la planificación de tareas. Esta fase operativa frecuentemente presenta problemas debido a que los planes no son realizables dentro del horizonte de planeación propuesto. Esto puede ser ocasionado por un déficit en la capacidad del sistema o porque situaciones imprevistas no permiten la finalización de las tareas en el plazo establecido. El proceso es dinámico y los cambios en condiciones de la demanda y abastecimiento ocurren muy frecuentemente. El reto del programador es balancear demanda y abastecimiento todo el tiempo y evitando inventarios excesivos o caídas en el servicio al cliente.

2.2 La Planificación de Tareas.

La Planificación de Tareas nos permite hacer una adecuada administración de recursos limitados a través de eventos o actividades asignados a éstos, de manera que sean satisfechas las restricciones específicas del problema [35] y se logre la optimización del o los objetivos.

La Planificación o secuenciación de tareas también se conoce como Job Shop Scheduling Problem (JSSP), y consiste en la programación temporal de las operaciones o tareas en las que se descomponen un conjunto de trabajos [37].

Lenstra y Rinnooy (1984), definen la Planificación de tareas como la óptima asignación de recursos limitados a lo largo de un período de tiempo. Estos recursos se presentan en forma de máquinas y se asignan a actividades conocidas como trabajos, con la principal restricción de que en todo momento, no está permitido que una máquina procese más de un trabajo al mismo tiempo y ningún trabajo puede ser procesado simultáneamente por más de una máquina [34].

En un departamento de producción, el principal objetivo de la Planificación de Tareas es apoyar a sus diferentes áreas funcionales con el propósito de mejorar la competitividad y rentabilidad por medio de una toma de decisiones que optimicen el uso de recursos. En los procesos de manufactura, es necesario definir la secuencia de los diferentes productos que se van

a fabricar durante un periodo de tiempo dado [43]; la relevancia de la planificación de tareas radica en que determina en qué tiempos y en qué máquinas serán realizadas las operaciones.

2.2.1 Historia y Estado del Arte de la teoría de Planificación de tareas

Debido a su gran utilidad al aplicarse en diversos ámbitos (Ej.: elaboración de horarios en escuelas, programación de vuelos en aeropuertos, en un taller para definir la secuencia de reparación, programación de viajes para el transporte de viajeros y mercancías) y a la complejidad de su resolución (el problema de 10 trabajos para ser programados en 10 máquinas propuesto por Fisher y Thompson permaneció sin resolver por más de 25 años, aún cuando mucha investigación fue dedicada a éste [47]), los problemas de Planificación de Tareas han despertado gran interés en los investigadores, los cuales han desarrollado metodologías y algoritmos para eficientar su solución [17]

El rango de aplicación de la Planificación de Tareas abarca muchas áreas de conocimiento entre las que podemos mencionar las ciencias de la computación, investigación de operaciones, producción y servicios entre otras, por lo que un amplio espectro multidisciplinario de investigadores se ha dedicado al desarrollo de la teoría de Planificación de Tareas durante los últimos 50 años [27].

Entre los primeros trabajos que abordaron la programación de tareas, podemos mencionar la solución que S.M. Johnson publicó en 1954 para minimizar el tiempo de ejecución de n trabajos en 2 máquinas [31], bajo varias limitantes supuestas sin considerar el tiempo de manejo de materiales y la posibilidad de caer en un ciclo recurrente que bloqueara el sistema. En 1956, Jackson [28] proveyó una solución óptima para el problema de dos máquinas, con supuestos similares al algoritmo de Johnson pero con enrutamientos aleatorios. Son, *et al.* [49], condujeron un experimento para determinar las condiciones donde los algoritmo de Johnson y Jackson se mantienen óptimos y concluyeron que la calidad de estas soluciones es inversamente proporcional a el tiempo de manejo de materiales, con una mayor dificultad para mantener la calidad de la solución en el caso del algoritmo de Jackson, debido a la gran cantidad de soluciones al no existir condiciones de precedencia entre las operaciones de los trabajos.

Akers y Friedman introdujeron en 1955 [2] un enfoque geométrico que consistía en reducir el problema de la programación de dos máquinas en una ruta menor. Aggoune en 2004 [1] desarrolló una extensión polinomial de este enfoque geométrico, el cuál llamó enfoque geométrico temporizado para el problema de dos máquinas bajo condiciones de disponibilidad, teniendo como objetivo minimizar el tiempo total de proceso mediante una nueva definición de los vértices existentes e introducción de nuevos vértices, así como un programa dinámico para la evolución de éstos.

Los “métodos eficientes” de Johnson, Akers-Friedman y Jackson constituyeron un avance importante para la solución óptima de los problemas $2 \times m$ donde se manejan 2 trabajos y m máquinas. Desde entonces, y a pesar del exhaustivo trabajo de investigación y los logros alcanzados, ha sido difícil encontrar métodos eficientes que conduzcan a soluciones óptimas para problemas con $m \geq 3$ y $n \geq 3$, denotando m el número de máquinas y n el número de trabajos [27]. El problema también ha sido abordado mediante el uso de técnicas de programación matemática como la programación lineal entera mixta [38], que sin embargo sus elevados tiempos de ejecución en problemas de tamaño real la hacen poco apropiada para este tipo de problemas [43].

B. Giffler y G. L. Thompson, desarrollaron en 1960 [23] un algoritmo para resolver el problema de la Planeación de la Producción en un taller de manufactura por medio de minimizar la longitud de los programas de producción generando soluciones factibles en una parcialidad dentro del universo de soluciones sin atrasar ninguna otra operación y sin violar las restricciones tecnológicas. El procedimiento de generación de este algoritmo explora el espacio de búsqueda por medio de una estructura de árbol. Recientemente, algunos investigadores [13] han propuesto el enfoque heurístico de generación de programas mediante una modificación a este algoritmo donde se especifica una regla de prioridad para seleccionar una operación del total de operaciones en conflicto, Bierwirth y Mattfeld [13] presentaron una calendarización híbrida para reducir o aumentar el espacio de búsqueda por medio de la introducción de un parámetro al algoritmo de Giffler y Thompson. El trabajo de Giffler y Thompson [23], es considerado la base de todas las heurísticas basadas en reglas de prioridad de despacho.

En 1964 Roy y Sussmann [43] fueron los primeros en proponer una representación del problema de calendarización de trabajos mediante gráficos disyuntivos y Balas [9] en 1966 fue el primero en aplicar acercamientos enumerativos basados en estos gráficos.

En 1966, Fogel [18] propuso la técnica de programación evolutiva, en la cual la inteligencia se ve como un comportamiento adaptivo y enfatiza los nexos de comportamiento entre padres e hijos en vez de buscar emular operadores genéticos específicos [46]. John Holland se interesó en estudiar los procesos lógicos involucrados en la adaptación y vio a esta en términos de un formalismo en que los programas de una población interactúan y mejoran en base a un cierto ambiente que determina lo apropiado de su comportamiento [15]. Los algoritmos genéticos fueron presentados en 1975 por Holland [25] como técnicas de búsqueda estocástica basadas en el proceso de la evolución natural, y han sido aplicados exitosamente en un amplio rango de problemas de optimización combinatoria. En el año de 1992, Koza [33] propuso un método llamado Programación Genética cuya meta era lograr que las computadoras resuelvan problemas para las que no han sido explícitamente programadas. El método de Koza ha sido utilizado en un sinnúmero de aplicaciones como la compresión de imágenes, el diseño de circuitos, el reconocimiento de patrones y la planeación de movimientos de robots [15]. Actualmente han sido desarrolladas muchas aplicaciones de los algoritmos genéticos, como los inspirados en el sistema inmune artificial [17] para resolver el *Job Shop Scheduling Problem*.

La Búsqueda Tabú tiene su antecedente en Glover, que en el año de 1977 presentó un método para resolver problemas de programación entera con restricciones sustitutas [24] con un método diseñado para evitar óptimos locales mediante la penalización de ciertos movimientos considerados “tabú” que evitan que se caiga en ciclos durante la búsqueda.

En el año de 1994, Lee y Dicesare [35] utilizaron Redes de Petri para modelar problemas de calendarización en centros de manufactura, proponiendo un algoritmo heurístico de búsqueda que generara y buscara una buena secuencia de trabajos, siendo este mecanismo un proceso iterativo con un tiempo de procesamiento muy grande.

En el año de 1953, Metrópolis, *et al.* [39], desarrollaron un método de Monte Carlo para calcular las propiedades de cualquier sustancia que se considere compuesta de moléculas individuales que interactúan entre sí. Este procedimiento puede ser usado para simular una colección de átomos en equilibrio termodinámico a una temperatura dada. Con este procedimiento se pudo simular la manera en la cual los cristales metálicos se reconfiguran y alcanzan el equilibrio en el proceso de recocido. Esto inspiró a Kirkpatrick, *et al.* [32] a desarrollar el algoritmo de Recocido Simulado en el año de 1983, implementando el algoritmo de Metrópolis para hacer una aproximación numérica al simular el proceso de recocido de materiales

en un problema de optimización combinatoria. La técnica del algoritmo de Recocido simulado hace aleatorio el procedimiento de optimización local de manera que se reduce la posibilidad de quedar atrapado en una solución pobre pero localmente óptima, permitiéndole así obtener mejores soluciones. El algoritmo de Recocido Simulado ha sido aplicado en varios problemas de optimización combinatoria entre los cuales se encuentra el *Job Shop Scheduling Problem* [37].

Van Laarhoven *et al* [53], introdujeron en el año de 1992 un algoritmo de aproximación para el problema de minimizar el tiempo de completamiento para en la calendarización de trabajos en un taller de manufactura, basando su algoritmo en el método de recocido simulado. En el año de 1996, He *et al.* consideraron además las fechas de entrega de los trabajos con el objetivo de minimizar el atraso, utilizando un algoritmo que integra el recocido simulado y un algoritmo heurístico de intercambio mediante el empleo de una técnica de inserción.

A partir de los trabajos mencionados anteriormente se ha derivado una extensa investigación en la Planificación de Tareas, llegando hasta el desarrollo de sofisticadas técnicas de optimización y aproximación. Las técnicas heurísticas que hoy en día se conocen adaptan ideas conocidas desde hace mucho tiempo en otras disciplinas, por ejemplo, los algoritmos genéticos emulan los mecanismos de evolución, el recocido simulado se basa en un proceso físico de la industria metalúrgica y la búsqueda tabú toma conceptos de la Inteligencia Artificial.

En los últimos años se ha implementado el uso de técnicas hiperheurísticas, que consisten en la combinación de diferentes heurísticas simples en las que se busca que la interacción entre ellas proporcione mejores resultados para diversas instancias al balancear las debilidades de algunas técnicas con las fortalezas de otras.

La optimización difusa ha sido aplicada con éxito en la planificación de tareas bajo demanda incierta [11] teniendo como resultado soluciones subóptimas pero eficientemente flexibles para lidiar con las fluctuaciones; estos trabajos evolucionaron incluyendo también incertidumbre en la disponibilidad de los recursos [12], sin embargo, aún cuando este método es un enfoque preventivo para el manejo de información imprecisa, no reemplaza enfoques reactivos como la planificación de tareas auto-ajustable aunque la integración de ambos métodos es un campo promisorio para trabajos futuros.

2.3 Calidad de la solución

Una solución se considera óptima cuando se tiene la certeza de que no existe otra mejor solución dentro del espacio total de soluciones definido para ese problema.

Para un mismo caso, la definición de lo que es la solución óptima puede variar dependiendo del objetivo que se busque optimizar, y cuando se tiene un problema multiobjetivo, para cada uno de ellos haya una solución óptima, ya sea que se busque minimizar el tiempo de proceso, el tiempo de mano de obra o maximizar las ganancias [2].

La definición de solución óptima es mal concebida en los casos de calendarización ya que es difícil que pueda darse en un entorno de producción realista. Una asignación de tareas que resulta ser óptima en un momento dado puede no serlo después dado lo impredecible de las operaciones dentro de una fábrica.

Los métodos que obtienen una solución óptima para el problema de planificación de tareas, generalmente son aplicables solamente a problemas relativamente pequeños, ya que la dificultad computacional tiende a incrementarse con el tamaño del problema. Los problemas prácticos son usualmente resueltos mediante procedimientos heurísticos que generan una asignación de tareas que si bien no es óptima o está cerca de la optimalidad, se espera que su desempeño esté por encima del que se tenía antes de aplicar el heurístico y el tiempo de cómputo, es decir, de obtención de secuencias de producción válidas sea menor con el uso de este procedimiento.

El objetivo principal de una técnica de optimización es encontrar el óptimo o los óptimos locales de cualquier problema. La Optimización Global es un área de las matemáticas que se ocupa de desarrollar los formalismos que permiten garantizar la convergencia de un método hacia el óptimo global de un problema pero desafortunadamente esto sólo puede garantizarse en algunos casos limitados como son los problemas con espacios de búsqueda convexos [15].

2.4 Métodos de resolución para los Problemas de Planificación de Tareas.

Los métodos de resolución para los problemas de Planificación de Tareas se dividen en dos grandes categorías: los métodos exactos, que producen una solución óptima pero a un costo

computacional muy alto cuando se trata de problemas con más de dos máquinas y dos trabajos, y los métodos heurísticos o de aproximación que producen una aproximación a una solución óptima, una buena solución, en un tiempo aceptable.

2.4.1 Métodos Exactos

En estos métodos, también conocidos como de optimización o algoritmos polinómicos de resolución, el tiempo de ejecución se incrementa exponencialmente cuando el tamaño del problema de Planificación de Tareas aumenta linealmente.

Entre los métodos exactos, podemos mencionar los siguientes:

2.4.1.1 Enumeración Exhaustiva

Se refiere a la evaluación de todas las posibles soluciones para un problema dado. Consiste en desarrollar la totalidad de secuencias posibles de un problema definiendo el orden en el cuál los trabajos deberán ser procesados y evaluar cada secuencia. Entonces se selecciona aquella cuyo valor sea el mejor de la función objetivo. La completa enumeración de las soluciones para un problema garantiza que se encontrará la mejor asignación de recursos posible.

La principal desventaja de este método radica en que la examinación de todas las secuencias posibles implicaría el análisis de $n!$ combinaciones siendo n el número de trabajos, y si el valor de n es muy grande, esto puede ser una tarea imposible. Esto hace impráctica la aplicación de la enumeración exhaustiva en casi todos los casos, excepto para problemas pequeños, por ejemplo, de una sola máquina donde el número de trabajos sea menor que 10 [50].

2.4.1.2 Técnicas de Programación Matemática

El tipo más básico de programación matemática es la programación lineal, creada por George Dantzig en 1947 [41], que se refiere a un problema de optimización con un conjunto de restricciones lineales y una sola función objetivo lineal.

En 1979, Leonid Kachiyan [41] demostró mediante su algoritmo del elipsoide que la programación lineal tiene un algoritmo de tiempo polinomial en el modelo de la máquina de Turing, introduciendo radicalmente nuevos puntos de vista y técnicas a la programación lineal.

Existen varias clases de algoritmos para hacer frente a la programación lineal, siendo los dos más importantes el método simplex y el método de puntos interiores conocido como el algoritmo de Karmarkar, que fue publicado en 1984 y ha demostrado resolver problemas de programación lineal en tiempo polinomial [44].

La programación entera consiste básicamente en una programación lineal, con el requerimiento adicional de que sus variables deben ser enteras. Si sólo una parcialidad de las variables se requiere que sea entera permaneciendo el resto de variables como continuas entonces se trata de un modelo de programación lineal entera mixta, siendo el modelo de Manne uno de los más conocidos para resolver el problema de Planificación de Tareas [38]. En contraste con la programación lineal, no existe un algoritmo eficiente en tiempo polinomial para la programación entera y entera mixta. Muchos de los problemas de planificación de tareas pueden ser formulados en programación entera. Los mejores algoritmos para la resolución de problemas en programación entera son las técnicas de plano de corte (poliedro) y las técnicas de ramificación y acotamiento [44].

Los enfoques a este método cuyas aplicaciones han resultado más exitosas son los que abordan la relajación de Lagrange, donde las restricciones de capacidad y precedencia son relajadas usando multiplicadores de Lagrange no negativos y con términos de penalización incorporados en la función objetivo. También los métodos de descomposición han tenido aplicaciones con buenos resultados, dividiendo el problema original en sub-instancias más pequeñas y manejables [27], aunque aún estas estrategias suponen un esfuerzo computacional muy grande y los resultados presentan una gran desviación respecto al valor que se considera óptimo.

Se ha dicho que para el *Job Shop Scheduling Problem*, los métodos de programación matemática no guían hacia un resultado práctico [23] y que estos modelos de programación matemática aún no han alcanzado el progreso suficiente para resolver estos problemas de calendarización y asignación [27], por lo que para la resolución al problema de Planificación de Tareas deben usarse otros métodos.

2.4.1.3 Método de Ramificación y Acotamiento.

Morton y Pentico, explican la idea básica de la ramificación mediante la conceptualización del problema como un árbol de decisión [40]. Para cada punto de decisión, llamado nodo y que representa una solución completada parcialmente, crecen varias ramificaciones, una para cada posible decisión que seguiría completando dicha solución. Estas ramificaciones a su vez se convierten en nuevos nodos con sus propias ramificaciones y así sucesivamente. Cuando un nodo ya no tiene ramificaciones posibles representa una solución completa, que no necesariamente es buena. Mediante una evaluación de todas las soluciones completas, la mejor de éstas será la solución óptima; sin embargo esto sería computacionalmente muy costoso. Es aquí cuando el acotamiento entra en acción, proveyendo un medio de reducción de soluciones posibles, mediante el establecimiento de un límite inferior a partir del cual no serán considerados para ramificación todos los nodos que representen las posibles soluciones cuyo valor esté por debajo de éste límite inferior; de esta manera el proceso enumerativo es reducido y con ello su costo computacional.

Los métodos enumerativos de ramificación y acotamiento son ampliamente utilizados para obtener soluciones óptimas a problemas de Planificación de Tareas que son NP-difíciles, siendo su principal desventaja que su tiempo de ejecución es directamente proporcional al número de nodos a considerar [44]. A pesar de todo, este método permite reducir el número de secuencias a examinar respecto al método de enumeración exhaustiva y puede ser usado para resolver cualquier problema de optimización que tenga un número finito de soluciones posibles [50].

2.4.2 Métodos Heurísticos.

También llamados algoritmos de aproximación, al correr en tiempo polinomial sólo pueden garantizar una solución que sea un porcentaje fijo ρ de la mejor solución por lo que estos métodos también se conocen como algoritmos de aproximación ρ [27].

Los algoritmos metaheurísticos han sido desarrollados para resolver problemas difíciles de optimización combinatoria donde los heurísticos tradicionales no son efectivos. Se trata de algoritmos híbridos que combinan diferentes conceptos derivados de la evolución biológica, la inteligencia artificial y los mecanismos estadísticos [42]. Ejemplo de algoritmos metaheurísticos son el Recocido Simulado, la Búsqueda Tabú, el Algoritmo Voraz de búsqueda adaptativa, los Algoritmos Genéticos, entre otros.

A continuación se describen brevemente algunos de los métodos heurísticos y metaheurísticos más importantes.

2.4.2.1 Reglas de despacho

Las reglas de prioridad de despacho han sido utilizadas para el problema de Planificación de Tareas y como base para el desarrollo de los métodos de aproximación, volviéndose una técnica popular debido a su facilidad de implementación y a la reducción sustancial en los requerimientos computacionales [27]. Estas reglas pueden ser clasificadas de muchas maneras: respecto al tiempo pueden ser estáticas (no dependen de él) o dinámicas (dependientes del tiempo); otra manera de clasificarlas es respecto al nivel de información que manejan: local si sólo maneja información de la maquina donde se ejecuta o global si maneja información de varias máquinas [44].

El procedimiento general de las reglas de despacho es el siguiente: En cada etapa, a todas las operaciones que están disponibles para ser programadas se les asigna una prioridad y la operación con la más alta prioridad es la siguiente en ser programada; varias iteraciones son necesarias para obtener un resultado válido.

El algoritmo de Giffler y Thompson es considerado como la base común de las reglas de prioridad de despacho y su importancia deriva del hecho de que este algoritmo genera programaciones activas [27]. El óptimo global de cualquier problema de Programación de Tareas se encuentra en el conjunto de las programaciones activas (aquellas donde ninguna operación puede adelantarse sin que haya un atraso en el inicio de las operaciones previas) y que son un subconjunto contenido en el de las programaciones semiactivas (aquellas donde ninguna tarea puede ser adelantada sin modificar las preferencias o sin que se produzcan traslapes) [37]. La caracterización de estos subconjuntos es muy útil para reducir el espacio de búsqueda en procedimientos de optimización [6].

Este tipo de reglas son muy útiles en problemas con un solo objetivo [21], sin embargo los problemas reales entrañan una mayor complejidad por lo que se necesitan reglas de despacho más elaboradas o compuestas con el fin de obtener asignaciones aceptables. Estas reglas compuestas de despacho básicamente consisten en una combinación de varias reglas de despacho elementales.

2.4.2.2 Heurísticas basadas en cuellos de botella

Estos algoritmos están diseñados para manejar un escenario de planificación donde existen operaciones que son cuellos de botella. Se comienza por identificar las restricciones o las operaciones cuello de botella los cuales tendrán que ser programados primero para maximizar el desempeño del sistema [14].

2.4.2.3 Métodos Metaheurísticos de Búsqueda Local

Los métodos metaheurísticos de búsqueda local han sido aplicados exitosamente a muchos problemas de optimización combinatoria. Haciendo referencia a la estrategia que rige la búsqueda local, se tienen dos grandes tipos: La exploración en espacios vecinos y la búsqueda adaptativa. El recocido simulado y la búsqueda tabú son los más grandes representantes del método de

exploración en espacios vecinos; mientras que la búsqueda adaptativa es principalmente representada por los algoritmos genéticos y otros algoritmos evolutivos.

Este tipo de algoritmos son parte de los llamados “Algoritmos de tipo de mejora”, que a diferencia de los algoritmos de tipo constructivo, parten de una solución que puede haber sido seleccionada arbitrariamente y a partir de la manipulación de ésta se intenta obtener una mejor solución “vecina”. La vecindad entre dos soluciones existe cuando una de ellas puede ser obtenida a través de una modificación bien definida de la otra solución [44].

El principio básico de los algoritmos de búsqueda local es perturbar soluciones existentes para obtener mejoras en la solución. El funcionamiento general de los métodos de búsqueda local es de manera siguiente

- Paso 1. Se tiene una solución inicial
- Paso 2. En cada iteración, el método de búsqueda local realiza una búsqueda dentro de la vecindad de soluciones y evalúa las diferentes soluciones “vecinas”.
- Paso 3. El método acepta o rechaza candidatos a nueva solución de acuerdo con un criterio de aceptación-rechazo.
- Paso 4. Si una candidata a solución es rechazada se continúa la búsqueda, si es aceptada, sustituye a la solución actual.
- Paso 5. El algoritmo se repite desde el paso 2 hasta un criterio de paro.

Un método de búsqueda local no garantiza una solución óptima global sino simplemente dar una solución factible.

2.4.2.4 Recocido Simulado

En el año de 1953, Metrópolis et al. desarrollaron un método de Monte Carlo para calcular las propiedades de cualquier material o sustancia compuesta por moléculas interactuando individualmente [39]; con este procedimiento puede ser simulada la manera en que los cristales de metal se reconfiguran y alcanzan el equilibrio. El Recocido Simulado está basado en los principios de la física y de la ciencia de los materiales; a partir de la analogía con el proceso de recocido utilizado para encontrar estados de baja energía en los sólidos. En física, el recocido denota un proceso en el cual un sólido es primeramente derretido mediante el incremento de su temperatura, seguido de una reducción controlada de la temperatura encaminada a la obtención de un estado sólido de baja energía. Si el enfriamiento es realizado muy rápidamente, surgen numerosas irregularidades en la estructura de los sólidos, obteniendo estados de energía relativamente alta. Por el contrario, un cuidadoso recocido a través de una serie de niveles, donde la temperatura se mantiene bastante tiempo en cada nivel para alcanzar el equilibrio, conduce a estructuras más estables asociadas con estados de baja energía [22].

Tomando como base el algoritmo de Metrópolis, Kirkpatrick et al. [32] desarrollaron el algoritmo de recocido simulado para optimización global en problemas de optimización combinatoria mientras que Cerny en 1985 empleó un enfoque similar para el problema del agente viajero [57]. Desde su introducción, el recocido simulado ha ganado popularidad como un método de resolución de problemas combinatorios duros.

El recocido simulado es un enfoque de búsqueda estocástica capaz de escapar de óptimos locales mediante una modificación a la solución actual, que guía a una nueva solución la cual puede ser aceptada con el uso de una probabilidad de transición que depende de dos factores siendo el primero la diferencia entre el valor de la función objetivo de la solución actual y la solución candidata [57]; el segundo es un parámetro conocido como temperatura. La búsqueda se inicia con altas temperaturas, que hacen más probable que cualquier nueva solución sea aceptada, aún cuando no sea tan buena; conforme la búsqueda continúa, la temperatura declina (de acuerdo con un factor de enfriamiento predefinido) y con ello la probabilidad de que el valor de la solución candidata se aleje de una buena solución, ya que cuando la temperatura es lo suficientemente baja, sólo son aceptadas nuevas soluciones que mejoren el estado actual [22]. Hay dos tipos de algoritmo de temperatura: el homogéneo, donde el valor de la temperatura

permanece constante para un cierto número de iteraciones y el heterogéneo, donde la temperatura decrece mediante transiciones subsecuentes [58]. Contrariamente a la mayoría de los metaheurísticos, este método converge asintóticamente a un óptimo global (asumiendo un número infinito de iteraciones) [22].

De acuerdo con Pinedo [44], el recocido simulado es un procedimiento iterativo, donde para la iteración k existe una solución S_k , así como la mejor solución actual S_0 . $G(S_k)$ y $G(S_0)$ denotan los valores correspondientes de la función objetivo para esas soluciones, donde $G(S_k) \geq G(S_0)$. El algoritmo se mueve de una solución a otra en busca de la mejor. En la iteración k , la búsqueda por una nueva solución es llevada a cabo dentro de la vecindad de S_k , de esta búsqueda surge una solución “candidata” S_c . Si $G(S_c) < G(S_k)$ se hace un movimiento, estableciendo $S_{k+1} = S_c$. Si $G(S_c) < G(S_0)$ entonces $S_0 = S_c$. Sin embargo, si $G(S_c) \geq G(S_k)$ se hace un movimiento a S_c con probabilidad:

$$P(S_k, S_c) = \exp\left(\frac{G(S_k) - G(S_c)}{\beta_k}\right);$$

Con la probabilidad $1 - P(S_k, S_c)$ la solución candidata S_c es rechazada y se establece como $S_{k+1} = S_k$. La solución S_0 no cambia cuando es mejor que la solución candidata S_c . Los parámetros de control $\beta_1 \geq \beta_2 \geq \beta_3 \geq \dots > 0$ se denominan parámetros de refrigeración o de temperaturas (en analogía con el proceso de recocido). Frecuentemente β_k es escogido para tomar valores de a^k para alguna a entre 0 y 1.

En el recocido simulado está permitido realizar movimientos hacia peores soluciones con el objeto de evitar caer en un mínimo local y más tarde encontrar una mejor solución. Ya que β_k disminuye en cada iteración, la probabilidad de aceptar una solución que no aporte mejoras disminuye también con cada iteración. En cuanto al criterio de paro para el algoritmo, se puede considerar cierto número de iteraciones, o cierto número de iteraciones sin mejora, o determinado tiempo [44].

El método puede ser resumido de la siguiente manera [44]:

- Paso 1. Establecer $k = 1$ y seleccionar β_1
- Seleccionar una solución inicial usando algún método heurístico.
- Establecer $S_0 = S_1$

Paso 2. Seleccionar una solución candidata S_c de la vecindad de S_k

Si $G(S_0) < G(S_c) < G(S_k)$, entonces establecer $S_{k+1} = S_c$ e ir al Paso 3

Si $G(S_c) < G(S_0)$, establecer $S_0 = S_{k+1} = S_c$ e ir al Paso 3

Si $G(S_c) > G(S_k)$, entonces generar un número aleatorio U_k de una distribución uniforme (0,1)

Si $U_k \leq P(S_k, S_c)$, establecer $S_{k+1} = S_c$, de otro modo establecer $S_{k+1} = S_k$ e ir al Paso 3.

Paso 3. Seleccionar $\beta_{k+1} \leq \beta_k$.

Incrementar k en 1.

Si $k = N$ entonces detener el algoritmo, de otro modo regresar al paso 2.

La representación en pseudocódigo del método de recocido simulado es la siguiente:

RC()

Entrada: la instancia de un problema P

Salida: la mejor solución encontrada para el problema P

$T \leftarrow T_0$ *inicializa temperatura*

$\alpha \leftarrow \alpha_0$ *inicializa factor de enfriamiento*

$U \leftarrow U_0$ *solución inicial*

$L \leftarrow L_0$ *cadena de Markov*

Hacer

Para $i=1$ *hasta* $i=L$

$V \leftarrow$ *perturbación (U)*

Si $f(V) < f(U)$

 Entonces $U \leftarrow V$

Si no $e^{-\frac{f(U)-f(V)}{T}} \leq \text{rand}[0,1]$

 Entonces

$U \leftarrow V$

Fin para

$T \leftarrow \alpha T$

Mientras (se cumple criterio de terminación)

El algoritmo de recocido simulado puede ser modelado matemáticamente usando cadenas de Markov [48], que consisten en una secuencia de eventos, donde la probabilidad del resultado de un evento depende sólo de los resultados del evento anterior. En recocido simulado, un evento corresponde a una transición y el conjunto de los resultados posibles están dados por un conjunto de estados vecinos y cada movimiento depende únicamente de los resultados del intento anterior por lo tanto, aplica el concepto de cadenas de Markov.

La efectividad del recocido simulado depende en el diseño de la vecindad así como en la manera en que la búsqueda es conducida dentro de la ésta. Si la vecindad es diseñada de manera que facilite los movimientos hacia mejores soluciones y salga de mínimos locales, entonces el procedimiento funcionará de manera correcta [44].

Una ventaja del recocido simulado, es que es sencillo de programar para muchos problemas y una sola corrida del programa, aunque larga, es suficiente para obtener una buena solución comparable a la que pueden encontrar otros métodos más sofisticados y rápidos en ejecución, pero con un tiempo de implementación que puede llevar semanas entender y programar [4].

El recocido simulado se puede adaptar fácilmente a los nuevos problemas y, debido a su capacidad para evitar óptimos locales pobres, es apto en la obtención de muy buenos resultados [29].

2.4.2.5 Búsqueda Tabú

Esta técnica se basa en conceptos tomados de la Inteligencia Artificial, y se utiliza como una metaheurística que se superpone a una técnica de búsqueda para resolver problemas de optimización combinatoria. A diferencia del método de ramificación y acotamiento, la búsqueda tabú es de “inhibición débil” ya que los movimientos penalizados dejan de serlo después de un período de tiempo [16].

La Búsqueda Tabú es similar al recocido simulado en el aspecto de que también se mueve de una solución a otra, con la posibilidad de considerar soluciones menos óptimas [44]. También, como en el recocido simulado, es definida una vecindad para cada solución y de igual manera se realiza la búsqueda de una solución candidata en la vecindad, búsqueda que puede ser hecha

aleatoria y organizadamente. La diferencia básica entre la búsqueda tabú y el recocido simulado consiste en el mecanismo que es utilizado para la aprobación de una solución candidata; en la búsqueda tabú el mecanismo tiene una naturaleza determinística. En cualquier etapa del proceso se conserva una lista tabú de mutaciones, las cuales el proceso no tiene permitido hacer. Cada vez que un movimiento se debe a una mutación en la solución actual, la mutación reversa es añadida al principio de la lista tabú con el objeto de evitar retornar a un mínimo local que ya ha sido visitado con anterioridad [44].

Pinedo resume el algoritmo de la Búsqueda Tabú como sigue [44]:

Paso 1. Establecer $k = 1$

Seleccionar una solución inicial S_1 haciendo uso de algún método heurístico.

Establecer $S_0 = S_1$

Paso 2. Seleccionar una solución candidata S_c de la vecindad de S_k

Si el movimiento $S_k \rightarrow S_c$ está prohibido por una mutación en la lista tabú, entonces establecer $S_{k+1} = S_k$ e ir al paso 3.

Si el movimiento $S_k \rightarrow S_c$ está prohibido por una mutación en la lista tabú, entonces establecer $S_{k+1} = S_c$.

Añadir la mutación reversa al principio de la lista tabú;

Bajar una posición al resto de las entradas de la lista tabú

Eliminar la entrada inferior de la lista tabú

Si $G(S_c) < G(S_0)$, entonces establecer $S_0 = S_c$;

Ir al paso 3.

Paso 3. Incrementar k en 1

Si $k = N$ entonces detener el algoritmo, de otro modo regresar al paso 2.

2.4.2.6 Cómputo Evolutivo

La evolución, tal como fue propuesta por Charles Darwin, es acerca de la supervivencia del más apto. Los elementos requeridos para que la evolución tenga lugar son una muestra de individuos, la habilidad de éstos de cambiar sus características a través de las generaciones y un

medio ambiente que permita sólo la reproducción de individuos con ciertas características y capacidad de adaptación. De esta manera, a través de muchas generaciones, se garantiza que solamente sobrevivan los individuos con mayor aptitud para desempeñarse en su entorno.

Similarmente, los algoritmos evolutivos utilizan el principio de la evolución de Darwin para buscar respuestas a problemas complejos dentro de grandes espacios de solución. El campo del cómputo evolutivo es uno de los con más rápido crecimiento debido a que puede abarcar un amplio rango de problemas cuya resolución anteriormente se consideraba fuera del alcance de los métodos científicos vigentes, su aplicación es multidisciplinaria en áreas como medicina, ingeniería industrial, la milicia y la inteligencia artificial [18].

Fogel resalta las ventajas del método evolutivo sobre otros métodos tradicionales de optimización [18], siendo una de éstas la adaptabilidad de estos métodos dentro de un ambiente incierto sin necesidad de un replanteamiento del problema que sería computacionalmente costoso. Otra ventaja es la habilidad de estos métodos de generar suficientes soluciones buenas con la suficiente rapidez para que puedan ser de utilidad.

John Holland en su libro “Adaptación en sistemas naturales y artificiales” [25] mostró como el proceso evolutivo puede ser aplicado a sistemas naturales o artificiales mediante un panorama general de todos los sistemas de adaptación.

De acuerdo con Koza, el algoritmo evolutivo básico es el siguiente [33]:

1. Crear una muestra de soluciones aleatorias al problema
2. Evaluar la optimalidad de cada solución
3. Determinar cuáles de las soluciones pasan a formar parte de la nueva generación
4. Se crean nuevas soluciones a partir de las soluciones anteriormente escogidas
5. El criterio de paro es cuando cierto número de generaciones han surgido o cuando determinado tiempo ha pasado o un grado de optimalidad de la solución se ha alcanzado.

Entre las aplicaciones del cómputo evolutivo se encuentran la predicción, la automatización y el control, el problema del agente viajero, la planeación de rutas, el diseño y entrenamiento de redes neuronales, el reconocimiento de patrones, diseño en ingeniería entre otros [15].

2.4.2.7 Algoritmos Genéticos

Los Algoritmos Genéticos han sido utilizados con éxito durante la última década para resolver problemas de calendarización de eventos. De manera diferente a los métodos de búsqueda tradicionales, los algoritmos genéticos comienzan con un grupo inicial de soluciones y debido a su potencial paralelismo, realiza una búsqueda en el espacio total de soluciones por lo que difícilmente queda atrapado en un mínimo local [36]. Anderson y Ferris resaltan este paralelismo como una ventaja de los algoritmos genéticos frente a los demás métodos de búsqueda como el recocido simulado o la búsqueda tabú [4].

Azizi, et al., nos explican la forma general de un Algoritmo Genético [7], donde cada miembro de la población es considerado como un cromosoma que representa una solución y cuyo valor es asignado por el Algoritmo Genético de acuerdo con una función objetivo específica para el problema. Dos cromosomas (individuos) son seleccionados ya sea aleatoriamente o con una probabilidad a favor de mejorar la aptitud al producir una nueva generación. La reproducción es entonces llevada a cabo a través de dos mecanismos fundamentales. El primer mecanismo es el cruce, que combina partes de las características genéticas de los dos padres seleccionados para producir características genéticas en un nuevo individuo. El segundo mecanismo es la mutación mediante la cual ocurre una modificación espontánea de la composición genética. El nuevo individuo creado a partir de estos dos mecanismos recibe el nombre de hijo o descendencia y es diferente de sus padres, pero comparte algunas características comunes.

En cada generación, los individuos mejor adaptados se reproducen mientras que los menos adaptados mueren. El nacimiento, muerte y procesos de reproducción que determinan la composición de la siguiente generación pueden ser complejos y usualmente dependen de los niveles de adaptación de los individuos en la generación vigente [44].

La función objetivo es la que sirve para evaluar la calidad de las nuevas generaciones. Solamente los cromosomas con los más altos niveles de adaptación sobrevivirán y formarán parte de la nueva generación.

Gracias a estas más recientes técnicas, se ha podido avanzar considerablemente en la solución de problemas cada vez más complejos, aunque a pesar de estos avances, aún existen problemas en espera de una solución.

Entre los problemas que aún no se han podido resolver debido a su especial complejidad está el de la configuración de taller de manufactura o configuraciones del tipo Job Shop, cuya principal dificultad radica en la gran cantidad de posibles soluciones, que hacen difícil una evaluación que encuentre la solución óptima [56]. Para los problemas en los que no se ha encontrado un algoritmo de solución, es necesario considerar la aplicación de algoritmos de aproximación, que si bien no proporcionan la solución óptima, proporcionan una buena solución en un tiempo de cómputo razonable [55]. En los casos en que pueden aplicarse más de un algoritmo de aproximación a un problema determinado, se considera el algoritmo más eficaz a aquel con tiempo de ejecución más rápido o cuyo resultado se aproxime más al valor objetivo deseado o bien, que cumpla el mayor número de restricciones establecidas.

2.5 La complejidad computacional del Job Shop Scheduling Problem.

Alan Turing en 1937 demostró que algunos problemas son tan difíciles que ningún algoritmo puede resolverlos [52]. Existen dos causas de intratabilidad, siendo la más común que la dificultad del problema entrañe una cantidad de tiempo exponencial para encontrar la solución, la segunda causa se refiere a que las soluciones de un problema pueden ser tan extensas que no pueden ser descritas, ni siquiera enumerada por un algoritmo polinómico, aunque esta segunda causa podría implicar que el problema no está planteado de forma realista [55]. Para los algoritmos de tipo intratables, es necesario utilizar algoritmos polinómicos de aproximación, ya que los algoritmos polinómicos de resolución (métodos exactos) no resultan adecuados de acuerdo a lo mencionado.

El tiempo requerido por un algoritmo para resolver un problema se expresa en función de la variable longitud de datos l de cada caso, instancia o tamaño del problema. El tamaño de la instancia del problema está dado por $n \times m$ (n = número de trabajos, m = número de máquinas).

Cada secuencia de operaciones puede ser permutada independientemente de la secuencia de operaciones de otra máquina, por lo que el número de posibles soluciones para el *Job Shop*

Scheduling Problem está dado por $(n!)^m$ [17] denotando n el número de trabajos y m el número de máquinas. Este número constituye el espacio de búsqueda del problema. La enumeración completa de todas las posibles soluciones no es práctica.

La función del tiempo o complejidad temporal C de un algoritmo expresa el tiempo requerido para la solución de un problema, asociando a cada posible longitud de datos l , la mayor cantidad de tiempo $C(l)$ que el algoritmo necesita para resolver esa instancia del problema [55].

La distinción entre algoritmos con función de tiempo polinómica o exponencial es significativa cuando se considera la solución de casos de gran tamaño, ayudando a considerar el tiempo que se tardará en resolver un problema según aumenta la longitud del caso abordado. Para expresar la complejidad de un algoritmo suele usarse la notación “O” (gran O). La siguiente definición nos ayuda a precisar esta noción definiendo un *orden* en el espacio de funciones [55].

$$f(n) = O(g(n)) \text{ si y sólo si existe una constante } c \text{ tal que } \text{abs}(f(n)) \leq c \times \text{abs}(g(n)).$$

Por lo que la función del tiempo de un algoritmo es polinómica si su complejidad temporal C es $O(p(l))$ para un polinomio p de la longitud de datos l del problema.

Para cualquier algoritmo para el que no exista ningún polinomio $p(l)$ tal que su función de tiempo sea de orden $O(p(l))$, es un algoritmo exponencial”

Cuando un algoritmo tiene complejidad $O(g(n))$ significa que al correrlo en una computadora con los mismos datos, pero valores incrementales de n , los tiempos resultantes de ejecución serán siempre menores que el valor absoluto de $g(n)$ [15], siendo los siguientes tiempos los más comunes de los algoritmos:

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n)$$

Los algoritmos exponenciales tienen un límite de funcionamiento aceptable que depende de la longitud del problema, un “umbral” que si se traspasa causa inestabilidad, volviendo muy lenta la obtención de una solución. Los algoritmos polinómicos son más estables, con desempeño independiente de la instancia del problema.

La distinción anterior entre polinómico y exponencial ayuda a definir la *intratabilidad* de un problema, siendo considerado así cuando se puede demostrar que ningún algoritmo polinómico puede resolverlo. Frente a la intratabilidad, la existencia de un algoritmo polinómico califica al problema en cuestión como “tratable” y a este conjunto de problemas se les denomina de clase P [55]. P es una subclase de NP y consiste en el conjunto de problemas que pueden ser resueltos en tiempo polinomial [27] pero no determinísticamente .

El término determinista significa que “sin importar lo que haga el algoritmo, sólo hay una cosa que puede hacer a continuación, es decir, que el paso siguiente se determina por los pasos anteriores”, a su vez, el no-determinismo “es una herramienta imaginaria que hace que los problemas difíciles parezcan triviales y su mayor valía consiste en el hecho de que existe forma de convertir un algoritmo no determinista a uno determinista aunque el costo computacional sea muy elevado” [15].

Un problema es NP-completo si pertenece a la clase NP y es al menos tan difícil como cualquier otro problema en NP. En muchos casos, la combinación de múltiples objetivos y restricciones resulta en un espacio de búsqueda con crecimiento exponencial. Problemas como esos son conocidos como NP-difíciles, los cuales no pueden ser resueltos por métodos determinísticos dentro de un tiempo polinomial. El Job Shop Scheduling Problem es considerado dentro de los problemas numéricos intratables conocidos como NP-difíciles. Garey y Johnson demostraron que el Problema de Planificación de Tareas es un problema NP-difícil, y de entre esta clase es uno de los menos tratables [17].

Se considera al problema de calendarizar los trabajos en un taller de manufactura como uno de los problemas de optimización combinatoria más difíciles de resolver dentro de la clasificación de los problemas NP-difíciles [19].

Algunos problemas NP-difíciles pueden ser resueltos polinomialmente en relación con diferentes representaciones de las variables de entrada; estos algoritmos son conocidos como pseudo-polinomiales. El método estándar para comprobar que un nuevo problema es NP-difícil es

codificar un problema previamente conocido como intratable usando los recursos y el vocabulario del nuevo problema. Si existe una reducción en tiempo polinomial para este nuevo problema, entonces se dice que es NP-difícil.

Garey y Johnson lo explican de la siguiente manera: “Una vez que se ha probado que un problema es NP-completo, se simplifica el proceso para probar problemas NP-completos adicionales. Dado un problema $\Pi \in NP$, todo lo que se necesita hacer es demostrar que algunos problemas Π' previamente conocidos como NP-completos pueden ser transformados a Π ” [20]. De cualquier manera, si un problema es descrito como NP-difícil en un fuerte sentido, como es el caso del Job Shop Scheduling Problem, no es posible encontrar un algoritmo pseudo-polinomial para este problema a menos que $P=NP$.

Capítulo 3

Formulación del problema y descripción de la metodología.

3.1 Panorama general de la Industria de Cosméticos en México.

La Industria de los cosméticos en México registraba 79 fabricantes en el año 2005, los cuales generaban 27,000 empleos directos y 1,800,000 empleos a través del sistema de ventas directas; con una inversión de 830 millones de pesos en 2003 y 970 millones de pesos en 2004 [10].

El mercado mexicano ha registrado un crecimiento en los consumidores durante las últimas décadas debido a un cambio de actitud donde los productos cosméticos y de cuidado personal ya no son considerados suntuarios sino elementos indispensables en el arreglo personal tanto para hombres como para mujeres. Con la recuperación del mercado las tendencias se dirigen hacia el consumo de productos más sofisticados y que respondan a necesidades específicas del consumidor. Innovaciones como nuevas formulaciones y cambios en la presentación del producto, apoyados por la mercadotecnia, contribuyen a incrementar el consumo.

Sin embargo, algunas debilidades del sector cosmético en México son las siguientes [10]:

- Los bajos niveles de integración en la cadena productiva, principalmente falta de acceso a materias primas e insumos a precios competitivos. No hay competitividad en fabricación de insumos.
- Desarrollo limitado de nuevos productos ya que en México se da más la situación de maquila que la investigación y desarrollo. La industria de cosméticos mexicana es eficiente en la reproducción de modelos establecidos.
- Hay pocas empresas con maquinaria adecuada y además no se cuidan calibración y validación.

- Proceso productivo y control de calidad no vigilados por autoridades gubernamentales debido a que no se exige registro sanitario en el país, a pesar de lo cual la mayoría de las empresas tienen estándares propios de calidad.

- Dificultad para el acceso a distintos instrumentos financieros cuando se trata de pequeñas y medianas empresas. Para las Pymes, las limitantes son el desconocimiento de los canales de comercialización en los mercados internacionales y la falta de información para establecer estrategias de comercialización adecuadas.

- Medios de difusión masiva no accesibles a todas las empresas.

- Polarización tanto en las empresas como en el público consumidor.

- Discriminación por parte del consumidor mexicano al tener la impresión de que las marcas internacionales tienen mejor calidad que las nacionales ya que la imagen del producto es muy importante para él.

- El mercado informal afecta a la industria determinantemente, también afecta al consumidor al introducir al mercado mexicano productos de origen asiático de muy bajo precio pero que no cumplen con criterios sanitarios y de seguridad.

A continuación enlisto algunas características del mercado nacional de cosméticos [10]:

- El mercado está muy fragmentado y la competencia es muy fuerte sobre todo entre empresas multinacionales. En el país coexisten las principales empresas multinacionales junto con firmas nacionales, estando más enfocadas estas últimas al sector popular.

- La demanda de los cosméticos es muy variable ya que obedece a modas. Esta situación hace difícil predecir la demanda de estos productos.

- La tendencia es hacia mercados masivos, demostrada por el incremento de la utilización de cadenas de autoservicio para la venta de cosméticos.

- En cuanto al producto, los precios son muy competitivos y la calidad es buena. El empaque y embalaje son aspectos poco desarrollados por la orientación hacia el mercado interno.

- Aún cuando los consumidores desarrollan lealtad a una marca, la baja en el poder adquisitivo ha provocado un incremento en el uso de las “marcas propias” de las tiendas de autoservicio.

- Se ha incrementado el uso de ingredientes naturales y/o orgánicos. La tendencia de la gente se comienza a inclinar hacia productos cuya fabricación es amigable hacia el medio ambiente y en la que no se hayan realizado pruebas en animales.

- El Internet ha comenzado a ser un canal de venta aunque no todavía con mucha importancia debido a la desconfianza que genera en torno a la forma de pago.

3.2 Situación de la compañía

La compañía se dedica a la manufactura de productos cosméticos y de cuidado personal y su principal actividad es el ofrecer servicios de maquila a aproximadamente 10 marcas posicionadas en el mercado. No cuenta con una marca propia.

La empresa fabrica sobre pedido por lo que son los pedidos del cliente los que desencadenan todo el proceso productivo, desde la compra de materias primas, ya que debido a las políticas de la empresa, no se permite mantener inventarios o realizar adquisiciones que no estén respaldadas por una orden de compra por parte del cliente. Por lo mismo, no están consideradas las posibles fluctuaciones en la demanda.

Cada nuevo pedido por parte del cliente tiene una fecha comprometida de entrega. El departamento de Servicio al Cliente se encarga de recibir los pedidos e informa de éstos al área de Planeación de la Producción quien se encarga de analizar la factibilidad de entregar el pedido en la fecha de entrega solicitada, tomando en cuenta aspectos como el inventario y la capacidad de producción disponibles. A partir de la información que arroja este análisis, se establece lo siguiente:

- Si de acuerdo con las fechas de entrega de materias primas por parte de los proveedores, se cuenta con el tiempo necesario para la fabricación del producto antes de la fecha de entrega deseada por el cliente.

- Si se cuenta con suficiente mano de obra disponible para la fabricación de este nuevo pedido o si será necesario tiempo extra o contratar más personal.

- Si el porcentaje de utilización de las máquinas permite que el nuevo pedido pueda agregarse al plan de producción existente sin afectar otros pedidos.

- Una fecha de entrega factible que bien puede ser la misma que el cliente requiere o en caso contrario se negocia con él esta nueva fecha.

El apego a las fechas de entrega comprometidas con los clientes, es muy importante para la compañía, ya que es uno de los parámetros que determinan su posición competitiva y es muchas veces el factor que más influye para la decisión de compra del cliente, o en su permanencia como cliente ya que hay al menos otras cinco compañías que ofrecen un tipo de servicio similar.

Las principales dificultades que enfrenta la empresa es la incertidumbre en la llegada de los nuevos pedidos y un ambiente con gran variabilidad de la demanda. Esta empresa, al ser maquiladora, encara dificultades como son el no trabajar con un pronóstico de ventas confiable ya que al no contar con una marca propia, la única información sobre la demanda es proporcionada por los clientes, cada uno con su propia planeación de la demanda y políticas de confidencialidad que le hacen compartir o no la información. A esto se añade que muchas veces los pronósticos de ventas cambian drásticamente debido a sobreventas, errores en la planeación de la demanda por parte de los clientes, cambios en el comportamiento del mercado o temporalidad de la demanda.

Debido a la fluctuación en la llegada de los materiales y en ocasiones, la no disponibilidad de éstos por parte de los proveedores, no se puede mantener siempre una cantidad de trabajo en proceso constante en el sistema o tener una gran cantidad de órdenes listas para su manufactura por lo que el porcentaje de utilización de la capacidad instalada es variable, dependiendo, entre otras cosas, del abastecimiento oportuno.

Este tipo de situaciones acarrear dificultades al interior de la empresa, ya que así como en ocasiones hay subutilización de recursos porque no se tienen los materiales completos para comenzar algún trabajo, también se da el caso de que al enfrentar una modificación no prevista de la demanda no se cuenta en el momento con los recursos de capacidad suficientes para poder cubrir la demanda urgente a menos que sea a costa de sacrificar las fechas de entrega acordadas con el cliente para los trabajos de prioridad normal. Esto afecta la organización interna de la

empresa, ya que la lleva a cambiar los planes de producción actuales lo que ocasiona la pérdida de la sincronización entre las áreas productivas de la empresa y acarrea costos no deseables en cuanto a tiempos muertos, tiempo extra y material desperdiciado.

Desafortunadamente en ocasiones ni siquiera con este tipo de cambios dentro de la empresa se logra satisfacer los cambios en la demanda y los pedidos urgentes de los clientes, por lo que al no estar el producto en el tiempo que el cliente lo requiere, este cancela el pedido y ocasiona que la empresa se quede con el inventario que de no desplazarse en un período máximo de 6 meses se volverá obsoleto. Esto causa pérdidas económicas porque la inversión en materiales, mano de obra que se hizo al inventario obsoleto no puede recuperar de ninguna manera y a esto hay que añadirle los costos de destrucción y segregación.

Debido a la incertidumbre de la demanda y la urgencia de los clientes en los tiempos de entrega, no es posible hacer inversiones para aumentar la capacidad instalada hasta no tener algún proyecto o venta asegurada que justifique la inversión. Sin embargo, el no contar con tecnología reciente, principalmente para los equipos de manufactura, ha hecho que la empresa pierda competitividad y no sea considerada por los clientes para nuevos proyectos que implican nuevos procesos o tendencias recientes en el mercado.

El reto para la compañía entonces, es el apego a las fechas de entrega comprometidas con el cliente dentro de un ambiente con variabilidad e incertidumbre en la demanda, dadas las restricciones de fechas de entrega fijas y capacidad limitada.

La empresa debe hacer un uso más eficiente de su capacidad instalada para cumplir con los compromisos adquiridos con los clientes. La Planeación de la producción mejora la manera en que se aprovechan los recursos disponibles por medio de la coordinación de actividades que incrementan la productividad y minimizan los costos operativos. La Planeación de la producción ayuda a identificar conflictos en los recursos, ayuda a controlar la liberación de las órdenes de trabajo al área productiva, asegura que las materias primas requeridas sean ordenadas a tiempo, determina si los pedidos podrán entregarse en la fecha y cantidad comprometidas con el cliente y también ayuda a identificar períodos de tiempo disponibles para mantenimiento preventivo.

3.3 Formulación del problema

Cada semana debe elaborarse un plan de producción el cual nos indique cuál es la secuenciación y asignación de trabajos que se debe seguir en las diferentes máquinas, de manera que se cumpla con la cantidad y fechas comprometidas con el cliente.

El problema de asignación de tareas que nos interesa resolver, tiene un tamaño $m \times n$ y las siguientes características:

- Un conjunto finito de trabajos $\{J_i\}_{i=1}^n$
- Un conjunto finito M de m máquinas $\{M_k\}_{k=1}^m$ donde los trabajos serán procesados.
- Una cadena de m_i operaciones $O_{i1}, O_{i2}, \dots, O_{im}$ Para cada trabajo J_i , dando un total de $N = \sum_{i=1}^n m_i$ operaciones.
- Un tiempo ininterrumpido de proceso τ_{ik} de la operación O_{ik} del trabajo J_i , procesada en la máquina M_k .
- Una función objetivo $C_{\max}^* = \min(C_{\max})$ para encontrar la secuencia que minimize el tiempo máximo de terminación de la última tarea del sistema.
- Las restricciones de este problema son las mismas que el problema clásico de asignación de tareas, mismas que han sido enlistadas en la sección 1.2 de este trabajo de investigación y son las siguientes:
 - Los trabajos son unidireccionales.
 - La máquina que inicia no lo hace con el primer trabajo, cualquier máquina puede empezar o terminar.
 - Cada trabajo tiene su propio patrón de flujo o ruta de trabajo a través de las máquinas, el cuál es independiente de los demás trabajos.
 - No está permitido que dos operaciones del mismo trabajo se procesen simultáneamente.
 - Cada máquina puede procesar solamente un trabajo a la vez así como cada trabajo puede ser procesado por solamente una máquina a la vez.
 - Ninguna operación tiene prioridad sobre las demás.
 - Cada trabajo es procesado hasta concluirse, aún cuando haya que esperar y retardarse entre las operaciones procesadas.
 - Un trabajo puede iniciarse en cualquier momento, siempre y cuando esté disponible la máquina y no se haya especificado un tiempo de inicio para éste.
 - Los trabajos tienen que esperar a que la siguiente máquina esté disponible para continuar su proceso
 - Los tiempos de configuración y cambio de máquina son independientes del orden de procesamiento y están incluidos en los tiempos de procesamiento.
 - Hay sólo un tipo de máquina.
 - Es posible que las máquinas estén ociosas en algún momento del plan de trabajo.
 - Se requiere que para cada trabajo, la secuencia de operaciones contenga exactamente una operación para ser procesada por cada una de las máquinas del problema.

Una instancia del problema de Planificación de Tareas se define mediante dos matrices, la primera es una matriz en la que se muestra cada uno de los trabajos a ser procesados y el orden de cada una de las operaciones en las máquinas.

En el siguiente ejemplo tenemos una instancia de 3 trabajos para ser programados en tres maquinas ($m=3, n=3$) . Cada renglón corresponde a un trabajo (J_0, J_1 y J_2) y cada cuadro representa una operación , siendo el número de cada cuadro el número de la máquina en la que debe procesarse dicha operación.

Por ejemplo, para el trabajo 0 (J_0), la operación 1 (O_1) está programada para ejecutarse en la máquina 1 (m_1), la operación 2 (O_2) se ejecutará en la máquina 0 (m_0) y la operación 3 (O_3) se ejecutará en la máquina 2 (m_2).

Trabajo/Operación	Flujo de los trabajos		
	O1	O2	O3
J0	1	0	2
J1	0	1	2
J2	1	2	0

Tabla 1. Matriz de flujo de los trabajos.

La segunda matriz que define una instancia es la que muestra en cada cuadro el tiempo de proceso de cada una de las operaciones de los trabajos, correspondiendo cada renglón a un trabajo y cada cuadro a una operación, siendo el número dentro de cada cuadro el tiempo de procesamiento para esa operación.

Por ejemplo, para el trabajo 1 (J_1), la operación 1 (O_1) tiene un tiempo de proceso de 5 min., la operación dos (O_2) tiene un tiempo de proceso de un minuto y la operación 3 (O_3) tiene un tiempo de proceso de 5 minutos.

Trabajo/Operación	Tiempo de proceso		
	O1	O2	O3
J0	2	3	5
J1	5	1	5
J2	3	3	1

Tabla 2. Matriz de tiempos de los trabajos

Un ejemplo de un plan de trabajo factible para este problema es representado en la Figura 2 mediante una gráfica de Gantt, donde los renglones representan las máquinas y cada cuadro representa una operación. Los cuadros están marcados con el número de trabajo al que corresponde. El eje horizontal especifica el tiempo de ejecución de cada operación.

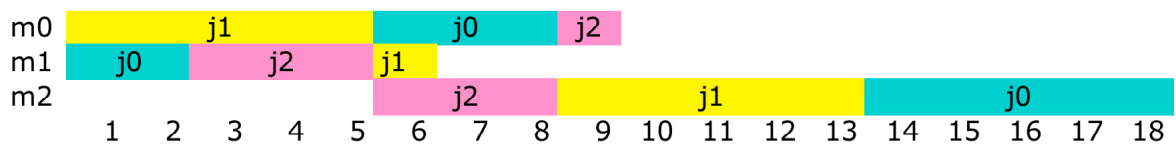


Figura 2. Plan de trabajo.

3.4 Distribución de planta.

El siguiente diagrama muestra la distribución física del área de manufactura de cosméticos de la empresa.

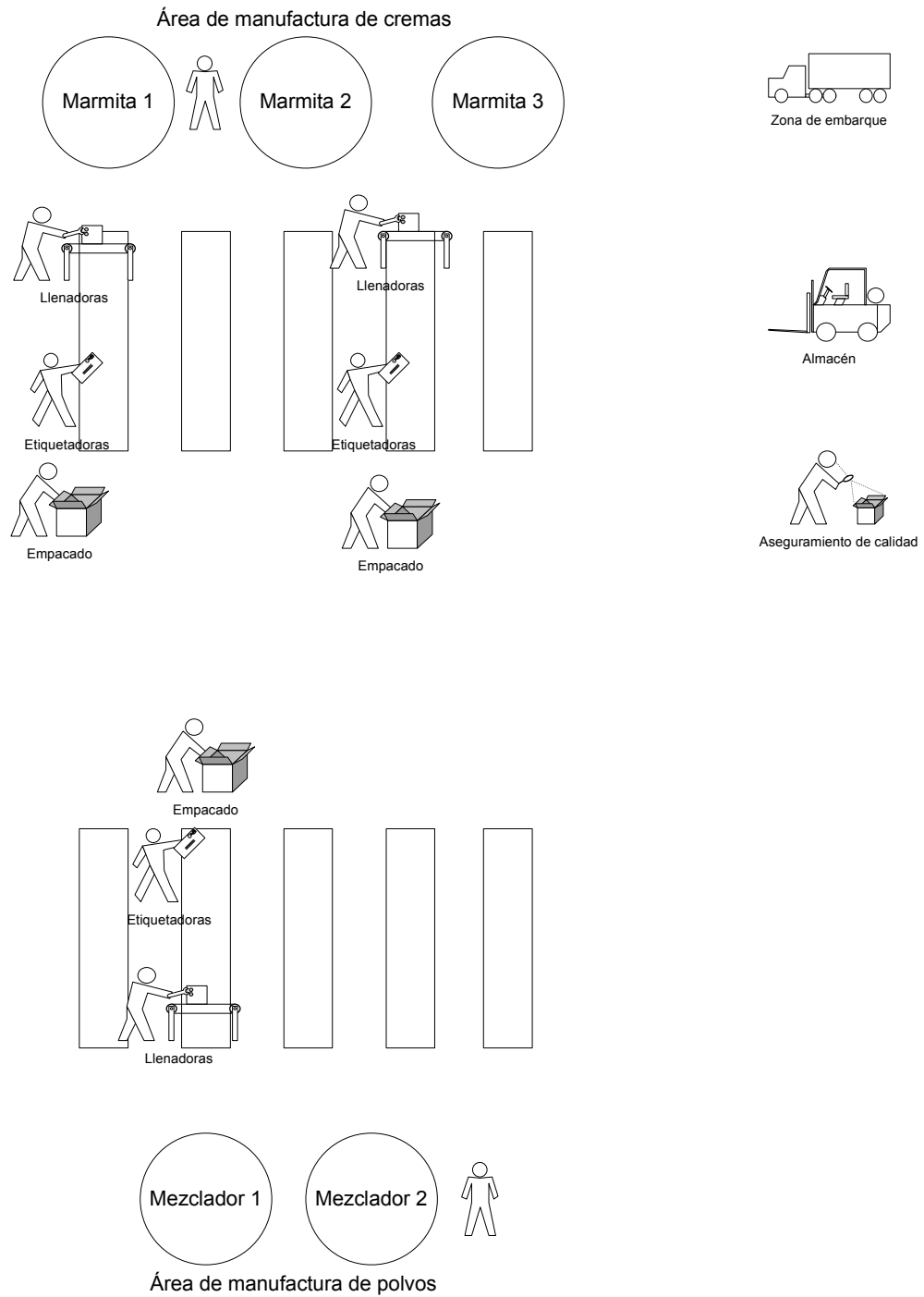


Figura 3. Distribución del área de manufactura.

3.5 Descripción de la metodología

La metodología propuesta se basa en el algoritmo de recocido simulado, mismo que se ha descrito en la sección 2.4.2.4 de este trabajo de investigación y cuya representación en pseudocódigo es la siguiente:

```
RC( )
  Entrada: la instancia de un problema P
  Salida: la mejor solución encontrada para el problema P
   $T \leftarrow T_0$       inicializa temperatura
   $\alpha \leftarrow \alpha_0$   inicializa factor de enfriamiento
   $U \leftarrow U_0$       solución inicial
   $L \leftarrow L_0$       cadena de Markov
  Hacer
    Para  $i=1$  hasta  $i=L$ 
       $V \leftarrow$  perturbación (U)
      Si  $f(V) \leq f(U)$ 
        Entonces  $U \leftarrow V$ 
      Si no
        Si  $e^{-\frac{f(U)-f(V)}{T}} \leq \text{rand}[0,1]$ 
          Entonces  $U \leftarrow V$ 
    Fin para
   $T \leftarrow \alpha T$ 
  Mientras (se cumple criterio de terminación)
```

En el Anexo 1 se muestra el código fuente para este problema.

3.5.1 Uso del algoritmo de recocido simulado en el método propuesto.

El algoritmo de recocido simulado comienza con una instancia previamente obtenida con el generador de números aleatorios del procesador de hojas de cálculo de Microsoft Excel. Esta instancia inicial consta de dos matrices, la que representa el flujo de los trabajos, es decir las máquinas en que deberán procesarse las operaciones de cada trabajo, y la matriz de tiempos de los trabajos, que corresponde al tiempo de operación para cada trabajo J_i de acuerdo con la máquina que lo procesa.

3.5.1.1 Creación de la solución inicial.

- a) De acuerdo con el tamaño del problema se calcula el número de celdas de la cadena que representa la solución inicial, multiplicando $m \times n$. Ej. Si $m=4$ y $n=4$, entonces la longitud de la cadena es 16.
- a) Se llena la cadena de manera ordenada, de manera que queden representadas todas las operaciones. El número de operaciones de cada trabajo n indica el número de veces que el trabajo n deberá ser representado en una celda. Para el ejemplo anterior, el problema 4×4 , con 4 operaciones en cada trabajo J_i ($O_{i0}, O_{i1}, O_{i2}, O_{i3}$), ocurre de la siguiente manera:

0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Figura 4. Representación inicial en cadena del problema.

Donde:

- Los cuatro 0's corresponden a las 4 operaciones del J_0 .
- Los cuatro 1's corresponden a las 4 operaciones del J_1 .

- Los cuatro 2's corresponden a las 4 operaciones del J_2 .
- Los cuatro 3's corresponden a las 4 operaciones del J_3 .

b) Se aplican $n \times m$ cambios aleatorios a la cadena obtenida en el paso anterior. Cada cambio aleatorio consiste en la selección aleatoria de dos posiciones de la cadena y su intercambio entre ellas. La función de intercambio valida que no se haga el intercambio tomando las dos veces el mismo valor de posición.

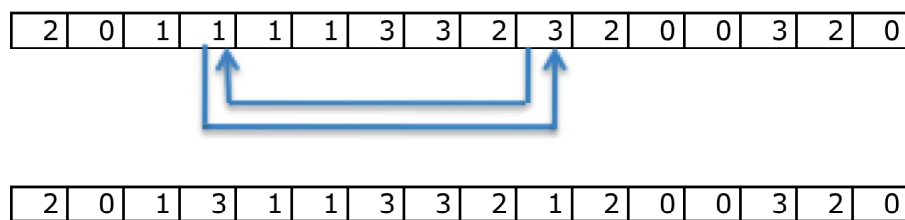


Figura 5. Representación de la función de intercambio aleatorio.

Esta función se ejecuta por única vez al iniciar el algoritmo. Esta solución inicial toma el nombre de “solución actual”.

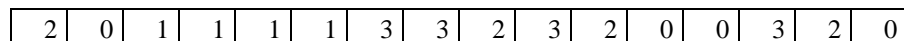


Figura 6. Representación de la solución inicial de un problema.

3.5.1.2 Creación de soluciones “vecinas”

a) Se toma la solución actual y se generan N soluciones alternas. Para generar soluciones alternas se toma la solución actual previamente definida haciéndole un solo intercambio aleatorio para crear cada una de éstas soluciones alternas.

- b) La función de intercambio que se usa consiste en seleccionar aleatoriamente 2 posiciones de la cadena de la solución actual e intercambiar sus valores correspondientes. La función garantiza que no se haga el intercambio tomando dos veces el mismo valor de posición.

3.5.1.3 Función de evaluación

La función objetivo para este problema de planificación de tareas es la siguiente:

$$Z = \min C_{\max}$$

Que se refiere a minimizar el tiempo requerido para terminar todos los trabajos (Makespan). Donde C_{\max} es el valor máximo entre los tiempos de ejecución de cada uno de los trabajos en el sistema:

$$C_{\max} \leftarrow \text{Max}\{C_0, C_1, C_2, \dots, C_n\}$$

Con el propósito de encontrar soluciones factibles de buena calidad, se desarrolló la siguiente función de evaluación:

.Función Objetivo()

Entrada: Una posible asignación de tareas S .

Salida: C_{\max} de las cadenas de trabajos.

Para cada secuencia de operaciones calcula C_i

$$C_{\max} \leftarrow \text{Max}\{C_0, C_1, C_2, \dots, C_n\}$$

Devuelve C_{\max}

Donde:

- S es un vector de tamaño $m \times n$ que representa la secuencia de una instancia del problema.
- C_{\max} es el tiempo de completamiento máximo
- C_i es el tiempo de completamiento para el trabajo i

3.5.1.4 Diagrama de Flujo

A continuación se muestra el diagrama de flujo del algoritmo de recocido simulado tal como es utilizado en el método propuesto:

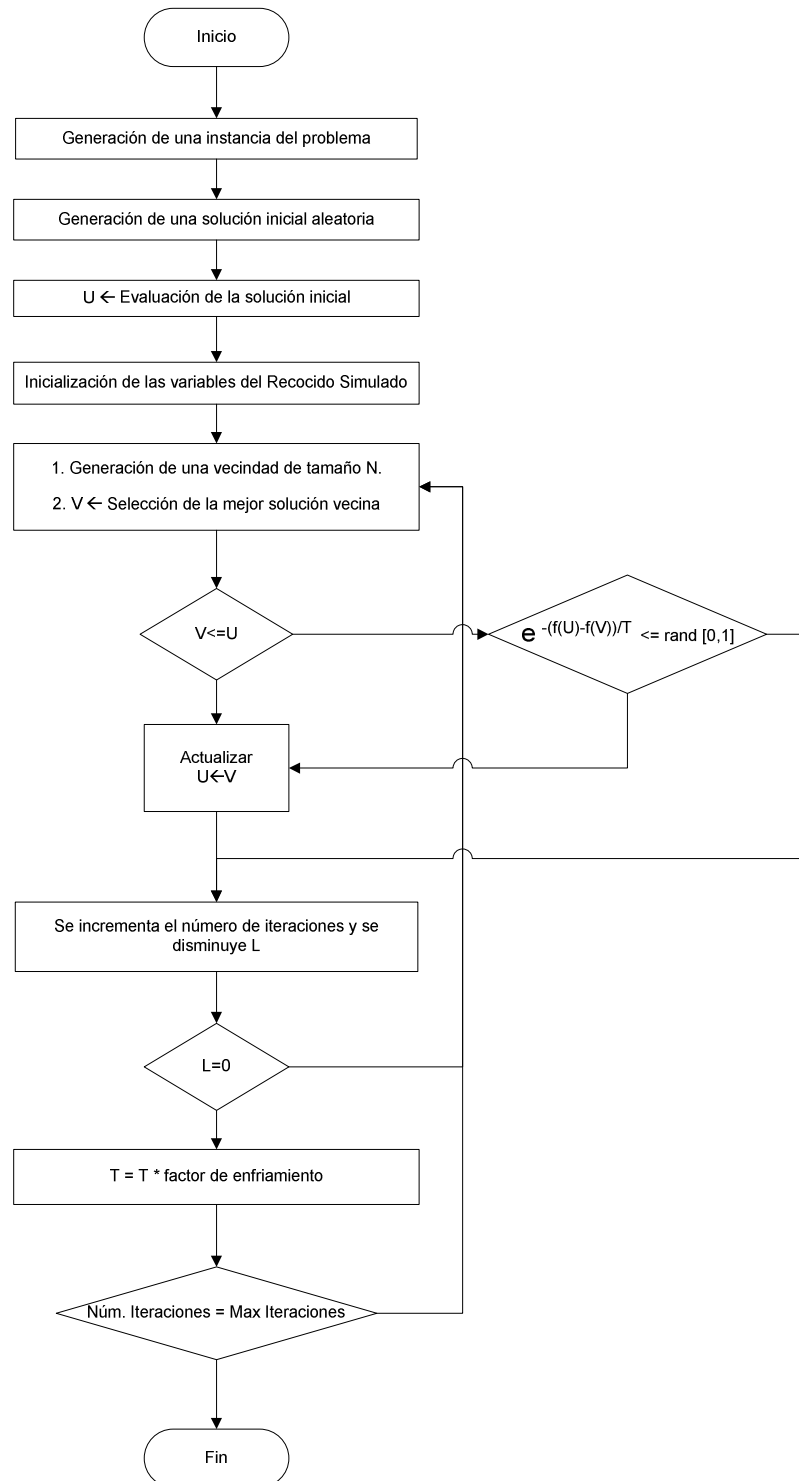


Figura 7. Diagrama de Flujo del algoritmo de recocido simulado para el método propuesto.

3.5.2 Elección de factores y niveles.

Las principales variables del recocido simulado que se determinan en una implementación de la metaheurística son las siguientes:

- Una temperatura inicial T_0
- Un factor de enfriamiento α
- La longitud de la cadena de Markov (L)
- Número máximo de iteraciones
- Tamaño de la vecindad (N)

Para definir el valor de estas variables, se eligieron 3 niveles diferentes a cada variable, los cuales se muestran en la tabla.

Temperatura T	Factor de enfriamiento α	Cadena de Markov L	Número máximo de iteraciones	Tamaño de la vecindad N
0.5	0.71	$n \times m$	$10n \times m$	5
1	0.80	$5n \times m$	$20n \times m$	10
2	0.98	$15n \times m$	$40n \times m$	20

Tabla 3. Factores y niveles

Estos valores se eligieron mediante la observación empírica del comportamiento del problema. Mediante el uso del software minitab, se eligieron los valores más adecuados para este tipo de problema. Los valores escogidos para la experimentación son los siguientes:

- Una temperatura inicial $T_0=1$
- Un factor de enfriamiento $\alpha=0.98$
- La longitud de la cadena de Markov $L=n \times m$

- Número máximo de iteraciones = $10n \times m$
- Tamaño de la vecindad $N=10$

3.5.3 Variable de respuesta.

La variable de respuesta es el mejor valor obtenido para el tiempo de completamiento para todas las operaciones de todos los trabajos (C_{max}). A este C_{max} le corresponde una secuencia de operaciones para la cuál en este trabajo se utilizará la representación en cadena como formato de salida para el resultado de la ejecución del algoritmo:

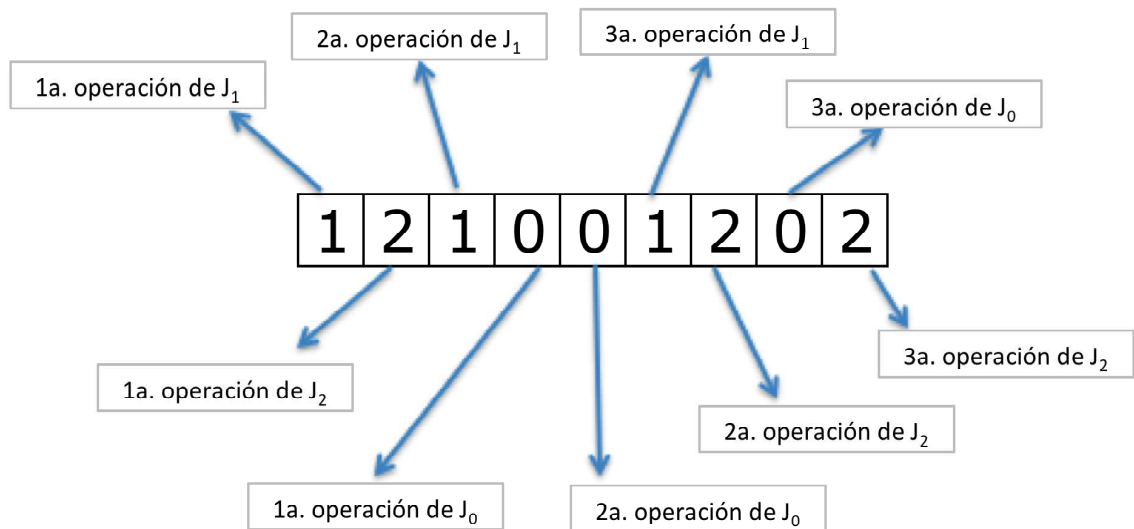


Figura 8. Representación en cadena del problema.

Donde:

- El primer elemento de la cadena corresponde a la primera operación de J_1 (O_{11})
- El segundo elemento de la cadena corresponde a la primera operación de J_2 (O_{21})
- El tercer elemento de la cadena corresponde a la segunda operación de J_1 (O_{12})
- El cuarto elemento de la cadena corresponde a la primera operación de J_0 (O_{01})

- El quinto elemento de la cadena corresponde a la segunda operación de $J_0 (O_{02})$
- El sexto elemento de la cadena corresponde a la tercera operación de $J_1 (O_{13})$
- El séptimo elemento de la cadena corresponde a la segunda operación de $J_2 (O_{22})$
- El octavo elemento de la cadena corresponde a la tercera operación del $J_0 (O_{03})$
- El noveno elemento corresponde a la tercera operación de $J_2 (O_{23})$

La representación en cadena es una manera compacta y sencilla de expresar las diferentes soluciones del problema. En esta representación de tamaño $m \times n$ se distribuyen las operaciones de todos los trabajos a lo largo de todo el arreglo, mediante celdas que representan una operación cada una, y cuyo orden de aparición indica el orden en que irán siendo procesadas.

Esta representación, además de ser compacta y sencilla, permite cumplir con las restricciones de precedencia y evitar traslapes entre las operaciones de un mismo trabajo.

3.6 Instancias

La intención de este trabajo fue utilizar instancias de tamaño real, tal como los que se presentan en la empresa objeto de estudio. Las diferentes instancias se diseñaron con el generador de números aleatorios del procesador de hojas de cálculo de Microsoft Excel, dentro de un rango establecido que comprende la variación real para dichos tamaños de problema dentro del ambiente de manufactura de la compañía objeto de estudio, con diferentes tamaños $m \times n$ simétricos de 3×3 , 4×4 , 5×5 , 6×6 y 7×7 y asimétricos de tamaño 3×4 , 4×5 , 5×6 , 6×7 y 7×8 . Para cada uno de estos tamaños de problema se diseñaron 10 instancias diferentes.

Se elaboró además, de manera manual, una instancia para cada tamaño de problema a fin de contar con un punto de referencia para comparar los resultados que brinda el método de solución propuesto respecto a la metodología que actualmente se utiliza en la empresa.

Con el fin de probar la capacidad del método propuesto para problemas de mayor tamaño, se resolvieron además una instancia con tamaño $m=20$ y $n=15$ y otra instancia con tamaño $m=100$ y $n=20$.

Capítulo 4

Discusión de resultados.

Siguiendo la metodología descrita en el capítulo anterior, se desarrolló una implementación de los algoritmos y procedimientos que la integran, empleando una plataforma para el desarrollo de aplicaciones de software que utiliza componentes en módulos de lenguaje Java.

Utilizando esta implementación, se realizó un trabajo experimental con las instancias obtenidas descritas en la sección 3.6. A manera de tabla se exponen las características de las instancias evaluadas y los resultados obtenidos de la aplicación del método propuesto. La prueba realizada con cada una de las instancias que corresponden a los diferentes tamaños de problema consta de 5 ejecuciones del algoritmo así como la elaboración manual de la secuencia.

Para evaluar el desempeño del método propuesto se compararon los valores obtenidos de C_{max} y tiempo de ejecución del método propuesto contra los valores que arrojó la elaboración manual de secuencias de operaciones para cada uno de los tamaños de problema.

Mediante pruebas de hipótesis se pudo confirmar el impacto que el uso del método propuesto tiene en la elaboración de secuencias de producción para un taller de manufactura.

4.1 Resultados obtenidos

A continuación se presentan los resultados obtenidos de la aplicación del método propuesto para cada una de las instancias y tamaños de problema. Se indica también el resultado que se obtuvo de la elaboración manual de una instancia para cada tipo de problema.

Nombre problema	m	N	Cmax (hr)	Solución Manual Cmax (hr)	Tiempo ejecución (seg.)	Tiempo ejecución manual (seg.)	Tipo
033	3	3	15	12	3	240	Simétrica
133	3	3	9	12	3	240	Simétrica
233	3	3	13	12	2	240	Simétrica
333	3	3	14	12	2	240	Simétrica
433	3	3	17	12	3	240	Simétrica
533	3	3	10	12	2	240	Simétrica
633	3	3	14	12	2	240	Simétrica
733	3	3	11	12	2	240	Simétrica
833	3	3	14	12	1	240	Simétrica
933	3	3	10	12	2	240	Simétrica
044	4	4	36	32	1	420	Simétrica
144	4	4	36	32	1	420	Simétrica
244	4	4	30	32	0	420	Simétrica
344	4	4	28	32	0	420	Simétrica
444	4	4	24	32	0	420	Simétrica
544	4	4	25	32	0	420	Simétrica
644	4	4	31	32	0	420	Simétrica
744	4	4	27	32	0	420	Simétrica
844	4	4	33	32	0	420	Simétrica
944	4	4	30	32	0	420	Simétrica
055	5	5	34	39	0	540	Simétrica
155	5	5	37	39	2	540	Simétrica
255	5	5	48	39	0	540	Simétrica
355	5	5	48	39	0	540	Simétrica
455	5	5	38	39	0	540	Simétrica
555	5	5	53	39	2	540	Simétrica
655	5	5	35	39	2	540	Simétrica
755	5	5	49	39	2	540	Simétrica
855	5	5	56	39	3	540	Simétrica
955	5	5	35	39	2	540	Simétrica
066	6	6	55	58	0	720	Simétrica
166	6	6	65	58	0	720	Simétrica
266	6	6	52	58	3	720	Simétrica
366	6	6	55	58	0	720	Simétrica
466	6	6	59	58	2	720	Simétrica
566	6	6	61	58	2	720	Simétrica
666	6	6	54	58	2	720	Simétrica
766	6	6	55	58	6	720	Simétrica
866	6	6	49	58	2	720	Simétrica
966	6	6	55	58	2	720	Simétrica
077	7	7	69	72	3	960	Simétrica
177	7	7	75	72	2	960	Simétrica
277	7	7	77	72	1	960	Simétrica
377	7	7	77	72	1	960	Simétrica

Nombre problema	m	N	Cmax (hr)	Solución Manual Cmax (hr)	Tiempo ejecución (seg.)	Tiempo ejecución manual (seg.)	Tipo
477	7	7	75	72	3	960	Simétrica
577	7	7	75	72	3	960	Simétrica
677	7	7	84	72	2	960	Simétrica
777	7	7	84	72	2	960	Simétrica
877	7	7	71	72	2	960	Simétrica
977	7	7	63	72	2	960	Simétrica
034	3	4	21	24	1	300	Asimétrica
134	3	4	26	24	1	300	Asimétrica
234	3	4	34	24	0	300	Asimétrica
334	3	4	30	24	0	300	Asimétrica
434	3	4	25	24	0	300	Asimétrica
534	3	4	21	24	0	300	Asimétrica
634	3	4	28	24	0	300	Asimétrica
734	3	4	18	24	1	300	Asimétrica
834	3	4	24	24	0	300	Asimétrica
934	3	4	28	24	1	300	Asimétrica
045	4	5	42	35	1	480	Asimétrica
145	4	5	26	35	1	480	Asimétrica
245	4	5	39	35	2	480	Asimétrica
345	4	5	32	35	0	480	Asimétrica
445	4	5	35	35	1	480	Asimétrica
545	4	5	35	35	1	480	Asimétrica
645	4	5	32	35	1	480	Asimétrica
745	4	5	30	35	1	480	Asimétrica
845	4	5	30	35	1	480	Asimétrica
945	4	5	30	35	1	480	Asimétrica
056	5	6	40	45	1	600	Asimétrica
156	5	6	55	45	1	600	Asimétrica
256	5	6	43	45	1	600	Asimétrica
356	5	6	49	45	0	600	Asimétrica
456	5	6	46	45	1	600	Asimétrica
556	5	6	46	45	1	600	Asimétrica
656	5	6	50	45	1	600	Asimétrica
756	5	6	51	45	1	600	Asimétrica
856	5	6	45	45	1	600	Asimétrica
956	5	6	46	45	1	600	Asimétrica
067	6	7	68	61	1	840	Asimétrica
167	6	7	58	61	1	840	Asimétrica
267	6	7	55	61	1	840	Asimétrica
367	6	7	55	61	1	840	Asimétrica
467	6	7	61	61	1	840	Asimétrica
567	6	7	56	61	1	840	Asimétrica
667	6	7	59	61	1	840	Asimétrica
767	6	7	62	61	1	840	Asimétrica

Nombre problema	m	n	Cmax (hr)	Solución Manual Cmax (hr)	Tiempo ejecución (seg.)	Tiempo ejecución manual (seg.)	Tipo
867	6	7	64	61	1	840	Asimétrica
967	6	7	53	61	0	840	Asimétrica
078	7	8	70	79	6	1080	Asimétrica
178	7	8	81	79	0	1080	Asimétrica
278	7	8	77	79	1	1080	Asimétrica
378	7	8	80	79	1	1080	Asimétrica
478	7	8	67	79	0	1080	Asimétrica
578	7	8	73	79	1	1080	Asimétrica
678	7	8	86	79	1	1080	Asimétrica
778	7	8	78	79	1	1080	Asimétrica
878	7	8	79	79	1	1080	Asimétrica
978	7	8	79	79	1	1080	Asimétrica

Tabla 4. Resultados arrojados por el método propuesto

La columna denominada “Cmax” enlista la mejor solución obtenida para el valor Cmax (tiempo total de proceso) mediante la aplicación del método propuesto para cada una de las instancias, mientras que la columna “Solución Manual Cmax” nos indica el valor Cmax que se obtuvo al elaborar manualmente una secuencia de operaciones del mismo tamaño. Ambos valores están expresados en horas.

La columna “Tiempo ejecución” enlista el tiempo que el método propuesto necesitó para obtener la secuencia de operaciones con el mejor valor Cmax hasta su criterio de paro. La columna “Tiempo ejecución manual” nos indica el tiempo que el método manual necesitó para obtener la secuencia de operaciones con la mejor solución posible para el valor Cmax. Los valores de ambas columnas están expresados en segundos.

Las siguientes gráficas nos muestran el comportamiento del valor Cmax comparando los resultados arrojados por el método propuesto contra los obtenidos manualmente:

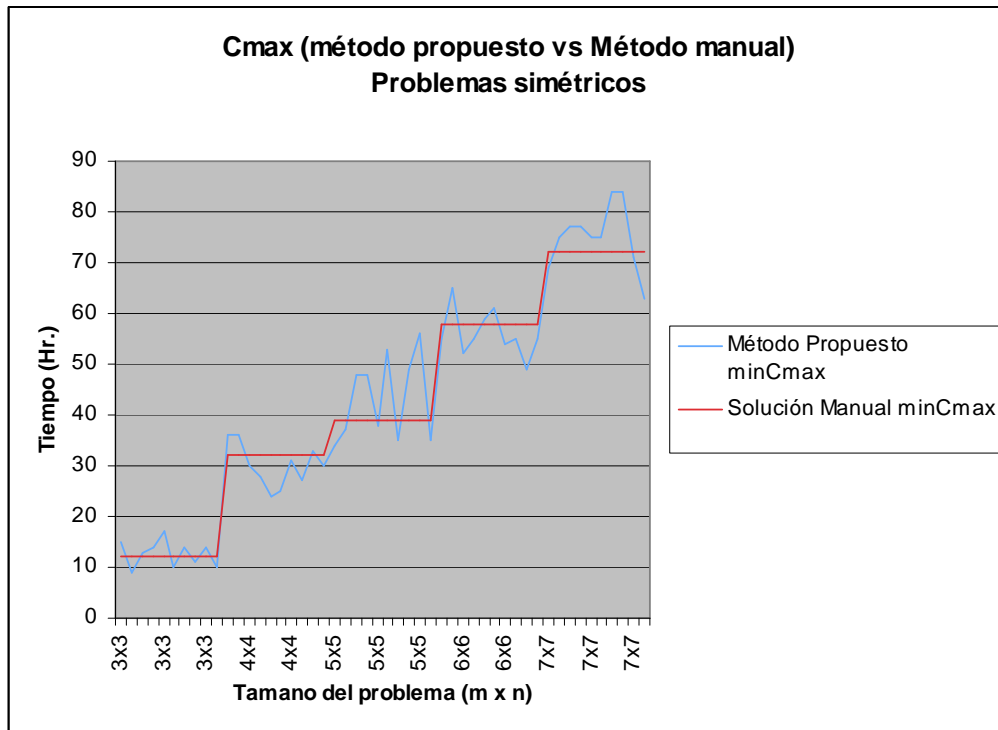


Gráfico 1. Cmax propuesto vs. Cmax manual para problemas simétricos

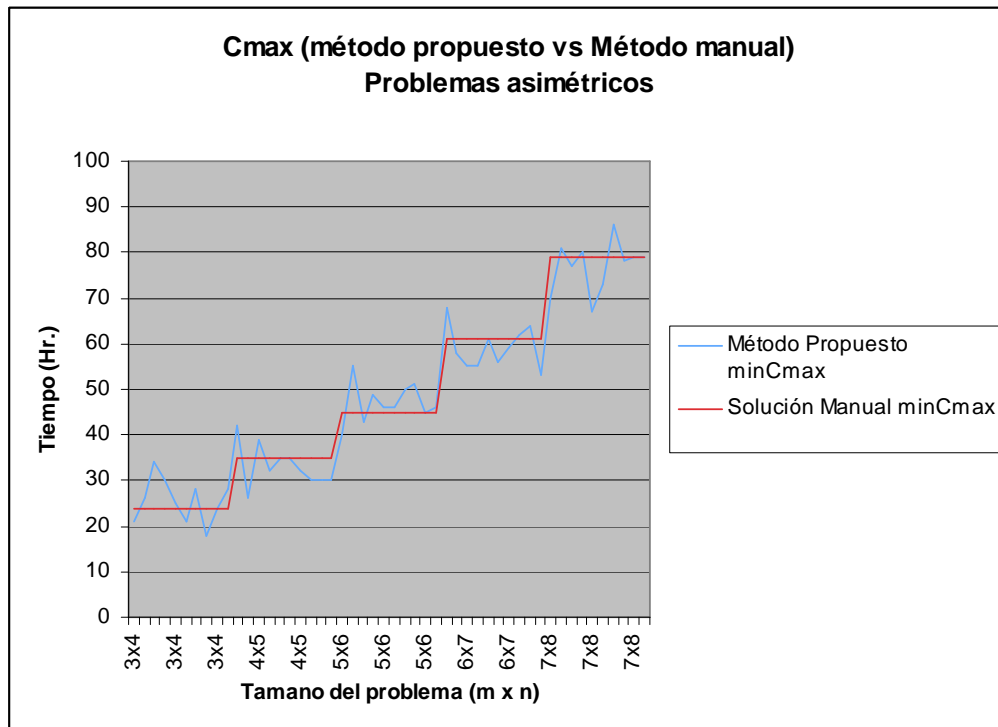


Gráfico 2. Cmax propuesto vs. Cmax manual para problemas asimétricos

Las gráficas siguientes hacen una comparación entre el tiempo de ejecución de ambos métodos, es decir, el tiempo de generación de secuencias factibles:

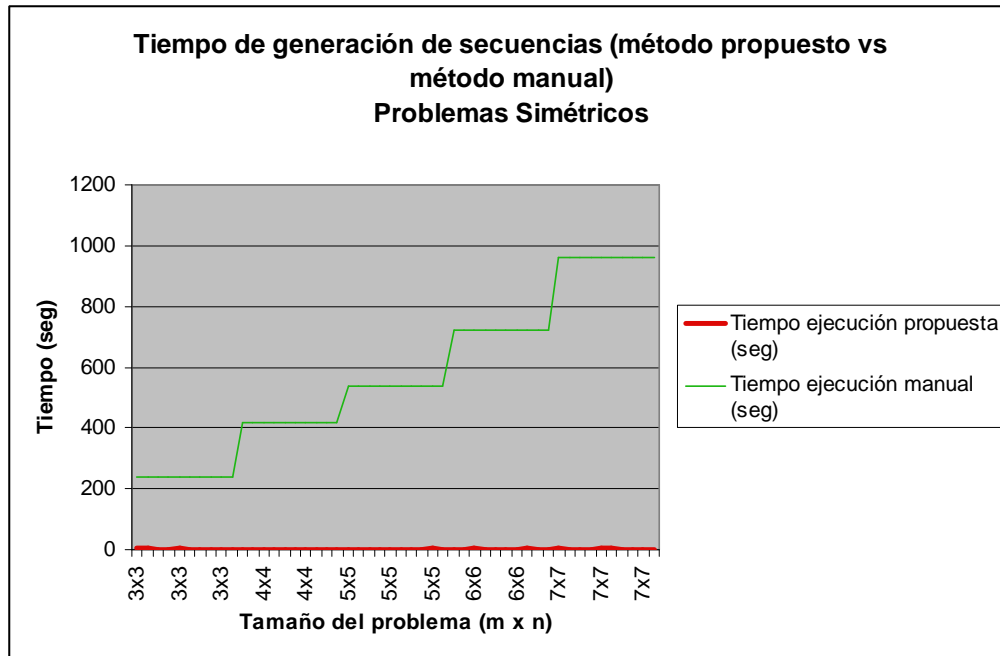


Gráfico 3. Tiempo de ejecución propuesto vs. Tiempo de ejecución manual para problemas simétricos.

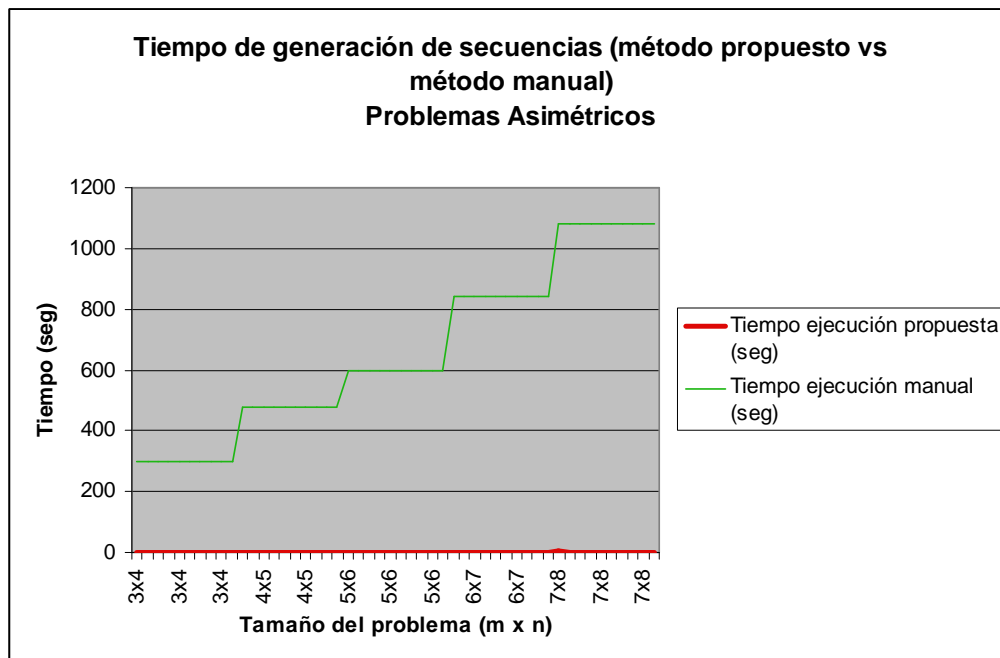


Gráfico 4. Tiempo de ejecución propuesto vs. Tiempo de ejecución manual para problemas asimétricos.

4.2 Pruebas de hipótesis

Se realizaron dos tipos de pruebas de hipótesis con los datos obtenidos en la experimentación.

4.2.1 Prueba de hipótesis de tiempo total de ejecución del método.

La primera prueba de hipótesis consistió en comparar el tiempo de generación de secuencias válidas para ambos métodos (elaboración manual y método propuesto).

La hipótesis alternativa de esta segunda prueba sugiere que el método propuesto genera secuencias en un tiempo menor comparado con el método manual.

$$H_0: \mu_s = \mu_m$$

$$H_1: \mu_s < \mu_m$$

Donde:

H_0 = Hipótesis nula (de efecto nulo)

H_1 = Hipótesis alternativa

μ_s = Promedio del tiempo requerido (seg.) para la generación de secuencias con el método propuesto.

μ_m = Promedio del tiempo requerido (seg.) para elaborar una secuencia con el método manual.

Los resultados para cada uno de los tamaños de problema que se obtuvieron de esta primera prueba de hipótesis aparecen en la siguiente tabla:

N	m	n	Media Manual (seg.)	Media Propuesta (seg.)	Desv. Std.	Valor P
10	3	3	240	2.2	0.632	0
10	4	4	420	0.2	0.422	0
10	5	5	540	1.3	1.16	0
10	6	6	720	1.9	1.792	0
10	7	7	960	2.1	0.738	0
10	3	4	300	0.4	0.516	0
10	4	5	480	1	0.471	0
10	5	6	600	0.9	0.316	0
10	6	7	840	0.9	0.316	0
10	7	8	1080	1.3	1.703	0

Tabla 5. Resultados de la primera prueba de hipótesis

Para todos los tamaños de problema evaluados, el valor P es menor que 0.05, por lo que se rechaza la hipótesis nula, aceptando con esto la hipótesis alternativa ($H_1: \mu_s < \mu_m$) que señala que sí existe una diferencia en el tiempo de generación de secuencias con el método propuesto y que este es significativamente menor.

4.2.2 Prueba de hipótesis de tiempo total de proceso obtenido.

La segunda prueba de hipótesis compara el tiempo total de proceso (makespan, Cmax) obtenido con el método propuesto contra el Cmax obtenido mediante la elaboración manual de secuencias, teniendo como hipótesis nula que el método que utiliza el algoritmo de recocido simulado obtiene soluciones tan confiables como las que se obtienen con el método de elaboración manual.

$$H_0: \mu_s = \mu_m$$

$$H_1: \mu_s \neq \mu_m$$

Donde:

H_0 = Hipótesis nula (de efecto nulo)

H_1 = Hipótesis alternativa

μ_s = Promedio del tiempo total de proceso C_{max} (hr.) obtenido con el método propuesto.

μ_m = Promedio del tiempo total de proceso C_{max} (hr.) obtenido con el método manual.

Los resultados obtenidos para cada uno de los tamaños de problema se muestran en la siguiente tabla:

m	n	Media Manual (hr.)	Media Propuesta (hr.)	Desv. Std.propuesta	Valor P
3	3	12	12.7	2.584	0.392
4	4	32	30	4.16	0.129
5	5	39	43.3	8.33	0.102
6	6	58	56	4.57	0.166
7	7	72	75	6.38	0.137
3	4	24	25.5	4.77	0.32
4	5	35	33.1	4.75	0.206
5	6	45	47.1	4.28	0.121
6	7	61	59.1	4.68	0.199
7	8	79	77	5.58	0.257

Tabla 6. Resultados de la segunda prueba de hipótesis

La columna “Media Manual” enlista los valores promedio obtenidos para las poblaciones que conforman los diferentes tamaños de problema mediante la elaboración manual de secuencias. Las columnas “Media propuesta” y “Desviación estándar propuesta” son los valores del promedio y desviación estándar obtenidos mediante el método propuesto.

Para todos los tamaños de problema analizados, como el valor P es mayor que 0.05, se acepta la hipótesis nula, lo que significa que no hay diferencia significativa entre los resultados obtenidos con el método propuesto comparado con el método manual.

Los resultados para cada uno de los grupos que conforman los diferentes tamaños de las instancias analizadas indican que el desempeño del método propuesto es aceptable ya que arroja soluciones tan confiables como las que se obtienen utilizando el método manual.

4.3 Análisis de capacidad del método propuesto

Con el fin de verificar si el método propuesto es capaz de solucionar instancias de mayor tamaño, se hicieron dos corridas con tamaños de problema 20x15 y 100x20 arrojando los siguientes resultados.

- $m=20$ y $n=15$: El método tardó 10 segundos en encontrar una solución cuyo valor de C_{max} fue de 634 horas.
- $m=100$ y $n=20$: Para este tamaño de problema, el método tardó 8 minutos con 58 segundos en obtener una solución, cuyo valor de C_{max} fue de 14,756 horas.

Estos resultados obtenidos, si bien no llegan hasta el punto de stress máximo para el método, demuestran que su capacidad de éste es suficiente para resolver la mayoría de los tamaños de problema que se dan en un ambiente de manufactura de una empresa de tamaño y tipo como la estudiada en este trabajo.

Capítulo 5

Conclusiones y trabajo futuro.

5.1 Conclusiones

Tal como se expuso en capítulos anteriores, la planificación de tareas constituye una de las herramientas más importantes para la optimización de recursos dentro de una empresa.

El objetivo de este trabajo fue proponer un método inspirado en un algoritmo metaheurístico, que principalmente permitiera:

- Obtener soluciones confiables, equiparables a las obtenidas por el método de elaboración manual de secuencias que es el que actualmente se utiliza en la compañía objeto de estudio.
- Reducir el tiempo dedicado a la elaboración de secuencias factibles.
- Conocer el orden en que los trabajos deben ser ejecutados para reducir el tiempo total de completamiento de todas las operaciones (makespan).

Se encontró que el método propuesto tiene un buen desempeño en comparación a los métodos que actualmente se utilizan, resultando un valor de C_{max} equiparable al que se obtiene mediante el método manual actualmente utilizado por lo que se puede decir que arroja soluciones confiables. Con esto se da respuesta a la primera pregunta de investigación.

La segunda pregunta de investigación halla respuesta en el hecho de que se pudo observar que el tiempo de generación de secuencias fue significativamente menor al tiempo que consume el método manual, por lo que su desempeño en este aspecto fue superior.

La respuesta a la tercer pregunta de investigación se obtiene en cada ejecución del algoritmo, ya que cada solución arrojada indica un orden específico en que deben programarse las operaciones; esta información es utilizada por el área de manufactura, ya que indica el plan de producción que deberá seguir, cuáles son y en qué tiempo deben realizarse los trabajos en determinada máquina.

Esta solución obtenida por el método propuesto, traducida en un plan de producción, está enfocada a reducir el tiempo de completamiento de todas las operaciones y con esto, apoyar en el cumplimiento de la fecha de entrega del producto que fue acordada con el cliente.

Se concluye que mediante la utilización de este método si es posible optimizar el proceso de planeación de la producción a corto plazo mediante la reducción de tiempo dedicado a la generación de secuencias factibles de operaciones. Debido a su desempeño superior, se considera como una buena opción para sustituir al método que actualmente se utiliza.

5.2 Trabajo futuro.

El método propuesto en este trabajo se enfocó en la resolución de un solo objetivo: la minimización del tiempo total de completamiento de los trabajos en un taller de manufactura. Parte del trabajo futuro posible es rediseñar el método para poder incluir planteamientos multiobjetivo del problema de planificación de tareas.

Es posible extender este método a tamaños de problema más grandes o incrementando las variables de restricción. También los tiempos de preparación de máquina (que en este trabajo están dentro del tiempo de ejecución de cada operación) pueden ser incluidos de manera que dependan del orden de ejecución de las tareas, y que mediante la adecuada programación de los trabajos puedan reducirse al programar adyacentemente lotes de productos similares que no requieran grandes cambios en el formato de las máquinas.

Anexo 1

Código fuente.

```
package jssp;

public class Main {

    int [][] OT = {{3,3,3,3},    //This matrix represents the Operation Time for each Ji according its machine
                  {2,3,4,2},
                  {3,2,1,4}};

    int [][] JC = {{0,1,2,3},    //Job Chain for the Ji job
                  {0,2,1,3},
                  {1,0,2,3}};

    private static final int N=10;    //Neighborhood size
    private static final int MAXINT=999999;
    private int maxIt;    //Max iter

    double T;    //Temperature
    double alpha;    //Cooling factor
    int L;    //Markov chain
    int m = 4;    //machines
    int n = 3;    //Jobs
    int mut = 1;    //Mutation size

    int [] S = new int[n*m];    //Solution vector
    int [][] SS = new int[N][n*m];    //Matrix candidate solutions
```

```

int [] Ci = new int[n];           //Completion time for i job

int [] MT = new int[m];         //Machine time

int [] O = new int[n];         //Operation time secuece for the Oim

void initialSolution(){         //This function creates a random initial solution

    int i,j,nw,ol,r=0;

    for( i = 0; i < n; i++ )

        for( j = 0; j < m; j++ )

            S[r++]=i;

    for(int k = 0 ; k < n*m ; k++){

        java.util.Random a = new java.util.Random();

        nw = a.nextInt(n*m-1);

        do{

            ol = a.nextInt(n*m-1);

        }while(nw == ol);

        int tmp = S[nw];

        S[nw] = S[ol];

        S[ol] = tmp;

    }

}

void printSol(){                //This function prints the current solution

    for(int y = 0 ; y < n*m ; y++) System.out.print(S[y]+" ");

    System.out.println();

}

void rsjssp(){                  //This function is a Simulated Annealing implementation

    java.util.Random nr = new java.util.Random();

    this.T = 1.0; //Initial temperature

```

```

this.alpha = 0.98; //Cooling factor initialization (values [0,1])

this.L = n*m;    //Markov chain initialization

this.maxIt = 10*n*m; //Max iter initialization

int u = 0;    //Value current solution

int v = MAXINT; //Neighbord value

int tmp;

int bs = 0;

initialSolution(); //Creates a new initial solution

u=evalSolution(S); //Evaluates S

System.out.print("Sol: "+u +" Temp: "+T+" ");

    printSol();

do{

    for(int l = 0 ; l < L ; l++){

        createNeighbord();

        for(int k=0;k<N;k++){

            tmp = evalSolution(SS[k]);

            if(tmp<v){ v = tmp; bs=k;}

        }

        if( v <= u){

            updateSolution(bs);

            u = v;

            System.out.print("Sol: "+u +" Temp: "+T+" ");

            printSol();

        }

        else if(Math.exp(-1.0*(double)(u-v)/T)<nr.nextDouble()){ //Stochastic part of Simulated annealing

            updateSolution(bs);

            u=v;

            System.out.print("Sol: "+u +" Temp: "+T+" ");

            printSol();

```



```

    }

    maxIt--;

}

T*=alpha;    //Temperature update every loop of size L
}while(maxIt>0); //While an iteration number is met
}

private void createNeighbord() { //This funtion creates a neighborhood of size N from the current solution

    int ol, nw;                //Every neighbor is created with a sinlge "mut" permutations

    for(int p = 0 ; p < N ; p++){

        for(int i = 0 ; i < n*m ; i++)

            SS[p][i] = S[i];

        for(int k = 0 ; k < mut ; k++){

            java.util.Random a = new java.util.Random();

            nw = a.nextInt(n*m-1);

            do{

                ol = a.nextInt(n*m-1);

            }while(nw == ol);

            int tmp = SS[p][nw];

            SS[p][nw] = SS[p][ol];

            SS[p][ol] = tmp;

        }

    }

}

private int evalSolution(int[] S) { //This function is an objective function to eval the JSSP

    int i,j,Cmax=0;            //Solution

    for(i = 0 ; i < n ; i++){ //

```

```

    Ci[i] = 0;

    O[i] = 0;
}
for(i = 0 ; i < m ; i++ )

    MT[i] = 0;

for(j=0;j<n*m;j++){
    if(MT[JC[S[j]][O[S[j]]]] > Ci[S[j]]){ //Check if the machine is free

        Ci[S[j]] = MT[JC[S[j]][O[S[j]]]] + OT[S[j]][O[S[j]]]; //if the machine is not free wait

        MT[JC[S[j]][O[S[j]]]] = Ci[S[j]];

    }

    else{ Ci[S[j]] += OT[S[j]][O[S[j]]]; //Else use the machine

        MT[JC[S[j]][O[S[j]]]] = Ci[S[j]];

    }

    O[S[j]]++; //Update operation chain for Ji

}

for(i = 0 ; i < Ci.length ; i++) if(Ci[i] > Cmax) Cmax = Ci[i];

return Cmax;

}

private void updateSolution(int k) { //This function updates the new solution

    for(int i = 0 ; i < m*n ; i++) S[i] = SS[k][i];

}

public static void main(String[] args) { //The main function that initialize all the program

    new Main().rsjssp();

}

}

```

Bibliografía

- [1] Aggoune, R. (2004). Two-Job Shop Scheduling Problems with Availability Constraints. International Conference on Autonomous Planning and Scheduling L. d. T. Industrielles, Centre de Recherche Public Henri Tudor.

- [2] Akers, S. B., Jr. and J. Friedman (1955). "A Non-Numerical Approach to Production Scheduling Problems." *Operations Research* 3(4): 429-442.

- [3] Akers, S. B., Jr. (1956). "A Graphical Approach to Production Scheduling Problems." *Operations Research* 4(2): 244-245.

- [4] Anderson, E. J. and M. C. Ferris (1994). "Genetic Algorithms for Combinatorial Optimization: The Assemble Line Balancing Problem." *Inform Journal On Computing* 6(2): 161-173.

- [5] APICS Manual (2002). "Building Competitive Operations, Planning and Logistics: Manufacturing Planning and Controlling", Module 2, Section C.

- [6] Artigues, C. and Lopez P. (2000). "Extending Giffler-Thompson algorithm to generate active schedules for jobshops with sequence-dependent set up times". LIA Report 129, University of Avignon.

- [7] Azizi, N., S. Zolfaghari, et al. (2007). "A New Meta-heuristic Approach for Combinatorial Optimization and Scheduling Problems". 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007), Hawaii, USA.

- [8] Azizi, N., S. Zolfaghari, et al. (2008). Solving a task scheduling problem using genetic algorithms. 18th International Conference on Flexible Automation and Intelligent Manufacturing conference (FAIM2008), Skovde, Sweden.

- [9] Balas, E. (1969). "Machine Sequencing Via Disjunctive Graphs: An Implicit Enumeration Algorithm." *Operations Research* 17(6): 941-957.
- [10] Bancomext (2005). "La industria cosmética en México: Oportunidades en los mercados externos". CANIPEC, Oaxaca, Oax. Junio 2005.
- [11] Batista, D. M., Drummond, A. C., and da Fonseca, N. L. (2008). Scheduling grid tasks under uncertain demands. In *Proceedings of the 2008 ACM Symposium on Applied Computing (Fortaleza, Ceara, Brazil, March 16 - 20, 2008)*. SAC '08. ACM, New York, NY, 2041-2045. DOI= <http://doi.acm.org/10.1145/1363686.1364181>
- [12] Batista, D. M., Drummond, A. C., and da Fonseca, N. L. (2009). Robust scheduler for grid networks. In *Proceedings of the 2009 ACM Symposium on Applied Computing (Honolulu, Hawaii)*. SAC '09. ACM, New York, NY, 35-39. DOI= <http://doi.acm.org/10.1145/1529282.1529289>.
- [13] Bierwirth, C. and D. C. Mattfeld (1999). "Production Scheduling and Rescheduling with Genetic Algorithms." *Evolutionary Computation* 7(1): 1-17.
- [14] Chua, T. J., F. Y. Wang, et al. (2006). "A Heuristics-based Advanced Planning and Scheduling System with Bottleneck Scheduling Algorithm". *Proceedings of 11th IEEE Conference on Emerging Technologies and Factory Automation, ETFA 2006*. September 20-22, 2006, Diplomat Hotel Prague, Czech Republic.
- [15] Coello Coello, C. A. (2008). *Introducción a la Computación Evolutiva*. México D.F., CINVESTAV-INP Departamento de Computación.
- [16] Coello Coello, C. A. (1997). "Búsqueda Tabú: Evitando lo prohibido." *Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios* 5(49): 72-80.
- [17] Cortés Rivera, Daniel (2004). "Un sistema immune artificial para resolver el sistema del Job Shop Scheduling" Depto. De Ingeniería Eléctrica, CINVESTAV-IPN, Sección de Computación, México D.F. Marzo de 2004.

- [18] Fogel, D. B. (2000). "What is evolutionary computation?" *Spectrum, IEEE* 37(2): 26, 28-32.
- [19] Garey, M. R.; Johnson, D.S. et al. (1976). "The Complexity of Flowshop and Jobshop Scheduling." *Mathematics of Operations Research* 1(2): 117-129.
- [20] Garey, M. R. and Johnson, D. S. (1979) "Computers and Intractability; a Guide to the Theory of Np-Completeness." W. H. Freeman & Co.
- [21] Garrido, A. et al. (2000) "Heuristic methods for solving Job-Shop Scheduling Problems". *Proceedings of ECAI-2000 Workshop on New Results in Planning, Scheduling and Design. Berlín.* (ed. By Sauer, Köhler). Pp. 36-43. 2000.
- [22] Gendreau, M; Potvin, J.Y. (2005) "Metaheuristics in combinatorial optimization" *Annals of Operations Research* 140, 189-213, 2005.
- [23] Giffler, B. and G. L. Thompson (1960). "Algorithm for Solving Production Scheduling Problems." *Operations Research* 8: 487-503.
- [24] Glover, F. (1977). "Heuristic for integer programming using surrogate constraints." *Decision Sciences* 8(1): 156-166.
- [25] Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Cambridge, Mass, MIT Press, 1992.
- [26] Jain, A. S. and S. Meeran (1999). "Deterministic job-shop scheduling: Past, present and future." *European Journal of Operational Research* 113(2): 390-434.

- [27] Jain, A. S. and S. Meeran (1998). "A State of the Art Review of Job Shop Scheduling Techniques." Department of Applied Physics, Electronic and Mechanical Engineering; University of Dundee, Scotland, UK.: 48.
- [28] Jackson, J.R., (1956). "An extension of Johnson's result on job lot scheduling." *Naval Research Logistics Quarterly*, 3 : pp. 201-203.
- [29] Johnson, D.S., Aragón, C.R., McGeoch, L.A. and Schevon, C., (1989) "Optimization by Simulated annealing: An experimental evaluation"; Part I, graph partitioning. *Operations Research*, 37, 865-892.
- [30] Johnson, D.S., Aragón, C.R., McGeoch, L.A. and Schevon, C., (1989) "Optimization by Simulated annealing: An experimental evaluation"; Part II, graph coloring and number partitioning. *Operations Research*, 39, 378-406.
- [31] Johnson, S. M. (1954) "Optimal Two- and Three-Stage Production Schedules with Set-Up Times Included", *Naval Research Logistics Quarterly*, vol 1, 61-68.
- [32] Kirkpatrick, S.; Vecchi, M.P.; and Gelatt, C.D., (1983). „Optimization by Simulated Annealing“ *Science, New Series*, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680
- [33] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, MIT Press.
- [34] Lenstra, J. K., Rinnooy, A.H.G. (1984). "New Directions in Scheduling Theory." *Operations Research Letters* 2(6).
- [35] Lee, D.Y.; DiCesare, F., (1994) "Scheduling flexible manufacturing systems using Petri nets and heuristic search," *Robotics and Automation, IEEE Transactions on* , vol.10, no.2, pp.123-132.

- [36] Liu, L. and Y. Xi (2006). A Hybrid Genetic Algorithm for Job Shop Scheduling Problem to Minimize Makespan. 6th World Congress on Intelligent Control and Automation, Dalian, China.
- [37] López de Haro, S.J.; Sánchez Martín, P. y Conde Collado (2004). “Secuenciación de tareas mediante metaheurísticos”. VII Congreso de Ingeniería de Organización. Escuela Técnica Superior de Ingeniería. Universidad Carlos III de Madrid. Leganés (Madrid), 9-10 de Septiembre de 2004.
- [38] Manne, A. S. (1960). "On the Job-Shop Scheduling Problem." *Operations Research* 8(2): 219-223.
- [39] Metrópolis, N; Rosenbluth, Adriana; Rosenbluth, Marshall N; Teller, Augusta H.; Teller, Edward. “Equation of state Calculations by fast computing machines”. *The Journal of Chemical Physics*, 21(6):1087:1092. 1953.
- [40] Morton, T. E. and D. W. Pentico (1993). “Heuristic scheduling systems : with applications to production systems and project management”. New York, John Wiley & Sons, Inc.
- [41] Murray, Walter; Todd, Michael (2005). “Tributes to George Dantzig and Leonid Khachiyan”. *SIAG/OPT Views-and-News*. pp. 1-6, Volume 16, Numbers 1-2, October 2005.
- [42] Osman, I.H. and Kelly, J.P. (eds.) (1996), *Meta-Heuristics: Theory and Applications*, Ed. Kluwer Academic, Boston.
- [43] Peña, V. and L. Zumelzu (2006). *Estado del Arte del Job Shop Scheduling Problem*. U. T. F. S. María and D. d. Informática. Valparaíso, Chile.
- [44] Pinedo, M. (2002). *Scheduling: theory, algorithms, and systems*. Upper Saddle, N.J., Prentice Hall.

- [45] Reeves, C. R., Ed. (1993). Modern heuristic techniques for combinatorial problems. NY, USA, John Wiley & Sons, Inc.
- [46] Santana, L. V. and C. A. Coello Coello (2006). "Una introducción a la computación evolutiva y algunas de sus aplicaciones en Economía y Finanzas." Revista de métodos cuantitativos para la economía y la empresa.: 3-26.
- [47] Schutten, M.J.(1988), "Practical Job Shop Scheduling". Annals of Operations Research 83, pp.161-177.
- [48] Settani, Elena. (2000) "Improving Dorm Room Assignments Using Simulated Annealing". The University of New Mexico; Albuquerque, New Mexico. December 2000.
- [49] Son, Y., J. Venkateswaran, et al. (2001). "Initial research on schedule interface with shop floor control system". Systems, Man, and Cybernetics, 2001 IEEE International Conference on.
- [50] Sule, D. R. (2008). "Production Planning and Industrial Scheduling: examples, case studies and applications". Boca Raton, CRC Press.
- [51] Torrente, Daniela A. y Gusinsky de Gelman, Susana P, (2003). "Globalización e Industria. La competitividad de la industria Argentina y chaqueña". Facultad de Cs. Económicas – Universidad Nacional del Nordeste – UNNE
- [52] Turing, A. M. (1937). "On Computable Numbers, with an Application to the Entscheidungsproblem." Proc. London Math. Soc. s2-42(1): 230-265.
- [53] Van Laarhoven, P.J.M.; Aarts, E.H.L.; and Lenstra. J.K. (1992) "Job Shop Scheduling by Simulated Annealing". Operations Research, 40, 113-125.
- [54] Van Laarhoven, P.J.M.; Aarts, E.H.L., (1987), "Simulated Annealing: Theory and practice (Dordrecht, The Netherlands: Kluwer Academic).

- [55] Vazquez Espí, M. (1994). Recocido Simulado: Un nuevo algoritmo para la optimación de estructuras. Departamento de Estructuras de Edificación. Madrid, España, Escuela Técnica Superior de Arquitectura de Madrid. PhD Thesis: 223.
- [56] Vélez, M.C.; Castro, C.A. Y Maya J. (2003), “Algoritmo de búsqueda aleatoria para la programación de la producción”. Revista Universidad EAFIT, Vol. 39 No. 131, 2003. pp. 76-86
- [57] Weise, Thomas. (2008) “Global Optimization Algorithms: Theory and Applications”. 2nd Edition. Thomas Weise; 2008. [<http://www.it-weise.de/>]
- [58] Zolfaghari, S. and M. Liang (2002). "Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine-grouping problems." International Journal of Production Research 40(9): 2141 - 2158.