

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS ESTADO DE MEXICO



DESARROLLO DE UN ALGORITMO HEURÍSTICO PARA LA
CALENDARIZACIÓN DEL SISTEMA N/M/G/Cm:éx MEDIANTE
LA ASIGNACIÓN DE GRUPOS DE REGLAS DE DESPACHO

TESIS QUE PRESENTA

JOSE CARLOS PEREZ RODRIGUEZ

MAESTRIA EN SISTEMAS DE MANUFACTURA

MSMA, 99

AGOSTO, 2003

RESUMEN

Se han diseñado distintos métodos de calendarización para sistemas de producción tipo *Job Shop* basados en diferentes técnicas, tratando de cumplir con ciertos objetivos establecidos. Una de las técnicas más comunes es mediante algoritmos heurísticos, que tratan de encontrar una solución aceptable y realizable fundamentados en el estudio de problemas y situaciones. Un algoritmo heurístico muy utilizado en la industria es el basado en reglas de despacho, el cual consiste en seleccionar la siguiente operación a calendarizar de acuerdo a algún criterio establecido por una o varias reglas de selección que pueden estar basadas en la experiencia o en estudios, y de acuerdo a las características del problema, le dan prioridad a un trabajo sobre otro u otros. Una regla puede consistir en darle prioridad al trabajo más largo, o más corto, el primero en cumplir su fecha de entrega, o el de la operación siguiente más corta. Una combinación de reglas permite tener más criterios de selección y por lo tanto mejores alternativas. En este trabajo se presenta un algoritmo heurístico de máquinas sin demora basado en reglas de despacho que trata de reducir el tiempo total de fabricación de un conjunto de trabajos en un sistema de producción tipo *Job Shop* o tipo taller. Además, se presenta una recopilación de reglas de despacho y se enseña una nueva regla llamada MTOR, cuyo principal objetivo es ordenar la secuencia de operaciones en la o las máquinas cuello de botella con el fin de evitar que algún trabajo se retrase e incremente el tiempo total de fabricación. Además, esta regla también es aplicada en la calendarización general de un problema, sola o en combinación con otras reglas, con el fin de conocer por completo su comportamiento. Finalmente, se presenta un estudio de selección de reglas más apropiadas que colaboren en la reducción de tiempo de fabricación, para posteriormente, realizar una comparación de reglas y grupos de reglas de despacho con el fin de determinar que combinación de reglas es la más apropiada según un estudio estadístico para reducir el tiempo total de fabricación o $C_{\text{máx}}$.

INDICE

Resumen	1
1. Antecedentes. La industria y el problema de calendarización.	6
2. Estado del Arte de los Métodos de Calendarización.	17
2.1. Calendarización de una sola máquina.	17
2.1.1. Algoritmo de Rama y Límite (Branch and Bound)	18
2.1.2. Algoritmo modificado de Retroceso-Avance (Backward-Forward)	21
2.2. Calendarización de N Trabajos y M Máquinas.	28
2.2.1. Aproximación de Red para la Calendarización de <i>Job Shop</i> .	28
2.2.2. Método heurístico Cambio de Cuello de Botella (Shift Bottleneck Heuristic).	31
2.2.3. Algoritmo Codicioso Heurístico.	42
2.2.4. Algoritmo de Giffler and Thompson.	42
2.2.5. Algoritmo de Generación de Planes sin Retraso.	45
2.2.6. Algoritmo de Generación de Planes sin Retraso Modificado.	47
2.2.7. Algoritmo Genético.	50
2.2.8. Algoritmo Genético Basado en Reglas de Despacho (P-GA)	52
2.2.9. Algoritmo Genético Basado en el Heurístico Cambio de Cuello de Botella (SB-GA)	53
2.3. Método Estocástico de Comparación de Algoritmos.	55
2.4. Conclusiones del Capítulo.	57
3. Planteamiento del Problema y Objetivos.	59
3.1. Planteamiento del Problema.	59
3.2. Objetivos.	61
4. Recopilación de Reglas de Despacho.	62
5. Desarrollo.	82
5.1. Descripción del Algoritmo Heurístico.	82
5.2. Estudios Previos de Reglas de Despacho.	85
5.3. Análisis de Reglas de Despacho.	89
5.4. Regla MTOR.	92
5.5. Análisis de Grupos de Reglas y Generación de Subgrupos.	95

5.6. Programa de Calendarización.	96
6. Resultados del análisis comparativo entre Reglas de Despacho.	98
6.1. Análisis comparativo entre Reglas de Despacho individuales.	98
6.2. Análisis comparativo entre Grupos de Reglas de Despacho.	100
6.3. Análisis de medias y varianza	105
7. Conclusiones.	107
Referencias y Bibliografía	109
Anexo A	110
A1. Lista de Variables.	110
A2. Diagrama de Flujo.	111
A3. Código de Programación.	127
Anexo B	135
Anexo C	137
C1. Problemas de Calendarización	138
C2. Resultados de Problemas de Calendarización	152
Anexo D	157
D1. Resultados de Minitab	158
Anexo E. Derechos de autor	160

ÍNDICE DE FIGURAS

1.1 Diagrama del flujo de información en un sistema de manufactura	7
1.2 Los tres arreglos básicos	10
2.1 Diagrama de Nodos del Algoritmo Rama y Límite	21
2.2 Tiempos conocidos en el diagrama de red.	29
2.3 Diagrama de red de un problema tipo Job Shop.	30
2.4 Conexión entre operaciones de una misma máquina.	30
2.5 Diagrama de red de problema tipo Job Shop.	33
2.6 Iteración 1, solución de la máquina 3.	36
2.7 Iteración 2, solución de las máquinas 2 y 3.	37
2.8 Iteración 2, máquina 2 resuelta, máquina 3 retirada.	38
2.9 Iteración 3, solución de las máquinas 1, 2 y 3.	39
2.10 Iteración 3, solución de las máquinas 1 y 2, máquina 3 retirada.	40
2.11 Iteración 3, solución de las máquinas 1 y 3, máquina 2 retirada.	41
2.12 Grafica de Gantt de solución obtenida por Giffler & Thompson.	44
2.13 Grafica de Gantt. Método Generación de Planes sin Retraso.	47
2.14 Grafica de Gantt de solución obtenida por Dorantes-Gálvez.	49
2.15 Enumeración Genética.	50
2.16 Cruzamiento simple entre dos cadenas.	52
2.17 Mutación de una cadena.	53
2.18 Árbol de enumeración de M^* .	54
4.1 Reglas FCFS y FISFS.	66
4.2 Regla WINQ.	72
5.1 Tiempos de ciclo promedio de SPT-AWTL.	86
5.2 Tiempos de ciclo promedio de SPT-MWTL.	86
5.3 Medias y varianzas de las reglas en dos modos de fabricación.	88
5.4 Diagrama de Venn de reglas de despacho.	90
5.5 Grafica de Gantt. Algoritmo de Máquinas sin demora.	94
5.6 Estructura del programa de calendarización.	97
6.1 Comparación de $C_{máx}$ entre reglas de despacho individuales.	99
6.2 Comparación entre reglas y grupos de reglas de despacho.	100

6.3 Efectos de los grupos A con grupos de cuellos de botella.	102
6.4 Resultados de $C_{m\acute{a}x}$ ordenados de acuerdo a la desviaci3n estandar.	102
6.5 Resultados de $C_{m\acute{a}x}$ ordenados de acuerdo a la media.	103
6.6. Comparaci3n de porcentajes de mejores resultados de $C_{m\acute{a}x}$.	104
6.7 Gráfica de medias de Grupos de Reglas de Despacho.	106

ÍNDICE DE TABLAS

1.1 Clasificaci3n de las metodologías de calendarizaci3n.	12
1.2 Clasificaci3n de los niveles de calendarizaci3n.	13
2.1 Ejemplo para Algoritmo Rama y Límite.	20
2.2 Datos para problema de calendarizaci3n.	25
2.3 Datos para construcci3n de red.	29
2.4 Datos para ejemplo del Método Cambio de Cuello de Botella	33
2.5 Soluci3n de un problema 3x3 por el Método de Giffler & Thompson.	44
2.6 Soluci3n de un problema 3x3 por el Método Generaci3n de Planes sin Retraso.	46
2.7 Soluci3n de un problema 3x3 por el Método Dorantes-Gálvez.	49
4.1 Datos para ejemplo de calendarizaci3n.	65
4.2 Datos para ejemplo de calendarizaci3n.	67
4.3 Datos para ejemplo de calendarizaci3n.	70
4.4 Datos para ejemplo de calendarizaci3n.	75
5.1 Reglas incluidas en cada conjunto del diagrama de Venn.	90
5.2 Soluci3n de un problema 3x3 por Algoritmo de Máquinas sin Demora.	94

1. ANTECEDENTES.

LA INDUSTRIA Y EL PROBLEMA DE CALENDARIZACIÓN.

En la actualidad, uno de los principales factores que debe cumplir una empresa es la velocidad de respuesta a una demanda; además de la flexibilidad, la calidad, el volumen de producción y el costo. La empresa que no cumple con los estándares necesarios, a un corto plazo sus productos no podrán competir en el mercado.

Los clientes cada día son más exigentes y basan su elección en elaborados estudios de rentabilidad sobre diferentes opciones, por lo que es necesario modificar la estrategia de manufactura flexibilizando la producción, reduciendo los tiempos de fabricación y de entrega, aumentando la calidad, y ofreciendo el producto a un costo competitivo.

Entre los sistemas más importantes que ayudan a hacer más eficiente a una empresa, se encuentran los sistemas de comunicación que aseguran su correcto funcionamiento. En la Figura 1.1, se muestra el flujo de información entre las diferentes funciones de la empresa.

Una fábrica moderna frecuentemente tiene un sistema de información elaborado que incluye una computadora central y una base de datos. El software que controla tan elaborado sistema de información es generalmente llamado un sistema de planeación de los recursos de la empresa (ERP).

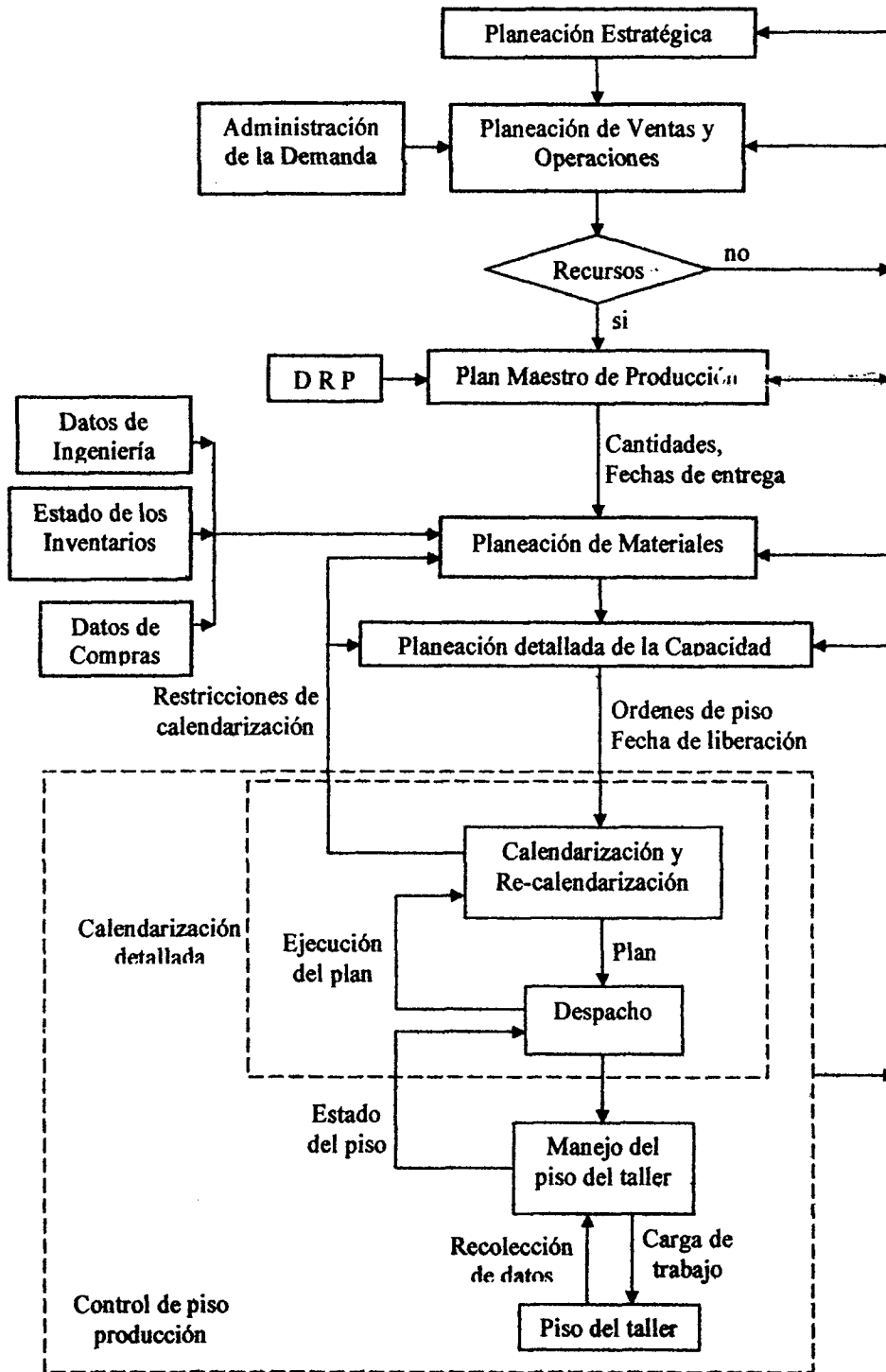


Fig. 1.1 Diagrama del flujo de información en un sistema de manufactura.

El plan maestro de producción (MPS), representa lo que la compañía planea producir en configuraciones, cantidades y fechas específicas; y debe de considerar los pronósticos, el plan de producción, disponibilidad de la capacidad, políticas, e inventarios.

El inventario es un recurso almacenado que es utilizado para satisfacer una necesidad actual o futura. El inventario puede servir para muchas funciones importantes que agregan flexibilidad en la operación de la empresa. Seis usos del inventario son:

1. Proporcionar bienes en existencia para conocer la demanda anticipada de los consumidores.
2. Para separar los procesos de producción y distribución. Por ejemplo, si la demanda de un producto es alta y sólo durante el verano, una compañía puede almacenar durante el invierno y así evitar los costos de escasez en verano.
3. Tomar ventaja de descuentos por cantidad porque las compras en grandes cantidades puede reducir substancialmente el costo de los bienes.
4. Poner una valla contra la inflación y el cambio de precios.
5. Proteger contra escasez que pueda ocurrir debida al mal tiempo, problemas de calidad o entregas inapropiadas. "Existencias de seguridad" pueden reducir el riesgo de escasez.
6. Permitir que las operaciones continúen suavemente con el uso del inventario de trabajo en proceso.

Las compañías mantienen cuatro tipos de inventarios: inventario de materia prima; inventario de trabajo en proceso; inventario de suministros de mantenimiento, reparación y operación (MRO); e inventario de bienes terminados.

El inventario de materia prima ha sido comprado, pero no procesado. Las existencias pueden ser usadas para separar al proveedor del proceso de producción. Sin embargo, la aproximación preferida es eliminar la variabilidad del proveedor en cuanto a calidad, cantidad o tiempo de entrega con el fin de que la separación no sea necesaria.

El inventario de trabajo en proceso (WIP) ha tenido algunos cambios, pero el trabajo no se ha terminado, y existe debido al tiempo necesario para fabricar el producto (llamado tiempo de ciclo). Reduciendo el tiempo de ciclo, se reduce el inventario.

El inventario de suministros de Mantenimiento, de Reparación y Operación existe por la necesidad de reducir el tiempo para mantenimiento y reparación, que en algunos equipos es desconocido.

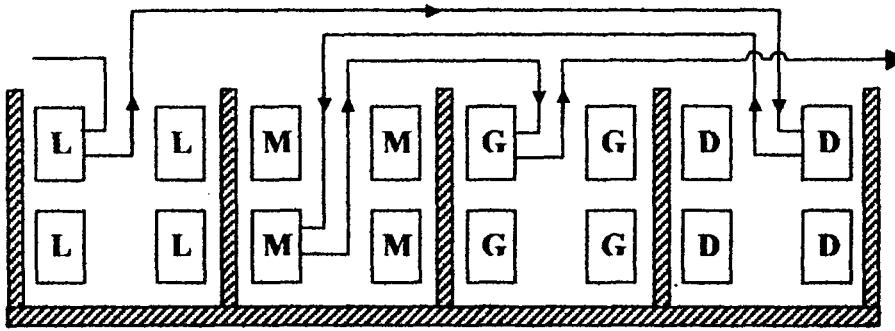
El inventario de bienes terminados existe porque la demanda de los consumidores en un determinado tiempo puede ser desconocida.

El plan de producción establece por completo el nivel de salida en términos generales (por ejemplo: familias de productos, horas estándar) . Tales planes incluyen una variedad de entradas, incluyendo planes financieros, demanda del consumidor, capacidades de ingeniería, disponibilidad de trabajo, fluctuaciones de inventario, y otras consideraciones. El plan de producción establece los límites superior e inferior en el plan maestro de producción.

El poder calendarizar las actividades de la empresa es un factor decisivo en su conducción, ya que permite aprovechar con mayor eficiencia los recursos disponibles. La calendarización es hecha con frecuencia interactiva con un sistema de soporte de decisión que es instalado en una computadora personal o estación de trabajo y conectado a un sistema ERP.

En manufactura, la función de calendarizar debe interactuar con otras funciones de decisión dentro de la planta. Un sistema que es ampliamente utilizado, es el sistema de planeación de requerimientos de materiales (MRP). Después de que un plan ha sido generado, es necesario que toda la materia prima y los recursos estén disponibles en los tiempos especificados. El MRP es un conjunto de técnicas que utiliza datos de la lista de materiales, de inventarios y del plan maestro de producción para calcular los requerimientos de materiales, liberar órdenes de compra, y colocar órdenes abiertas desfasadas en el tiempo, ajustando en el inventario las órdenes programadas. Las fechas en que todos los trabajos estarán terminados tienen que ser determinadas por el plan de producción y sistema de calendarización, y por el sistema MRP.

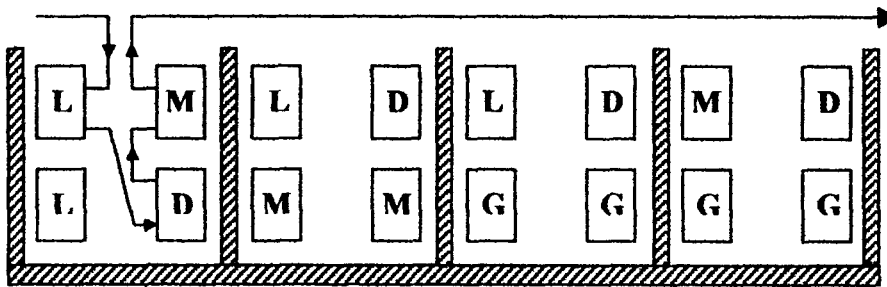
Además de todas las funciones en una empresa, también es importante el sistema de producción utilizado. Existen diferentes configuraciones en las cuales se pueden colocar las máquinas dependiendo del tipo de producción. Los tres arreglos básicos son: Arreglo Funcional, Arreglo Celular y Arreglo Lineal, que se muestran en la Figura 1.2.



Arreglo Funcional

Los equipos (L,M,G,D) son agrupados.

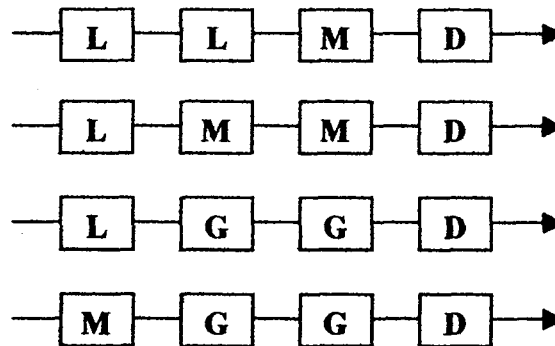
El flujo es extremadamente variado para cada producto.



Arreglo Celular

Una célula para cada familia de producto.

El flujo es regular para cada familia de producto, pero varía para cada producto dentro de la familia.



Arreglo Lineal

Una línea para cada producto o familia de productos.

El flujo es regular.

Figura 1.2 Los tres arreglos básicos [1]

Para fabricar un producto es posible utilizar cualquiera de los arreglos; sin embargo, cada una de las configuraciones ofrece diferentes ventajas y desventajas dependiendo del tipo de producto. El utilizar un arreglo equivocado puede traer como consecuencia no poder competir contra empresas del mismo ramo que tienen un arreglo apropiado.

Cada arreglo presenta diferentes problemas de cómo ordenar las actividades del taller para optimizar el uso de los recursos, y a éste se le denomina un problema de calendarización.

La calendarización consiste de planificar y priorizar actividades a desarrollarse en una secuencia ordenada, es una herramienta para optimizar el uso de los recursos disponibles.

Durante muchos años ha existido la inquietud de calendarizar las actividades de la empresa, y han aparecido innumerables métodos basados en distintos fundamentos como se puede apreciar en la Tabla 1.1.

En la gran mayoría de los sistemas ERP no se incluyen las tareas de calendarización en sus módulos de manufactura, y si las incluyen los módulos no están lo suficientemente desarrollados o son muy caros, dificultando su aplicación en situaciones reales dentro de una empresa.

La calendarización conduce al incremento de la utilización, de la eficiencia y capacidad; reduciendo el tiempo requerido para completar los trabajos, disminuyendo el inventario en proceso y aumentando la utilización de las máquinas; y como consecuencia, generar ganancias para la organización.

Existe una clasificación [2] de niveles de calendarización que se muestra en la tabla 1.2

Modelos	Dinámico					
	Analítico (Prescriptivo)		Numérico (Descriptivo)			
		Optimización	Heurísticas	Determinísticas	Estocásticas	
Programación no lineal	Optimización Clásica	Newton Raphson	Newton Raphson Método Jacobiano			
		Restricciones de igualdad	Método de derivadas restringidas Método Lagrangeano	Método Lagrangeano Técnica de propósito	Técnicas de propósitos	
		Restricciones de desigualdad	Extensión del método Lagrangeano Condiciones de Kuhn-Tucker			
	Algoritmos no lineales irrestrictos	Técnicas del propósito	Modelo de regresión Modelo de promedio móvil			
		Método de búsqueda directa	Búsqueda dicotómica			
		Método del gradiente	Método de la cuesta de mayor pendiente Método simplex regular			
	Algoritmos lineales no restringidos	Programación separable	Método de bases restringida Método de programación entera mixta			
		Programación cuadrática	Programación cuadrática			
		Programación geométrica	Programación geométrica			
		Programación estocástica	Programación estocástica			Programación estocástica
Programación lineal	Programación lineal real	Métodos de combinaciones lineales Algoritmo Simplex	Método de combinaciones lineales Método de gradientes más general Programación lineal óptimo continuo			
		Método simplex	Método plano de corte Enumeración implícita			
	Programación lineal entera	Programación lineal relajado	Método fraccional Algoritmo aditivo			
		Método de transporte Método de ramificación y acotación Método simplex Método de asignación	Método de calendarización	Método de calendarización	Método de calendarización	Método de calendarización
			Método de ramificación y acotación	Método de ramificación y acotación	Método de ramificación y acotación	Modelo de transporte
			Método de asignación			
			Modelo de árbol de extensión mínima	Modelos de balanceo de líneas Modelo de transbordo	Teoría de líneas de espera	Teoría de líneas de espera
		Modelo de redes	Modelo de transporte con una fuente y un destino	Modelo de transporte con una fuente y un destino		
			Modelo de flujo máximo	Modelo de flujo máximo		
			Modelo de la red capacitada de costo mínimo	Método simplex de la red capacitada		Conexidad estocástica
Modelo de la ruta más corta						
Modelo de redes	Modelo de asignación					
	Enumeración implícita					
	Principio de optimización	Principio de optimización		Construcción progresiva		
	Ecución recursiva de retroceso	Procedimiento de retroceso	Modelo Silver-Meal	Problema de decisión de Martov		
Teoría de decisiones y juegos	Decisiones de riesgo	Algoritmo de programación de etapas múltiples	Procedimiento de avance	Método de simulación	Métodos de simulación Algoritmos genéticos	
	Árboles de decisión					Criterio del valor esperado Criterio del valor esperado varianza Criterio del nivel de aceptación Criterio del futuro más probable
						Árboles Árboles de decisión Criterio de Laplace Criterio mínimo Criterio de Savage Criterio de Hurwicz
						Solución óptima de juego de dos personas y suma cero Estrategias mixtas
	Decisión bajo incertidumbre				Solución gráfica de juegos de (ZIN) y (MAX)	
	Teoría de juego				Solución gráfica de juegos de (MIN) por programación lineal	
	Algoritmos basados en I.A.	Sistemas expertos Técnicas de inferencia				
	Módulo de inventarios	Distribución		Modelo heurístico o de Silver Meel	Método estático de un solo artículo (CPE)	Modelo de revisión continua Modelo de un solo período
					Tamaño de lote económico	Modelo de la demanda instantánea sin costo fijo
				Modelo de revisión continuo	Modelo de la demanda uniforme sin costo fijo	
				Modelo estático de múltiples artículos con limitaciones en el almacen	Modelo de la demanda instantánea y costo fijo	
				Modelos de programación de la producción en N períodos	Modelo de múltiples períodos	
				Modelo sin escasez	Modelo sin demora que permite satisfacer la demanda pendiente	
		Modelo con escasez	Modelo sin demora en entrega que no satisface pedidos pendientes			
			Principio de optimización	Modelo con demora positiva en la entrega que satisface pedidos pendientes		
				Modelo con demora positiva en la entrega que no satisface pedidos pendientes		

Tabla 1.1 Clasificación de las metodologías de calendarización

Tabla 1.2 Clasificación de los niveles de Calendarización

Nivel	Ejemplo del Problema	Horizonte
1. Planeación a largo plazo.	Expansión de la planta, arreglo de la planta, diseño de la planta.	2-5 años
2. Planeación a medio plazo.	Suavización de la producción. Logística.	1-2 años
3. Planeación a corto plazo.	Plan de requerimientos, órdenes a piso, establecimiento de fechas de entrega.	3-6 meses
4. Calendarización	Rutina de <i>Job Shop</i> ¹ , balance de la línea de ensamble, tamaño de lote de procesamiento.	2-6 semanas
5. Calendarización reactiva/control	Trabajos "calientes", averías de las máquinas, tardanza de materiales.	1-3 días

La mayoría de las investigaciones en el área de calendarización industrial requieren el soporte de herramientas matemáticas complejas, lo que hace de esta investigación algo muy difícil, entonces es necesario desarrollar soluciones que sean fáciles de comprender y además que proporcionen un gran potencial para ser utilizadas efectivamente.

En la práctica, la calendarización se genera con algoritmos matemáticos o con reglas basadas en la práctica. Los algoritmos de calendarización desarrollan planes de producción que tienden a optimizar un criterio medible tal como: minimizar la desviación de la fecha de vencimiento, minimizar la penalización por tardanza, o minimizar el máximo atraso. Las reglas basadas en la práctica tratan de encontrar una calendarización que sea factible bajo el ambiente de operación, principalmente trabajando con reglas de "si...entonces".

Los problemas de calendarización generalmente se denotan por N/M/A/B, indicando N trabajos, M máquinas, Tipo del flujo de trabajo A, y medida del funcionamiento B, que va a ser apropiadamente minimizada o maximizada.

¹ En este trabajo de Tesis se utilizan con frecuencia las palabras *Job Shop*, refiriéndose al sistema de producción tipo taller, debido a que su uso en inglés es muy generalizado para nombrar a este tipo de sistema.

Cuando los trabajos visitan las mismas máquinas en el mismo orden el número de posibles combinaciones para calendarizarlos es igual a $N!$ Sin embargo, si los trabajos pasan por diferentes máquinas en diferente secuencia, el número de posibles combinaciones se incrementa a $(N!)^M$. Des esta forma el número de secuencias se incrementa rápidamente, convirtiéndose en un problema de complejidad NP.

En teoría de complejidad, se hace una distinción entre problemas de optimización y problemas de decisión. Un problema de decisión se refiere a una respuesta de “sí o no”. Es claro que el problema de optimización y el problema de decisión están fuertemente conectados. Si existe un algoritmo de tiempo polinomial para el problema de optimización, entonces existe un algoritmo de tiempo polinomial al problema de decisión y viceversa.

Un concepto fundamental de teoría de complejidad es el concepto de reducción del problema. Muy frecuente ocurre que un problema combinatorio es un caso especial de otro problema, equivalente a otro problema, o más general que otro. En general, un algoritmo que trabaja bien para un problema combinatorio, trabaja también para otro con unas pequeñas modificaciones. Específicamente, se dice que un problema P se reduce a un problema P' , si para cada caso de P se puede construir un caso de P' . En teoría de complejidad, se utiliza una notación más estricta. Un problema P se reduce polinomialmente a P' si un algoritmo de tiempo polinomial para P' implica un algoritmo de tiempo polinomial para P .

Se tienen tres clases de problema:

- *Clase P*: Contiene todos los problemas de decisión para los cuales existe un algoritmo que conduce a una correcta respuesta “sí o no”.
- *Clase NP*: Contiene todos los problemas de decisión donde se puede verificar si la respuesta es correcta dada una pista apropiada (por ejemplo: para un problema de calendarización, una pista puede ser una secuencia o un plan). Verificar que una secuencia satisface alguna condición dada, lógicamente es más simple que encontrar una secuencia que satisfaga tal condición.

- *Clase NP-Duro*: Un problema P, tanto uno de decisión como uno de optimización, es llamado un problema NP-Duro si la clase completa de problemas NP se reduce polinomialmente a P.

Los problemas de calendarización tipo *Job Shop* son problemas de dificultad NP-Duro ya que es prácticamente imposible encontrar la mejor solución y no se tiene una sola pista de ésta.

Los problemas asociados con N trabajos y M máquinas se pueden dividir en dos categorías [3]: 1) "*Flow Shop*" y 2) *Job Shop*, y a su vez *Job Shop* se puede dividir en dos sub-categorías 2a) Taller Abierto y 2b) Taller Dependiente.

En "*Flow Shop*" los trabajos son procesados en múltiples máquinas en una secuencia idéntica. Sin embargo, el tiempo de procesamiento de cada trabajo en cada máquina puede ser diferente. El objetivo puede ser el minimizar el tiempo de flujo promedio o el costo total.

Job Shop es uno de los sistemas de producción más utilizados. Hay diferentes máquinas en el taller, y un trabajo puede requerir algunas o todas las máquinas en alguna secuencia específica. Los principales objetivos pueden ser minimizar el tiempo de fabricación o reducir la penalización por tardanza, reducir el tiempo total de fabricación llamado "makespan" o $C_{máx}$, etc. En Taller Abierto no existe una secuencia operacional de un trabajo de la que dependa otro trabajo, mientras que en Taller Dependiente el orden de procesamiento de uno o más trabajos depende del procesamiento de otros trabajos.

Muchas organizaciones tienen una o más plantas con sistemas de producción tipo *Job Shop*. Los departamentos y equipos son arreglados de acuerdo a un *layout* funcional; es decir; los equipos y procesos que son del mismo tipo se localizan en el mismo departamento. Ya que la fabricación es de muchos productos diferentes en bajos volúmenes, el uso de equipos y herramientas especializados no es factible, por lo tanto es necesario utilizar equipo y herramientas de propósito general. Más aún, el material circula a través del piso por diferentes secuencias en cada trabajo. El inventario de trabajo en proceso es alto. Los tiempos de ejecución son largos por el manejo de los materiales que se requiere para transportarlos de departamento a departamento, y los trabajos deben esperar largos periodos hasta que se encuentren disponibles el equipo y los operadores.

Los sistemas de producción tipo *Job Shop* son diseñados para proporcionar altos niveles de flexibilidad e innovación, los cuales necesitan responder a las demandas de los consumidores para una amplia variedad de productos nuevos o existentes. El principal problema que se presenta en los sistemas tipo *Job Shop* es la calendarización de actividades.

Los *Job Shop* también se pueden clasificar según el proceso de llegada de trabajos:

- *Estáticos*: No quiere decir que los clientes hagan pedidos al mismo tiempo, sino que la programación se hace una vez por semana. Todos los trabajos están disponibles en un instante t (generalmente $t = 0$). Es el más frecuente y sólo se cambia la planificación por urgencias.
- *Dinámicos*: Se actualiza el programa cuando hay un trabajo. Los trabajos llegan intermitentemente.

También se pueden clasificar por la forma de los datos en:

- *Determinísticos*: los tiempos de llegada y los tiempos de procesamiento son conocidos y fijos. Se resuelven con algoritmos.
- *Estocásticos*: Son variables aleatorias. El taller se comporta como una cola.

Como se puede apreciar, la calendarización es un elemento muy importante para el éxito de un sistema de manufactura, desde la optimización del uso de recursos hasta la reducción del precio de los productos.

Por la gran variedad de sistemas de producción, no existe un método que logre una efectiva calendarización en todos ellos. Como consecuencia, es necesario desarrollar diversos algoritmos mediante distintas técnicas para satisfacer las exigencias de cada sistema.

El tipo de sistema de producción, el tipo de producto, el tipo de proceso, el volumen de producción, el tiempo de procesamiento del producto (en algunos casos), y el tiempo en la toma de decisiones son algunos factores que deben considerarse en la elección de un método de calendarización.

2. ESTADO DEL ARTE DE LOS MÉTODOS DE CALENDARIZACIÓN

A continuación se enumeran algunos de los métodos más comunes para obtener la solución del problema de calendarización de sistemas de producción tipo *Job Shop* con el fin de realizar un análisis comparativo de éstas, mencionando sus ventajas y desventajas para después plantear una técnica que también mejore la solución del problema.

Se inicia mencionando el algoritmo de Rama y Límite y el algoritmo Modificado de Retroceso-Avance que pertenecen a la calendarización de una sola máquina, ya que posteriormente son utilizados por otros métodos. Para los problemas tipo *Job Shop*, se presentan los algoritmos: Aproximación de Red, Cambio de Cuello de Botella, Codicioso Heurístico, Gliffer and Thompson, Generación de Planes sin Retraso y el Modificado, Algoritmos Genéticos combinados con Heurísticos y finalmente se presenta un método Estocástico de comparación de algoritmos.

2.1 CALENDARIZACIÓN DE UNA SOLA MÁQUINA

Cuando varios trabajos requieren de los servicios de una misma máquina es necesario saber a qué trabajo se le dará prioridad. El problema de calendarización de una sola máquina

consiste en determinar la secuencia de trabajos que mejor cumple con algún objetivo. En la práctica existen muchos objetivos y dependiendo de las condiciones existentes en un momento dado unos pueden tener más importancia que otros. Algunos de ellos son:

- Reducir penalización por entregas tardías o tempranas.
- Cumplir con las fechas de entrega.
- Minimizar la tardanza promedio.
- Minimizar la máxima tardanza.
- Minimizar el número de trabajos retardados.
- Minimizar el tiempo de flujo promedio de los trabajos.

Es posible desarrollar todas las posibles combinaciones de N trabajos y determinar cuál es la que mejor cumple con los objetivos, esta aproximación es llamada Enumeración Exhaustiva y permite analizar $N!$ combinaciones. Sin embargo cuando el número de trabajos se va incrementando considerablemente, generalmente cuando $N > 10$, la tarea se vuelve imposible. Se han desarrollado diversos métodos que permiten obtener una muy buena solución en un tiempo considerablemente reducido.

A continuación se presentan dos algoritmos reconocidos en la literatura para el problema de calendarización de una sola máquina, que posteriormente serán utilizados en métodos de calendarización de N trabajos y M máquinas.

2.1.1 ALGORITMO DE RAMA Y LÍMITE (BRANCH AND BOUND ALGORITHM)

Este algoritmo permite resolver cualquier problema de optimización con un número finito de posibles soluciones y es llamado Optimización Combinatoria. El objetivo es reducir la penalización por tardanza, y [3] resume este algoritmo como sería aplicado en un problema de secuenciación en 8 pasos que se definen a continuación.

Paso 1. Determinar el trabajo que será procesado al final. En cualquier secuencia siempre hay un trabajo que será procesado al final, y si no hay restricciones, cualquier trabajo puede serlo. Se

dibuja un diagrama de árbol creando un nodo para cada trabajo que pueda ser procesado en la última posición.

Paso 2. Calcular el límite inferior de la penalización total asociada con el nodo del paso 1. La penalización total se calcula de la siguiente forma: $(PT - D_i) * L_i$, donde D_i es la fecha de vencimiento del trabajo i , PT es la suma de los tiempos de procesamiento de los trabajos restantes incluyendo el trabajo en consideración, y L_i la penalización para el trabajo i .

Paso 3. Seleccionar el nodo (nodo padre) con el menor valor de penalización en el límite más bajo para que sea ramificado. Cada uno de los trabajos restantes puede ser procesado en la siguiente posición, así que se construyen nodos para cada uno de estos trabajos que se deriven del nodo padre.

Paso 4. Calcular la penalización por entrega tardía de los trabajos en los nodos creados en el paso 3. La penalización se calcula igual que en el paso 2 y es acumulativa, es decir, el valor del nodo presente incluye el valor obtenido en el nodo padre.

Paso 5. Repetir los pasos 3 y 4 hasta que la primera secuencia es determinada. Este valor de penalización es el valor mínimo inicial de penalización.

Paso 6. Buscar en el diagrama de árbol los nodos que tienen la mínima penalización mayor al valor mínimo actual de penalización, y eliminarlos.

Paso 7. Ramificar cada uno de los nodos restantes, eliminando los nodos que tienen la mínima penalización mayor al valor mínimo actual de penalización. Reemplazar el valor mínimo actual con un nuevo valor mínimo si uno es encontrado.

Paso 8. Una vez que todos los nodos restantes han sido eliminados, el valor mínimo actual será la penalización total para la secuencia de trabajos representada por la ramificación resultante.

Ventajas y desventajas: Este método tiene la ventaja de que es muy probable que la solución obtenida sea la óptima ya que realiza una enumeración de todas las posibilidades, sin embargo su

principal desventaja, es que está totalmente orientado a obtener soluciones con objetivo de reducción de costos y no se puede tomar otro criterio.

Ejemplo:

Se desea encontrar la mejor secuencia de fabricación para tres trabajos para realizarse en una sola máquina con los datos que se muestran a continuación.

Tabla 2.1 Ejemplo para Método del Algoritmo Rama y Límite.

Trabajo	Tiempo de Procesamiento p_i	Fecha de Vencimiento d_i	Penalización L_i
1	16	23	3
2	43	51	5
3	20	25	2

En la Figura 2.1 se muestran los resultados del procedimiento. El primer paso consistió en crear los nodos 1, 2 y 3 correspondientes a los trabajos 1, 2 y 3, y se realiza en cálculo de los costos de cada trabajo si fuera el último en realizarse:

$$\text{Nodo 1: } [d_1 - (p_1 + p_2 + p_3)] * L_1 = [23 - (16 + 43 + 20)] * 3 = 168$$

$$\text{Nodo 2: } [d_2 - (p_1 + p_2 + p_3)] * L_2 = [51 - (16 + 43 + 20)] * 5 = 140$$

$$\text{Nodo 3: } [d_3 - (p_1 + p_2 + p_3)] * L_3 = [25 - (16 + 43 + 20)] * 2 = 108$$

El nodo 3 resulta ser el de costo más bajo, así que se ramifico con los trabajos restantes (1 y 2). Se repite el mismo procedimiento, pero ahora se le suma el costo del nodo padre.

$$\text{Nodo 4: } [d_1 - (p_1 + p_2)] * L_1 = [23 - (16 + 43)] * 3 = 216$$

$$\text{Nodo 5: } [d_2 - (p_1 + p_2)] * L_2 = [51 - (16 + 43)] * 5 = 148$$

Ahora el nodo de más bajo costo es el Nodo 5, así que se ramifica con la operación sobrante (1). Ya que la fecha de vencimiento del trabajo 1 es mayor a su tiempo de procesamiento entonces no hay costo y se conserva el de su nodo padre.

El costo obtenido en el Nodo 6 ($W \geq 148$) es comparado con los Nodos 1 y 2. El Nodo 1 tiene un costo mayor a 148, así que es eliminado, sin embargo el Nodo 2 resulta ser menor, así que se ramifica (Nodos 7 y 8). Se realiza el mismo cálculo:

$$\text{Nodo 7: } [d_1 - (p_1 + p_3)] * L_1 = [23 - (16 + 20)] * 3 = 39 \text{ (más 140 del nodo padre) } = 179$$

$$\text{Nodo 8: } [d_3 - (p_1 + p_3)] * L_3 = [25 - (16 + 20)] * 2 = 22 \text{ (más 140 del nodo padre) } = 162$$

Los costos obtenidos son mayores a 148, así que los nodos son eliminados. La mejor secuencia de fabricación es {1,2,3}.

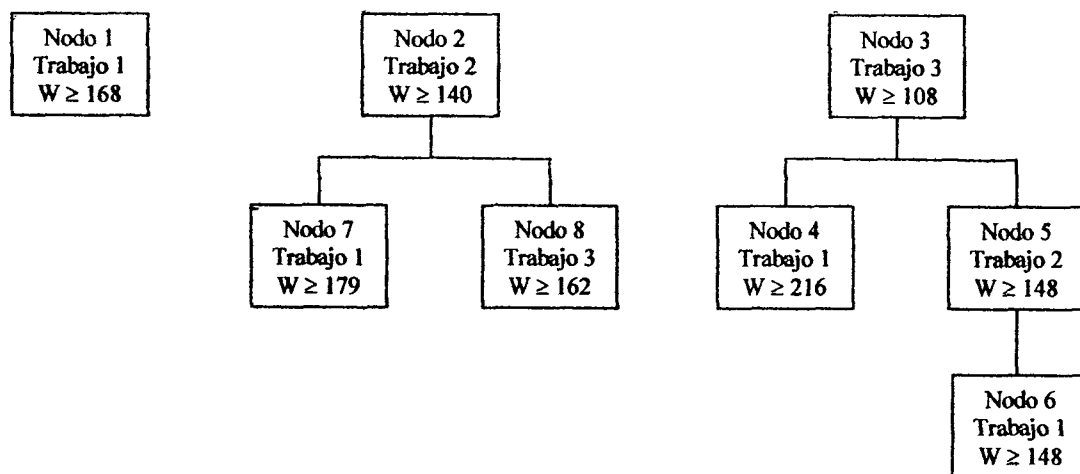


Fig. 2.1 Diagrama de Nodos del Algoritmo Rama y Límite.

2.1.2 ALGORITMO MODIFICADO DE RETROCESO-AVANCE (BACKWARD-FORWARD).

Cuando se trata de comparar entre varios métodos, generalmente se hace la suposición de que todos los trabajos se encuentran disponibles desde el tiempo cero, sin embargo no siempre se presenta este caso, por lo que es preciso desarrollar algoritmos que permitan la calendarización de

trabajos con diferentes tiempos de llegada. Este es el caso del algoritmo Modificado de Retroceso-Avance.

El algoritmo se encuentra dividido en dos partes. La primera es el Retroceso, que debe su nombre a que en esta fase se asignan los trabajos de la última posición a la primera, pero al modificar el método la asignación se realiza solo entre los trabajos que están compitiendo como se verá en esta misma sección. La segunda fase es la de Avance, donde se pretende mejorar la secuencia obtenida en la primera fase intercambiando de posición algunos de los trabajos.

Al igual que el algoritmo de Rama y Límite, aquí se presentan los pasos de cada fase como los presenta [3].

Fase de Retroceso

Paso 1. Declarar $C = 0$ y $k = 1$. C marcará el tiempo de finalización de los trabajos ya calendarizados, mientras que k será un auxiliar para indicar la posición en que cada trabajo será calendarizado.

Paso 2. Entre todos los trabajos no calendarizados, se selecciona el menor tiempo de llegada entre ellos y declararlo como ES.

Paso 3. Comparar los valores de ES y C . El menor asignarlo a S .

Paso 4. Seleccionar todos los trabajos cuyo tiempo de llegada es menor o igual a S . Guardar estos trabajos en el conjunto J .

Paso 5. Calcular T como la suma de los tiempos de procesamiento de todos los trabajos en el conjunto J . Calcular $H = T + S$ y asignar $C = H$.

Paso 6. Calendarizar todos los trabajos en el conjunto J en dirección de retroceso basado en penalización. La penalización TD_i del trabajo i se calcula de la siguiente forma: si $(H - D_i) > 0$, $TD_i = (H - D_i) * L_i$; de otra forma $TD_i = (D_i - H) * E_i$.

El criterio para seleccionar un trabajo es el de menor penalización. En el caso de un empate, el trabajo con mayor tiempo de procesamiento es el elegido. Si aún existe empate, escoger el trabajo con el mayor valor de tiempo de llegada, y si aún hay empate, romperlo arbitrariamente. Si hay n trabajos en el conjunto J , se asignan los trabajos del conjunto J comenzando en la posición $n+k-1$ y finalizando en la posición k . Una vez que un trabajo ha sido asignado, el valor de H es reducido por su tiempo de procesamiento y las penalizaciones de los trabajos restantes en el conjunto J son recalculados. El proceso se repite hasta que todos los trabajos en J sean asignados.

Paso 7. Calcular $k = k + n$.

Paso 8. Si k es menor o igual al total de trabajos, ir al paso 2; de otra forma detener la fase de Retroceso e ir a la fase de Avance.

Fase de Avance

Una vez obtenida una secuencia en la fase de Retroceso, en la fase de Avance se analiza la secuencia de trabajos de la posición 1 a la posición N . Definamos a k como el intervalo entre dos trabajos en la secuencia que serán intercambiados (por ejemplo posiciones 1 y 3 tienen un intervalo $k = 2$). Los pasos de la fase de avance son:

Paso 1. Calcular $k = N - 1$. Donde N es el número de trabajos.

Paso 2. Calcular $j = k+1$

Paso 3. Determinar el ahorro (o costo) intercambiando dos trabajos en la “mejor secuencia” con un intervalo de k . El trabajo calendarizado en la posición j es intercambiado con el trabajo calendarizado en la posición $j - k$ (si $j - k$ es cero o negativo, ir al paso 6). Calcular la penalización después del intercambio y compararla con la penalización de la “mejor secuencia”.

Paso 4. Si hay ahorro o cero en el paso 3, ir al paso 5; de otra forma, hay costo asociado con el intercambio, y el intercambio debe ser rechazado. Incrementar el valor de j en 1. Si j es igual o menor que N , ir al paso 3. Si j es mayor que N , ir al paso 6.

Paso 5. Si la penalización total ha disminuido, el intercambio es aceptado. Entonces, la nueva secuencia es ahora la “mejor secuencia”. Ir al paso 1. Aún si el ahorro es cero, se acepta el intercambio y se va al paso 1, a menos que la secuencia de trabajos haya sido analizada en alguna iteración anterior de esta misma fase, no se realiza el intercambio. Incrementar el valor de j en 1. Si j es menor que N , ir al paso 3. Si $j = N$, ir al paso 6.

Paso 6. Decrementar el valor de k en 1. Si $k > 0$, ir al paso 2. Si $k = 0$, ir al paso 7.

Paso 7. La secuencia resultante es la mejor secuencia generada por este procedimiento.

La forma de calcular la penalización en la fase de avance es de la siguiente forma.

Considérese:

- A_i Tiempo de llegada del trabajo i .
- CT Tiempo total de finalización de todos los trabajos analizados hasta el momento en la fase de avance. Inicialmente $CT = 0$.
- TD_i Penalización para el trabajo i .

Si $A_i > CT$, entonces $CT = A_i + P_i$; de otra manera $CT = CT + P_i$.

Si $CT - D_i > 0$, entonces $TD_i = (CT - D_i) * L_i$; de otra manera $TD_i = (D_i - CT) * E_i$.

Ventajas y desventajas: La principal ventaja de éste método es que se puede obtener la solución de problemas cuanto el tiempo de llegada de los trabajos no es el mismo para todos. Otra ventaja es que se puede obtener la solución de problemas con mayor número de trabajos que el método de Rama y Límite, sin embargo ésta puede hallarse o no muy cercana a la óptima ya que una gran cantidad de combinaciones son descartadas.

Ejemplo

El siguiente ejemplo es extraído de la publicación [3] donde se ejemplifica la calendarización de trabajos con tiempos de llegada diferentes en una sola máquina.

Considere los datos mostrados en la Tabla 2.2, donde P_i , A_i , D_i , L_i y E_i representan el tiempo de procesamiento, el tiempo de llegada, la fecha de vencimiento, penalizaciones tardía y temprana de los trabajos, respectivamente.

Tabla 2.2 Datos para el problema de calendarización de trabajos con diferentes tiempos de llegada para una sola máquina.

Trabajo	P_i	A_i	D_i	L_i	E_i
1	5	2	9	2	1
2	7	5	15	2	1
3	8	28	42	1	0
4	3	10	15	1	0
5	2	10	17	2	1

Fase de Retroceso

$k = 1$ y $C = 0$.

$ES = \text{menor}(2,5,28,10,10) = 2$.

$S = \text{mayor}(ES = 2, C = 0) = 2$.

$J = \{1\}$.

$T = 5$; $H = T + S = 5 + 2 = 7$; $C = H = 7$.

$n = 1$ porque hay solo un trabajo en el conjunto J .

Asignar el trabajo 1 en la posición $n + k - 1$ ($1 + 1 - 1 = 1$).

$k = k + n = 1 + 1 = 2$.

La secuencia parcial es 1 _ _ _ _ _.

$ES = \text{menor}(5,28,10,10) = 5$.

$S = \text{mayor}(ES = 5, C = 7) = 7$.

$J = \{2\}$.

$T = 7$; $H = T + S = 7 + 7 = 14$; $C = H = 14$.

$n = 1$ porque hay solo un trabajo en el conjunto J .

Asignar el trabajo 2 en la posición $n + k - 1$ ($1 + 2 - 1 = 2$).

$k = k + n = 1 + 2 = 3$.

La secuencia parcial es 1,2 _ _ _ _.

$ES = \text{menor}(28,10,10) = 10$.

$S = \text{mayor}(ES = 10, C = 14) = 14$.

$J = \{4,5\}$.

$T = 5$; $H = T + S = 5 + 14 = 19$; $C = H = 19$.

Penalización para el trabajo 4 = $(H - D_i) = (19 - 15) * 1 = 4$.

Penalización para el trabajo 5 = $(H - D_i) = (19 - 17) * 2 = 4$.

Ya que hay un empate, se selecciona el trabajo con el mayor tiempo de procesamiento. El tiempo de procesamiento del trabajo 4 es mayor, así que se selecciona el trabajo 4.

$n = 2$ porque hay 2 trabajos en el conjunto J.

Asignar el trabajo 4 en la posición $n + k - 1$ ($2 + 3 - 1 = 4$).

La secuencia parcial queda ahora 1,2 _ 4 _.

$H = 19 - \text{Tiempo de procesamiento del trabajo 4} = 19 - 3 = 16$

Solo queda un trabajo en el conjunto J (trabajo 5), se asigna a la posición k ($k = 3$).

La secuencia parcial es ahora 1,2,5,4 _.

$k = k + n = 3 + 2 = 5$.

$ES = 28$.

$S = \text{mayor}(28, 19) = 28$.

$J = \{3\}$.

$T = 8; H = T + S = 8 + 28 = 36; C = H = 36$.

Nótese que este trabajo puede ser asignado hasta el tiempo 28 y no al 19, que fue el tiempo de finalización del trabajo anterior.

$k = k + n = 5 + 1 = 6$

Ya que k excede el número total de trabajos (5), se detiene la fase de Retroceso. La secuencia obtenida hasta este punto es 1,2,5,4,3. Ahora se procede a la fase de Avance.

Fase de Avance

La penalización total para la secuencia 1,2,5,4,3 se calcula como sigue: (inicialmente $CT = 0$):

Trabajo 1: $CT = 7, TD_1 = 2$.

Trabajo 2: $CT = 14, TD_2 = 1$.

Trabajo 5: $CT = 16, TD_5 = 1$.

Trabajo 4: $CT = 19, TD_4 = 4$.

Trabajo 3: $CT = 36$, $TD_3 = 0$.

La penalización total para la secuencia 1,2,5,4,3 es la suma de todos los TD_i , que es 8.

Ciclo 1, Iteración 1. Mejor secuencia 1,2,5,4,3. $k = \text{Número de trabajos} - 1 = 5 - 1 = 4$

Intercambiar 1 y 3. Secuencia a examinar: 3,2,5,4,1.

Plan	28	(3/8)	36	(2/7)	43	(5/2)	45	(4/3)	48	(1/5)	53
D_i		(3)	42	(2)	15	(5)	17	(4)	15	(1)	9
Penalización	0+56+56+33+88=233										

Penalización total = $233 > 8$, por lo tanto no hay intercambio y se conserva la secuencia inicial.

Ciclo 1, Iteración 2. Mejor secuencia 1,2,5,4,3. $k = 3$. Intercambiar 1 y 4. Secuencia a examinar 4,2,5,1,3. Penalización total = $56 > 8$, no hay intercambio.

Ciclo 1, Iteración 3. Mejor secuencia 1,2,5,4,3. $k = 3$. Intercambiar 2 y 3. Secuencia a examinar 1,3,5,4,2. Penalización total = $136 > 8$, no hay intercambio.

Ciclo 1, Iteración 4. Mejor secuencia 1,2,5,4,3. $k = 2$. Intercambiar 1 y 5. Secuencia a examinar 5,2,1,4,3. Penalización total = $55 > 8$, no hay intercambio.

Ciclo 1, Iteración 5. Mejor secuencia 1,2,5,4,3. $k = 2$. Intercambiar 2 y 4. Secuencia a examinar 1,4,5,2,3. Penalización total = $18 > 8$, no hay intercambio.

Ciclo 1, Iteración 6. Mejor secuencia 1,2,5,4,3. $k = 2$. Intercambiar 5 y 3. Secuencia a examinar 1,2,3,4,5. Penalización total = $75 > 8$, no hay intercambio.

Ciclo 1, Iteración 7. Mejor secuencia 1,2,5,4,3. $k = 1$. Intercambiar 1 y 2. Secuencia a examinar 2,1,5,4,3. Penalización total = $30 > 8$, no hay intercambio.

Ciclo 1, Iteración 8. Mejor secuencia 1,2,5,4,3. $k = 1$. Intercambiar 2 y 5. Secuencia a examinar 1,5,2,4,3. Penalización total = $22 > 8$, no hay intercambio.

Ciclo 1, Iteración 9. Mejor secuencia 1,2,5,4,3. $k = 1$. Intercambiar 5 y 4. Secuencia a examinar 1,2,4,5,3. Penalización total = $9 > 8$, no hay intercambio.

Ciclo 1, Iteración 10. Mejor secuencia 1,2,5,4,3. $k = 1$. Intercambiar 4 y 3. Secuencia a examinar 1,2,5,3,4. Penalización total = $28 > 8$, no hay intercambio.

En este caso, el aplicar la fase de Avance no realizó ninguna mejora a la secuencia obtenida por la fase de Retroceso, por lo que la secuencia óptima obtenida por este método es 1,2,5,4,3 con una penalización total de 8.

2.2 CALENDARIZACIÓN DE N TRABAJOS Y M MÁQUINAS

Cuando N distintos trabajos tienen su propia secuencia de operaciones en M diferentes máquinas con tiempos de procesamiento variados, entonces nos encontramos ante un problema de calendarización del tipo *Job Shop*. El problema consiste en encontrar la mejor secuencia de operaciones en cada máquina para cumplir con un objetivo; por lo que se han desarrollado métodos basados en distintas técnicas. Aquí se presentan algunos métodos.

2.2.1 APROXIMACIÓN DE RED PARA LA CALENDARIZACIÓN DE *JOB SHOP*.

Una forma de representar las secuencias de actividades de N trabajos y M máquinas es creando una red. Al igual que en el Método de Ruta Crítica, existen dos elementos que son básicos para su formación: actividad y evento representados por una flecha y un nodo, respectivamente. Una actividad (flecha) es el tiempo y recursos requeridos para realizar una operación, mientras que un evento (nodo) representa el tiempo de inicio o finalización de una actividad. Ver Figura 2.2.

Para poder identificar las actividades críticas; es decir, aquellas que al retrasarse provocarían el atraso de todo el proyecto, se pueden desarrollar algunos cálculos básicos de tiempo en los nodos.

Tiempo más temprano pronosticado de finalización. Es el tiempo máximo requerido en la red para finalizar todos los trabajos. Se determina mediante el Método de la Ruta Crítica. Es el valor presentado sobre el último nodo. Ver Figura 2.2.

Tiempo de inicio más temprano. Es el tiempo más temprano posible para comenzar una actividad pero con la condición de que la operación inmediata anterior haya concluido. Se calculan comenzado del nodo Inicio y evitando que exista tiempo de ocio entre operaciones. Es el valor presentado sobre cada nodo en corchetes []. Ver Figura 2.2.

Tiempo de inicio más tardío. Representa lo más tarde posible que puede comenzar una operación para no retrasar el tiempo total de finalización. Para poder calcular estos tiempos, es necesario comenzar del último nodo (Fin) y procesar las operaciones en dirección al primer nodo. Es el valor presentado sobre cada nodo en llaves {}. Ver Figura 2.2.

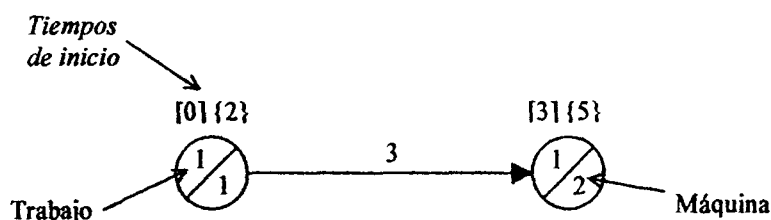


Fig. 2.2 Tiempos conocidos en el diagrama de red.

Por ejemplo, en la Tabla 2.3 se presentan los datos de 4 trabajos que requieren de los servicios de un total de 3 máquinas con sus tiempos de procesamiento respectivos.

Tabla 2.3 Datos para construcción de red.

Trabajos	Máquinas	Tiempos de Procesamiento
1	1,2	2,4
2	1,3	3,1
3	1,3,2	3,4,3
4	3,1,2	2,2,3

Tomando estos datos se puede construir el diagrama de red como se muestra en la Figura 2.3.

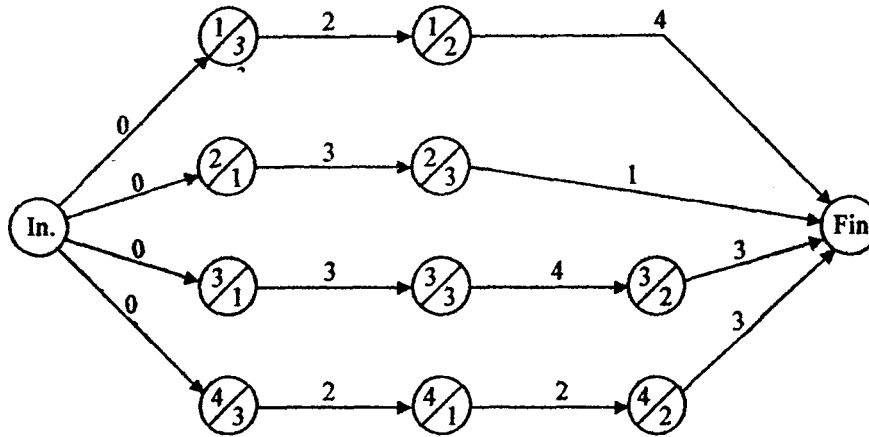


Fig. 2.3 Diagrama de red de un problema tipo *Job Shop*.

Las operaciones pertenecientes a diferentes trabajos pero que tienen que ser procesadas en la misma máquina pueden ser conectadas por líneas punteadas. Figura 2.4.

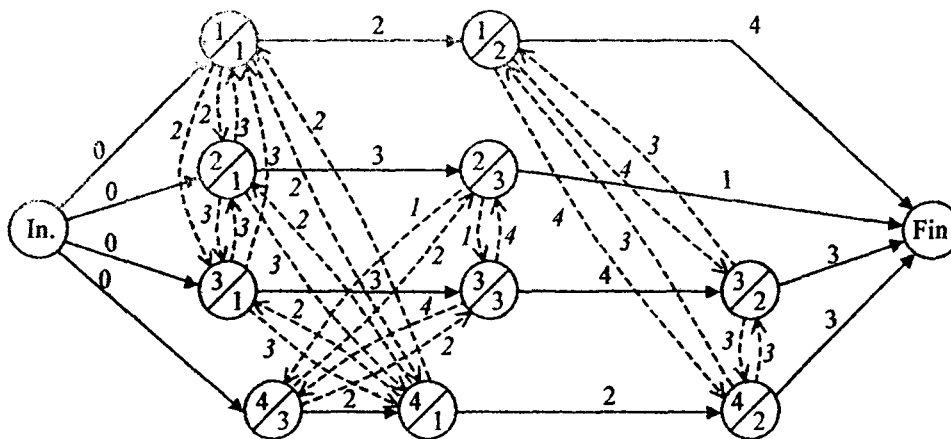


Fig. 2.4 Conexión entre operaciones de una misma máquina

Como cita [4], el problema de calendarización del *Job Shop* es encontrar el orden de las operaciones en cada máquina, es decir, seleccionar un arco entre todos los pares de direcciones opuestas de tal forma que la gráfica resultante sea acíclica (que no haya conflictos de precedencia entre operaciones) y que la longitud de la ruta más larga entre los nodos Inicio y Fin sea la mínima. La ruta máxima determina el tiempo total de fabricación.

Ventajas y desventajas: Este procedimiento permite observar de forma gráfica la secuencia de operaciones de cada máquina, además de que permite calcular más fácilmente los tiempos de

inicio y finalización de cada operación. Este procedimiento tiene como función principal la de servir de apoyo para otros métodos.

2.2.2 MÉTODO HEURÍSTICO CAMBIO DE CUELLO DE BOTELLA (SHIFTING BOTTLENECK HEURISTIC).

El Método Heurístico Cambio de Cuello de Botella desarrollado por [5] en 1988, según [4] es probablemente el procedimiento más poderoso conocido hasta ahora entre todos los heurísticos para el problema de calendarización del tipo *Job Shop*. En este procedimiento cada máquina aún sin calendarizar es considerada como un problema de calendarización de una sola máquina. La calidad de la solución obtenida depende fuertemente de las secuencias obtenidas en los problemas de una sola máquina. En el procedimiento original se utiliza el método de Rama y Límite, mientras que en el procedimiento modificado según [3] se hace uso del método de Retroceso-Avance. Además, se hace necesario el uso de diagrama de red para un manejo más cómodo de los datos a lo largo del procedimiento.

El procedimiento para el Método Modificado de cambio de Cuello de Botella se describe en los siguientes pasos:

Paso 1. Considérese a M como el conjunto de todas las máquinas a calendarizar, siendo M' todas las máquinas a las que se les ha asignado una secuencia. Inicialmente $M' = \{\emptyset\}$. Se realiza el cálculo de los tiempos de inicio (temprano y tardío) de todos los nodos.

Paso 2. Todas las máquinas que no han sido calendarizadas; es decir, las que se encuentran en M se resuelven de forma separada como problemas de una sola máquina. Todas las operaciones que serán realizadas en una máquina se toman como trabajos de ésta, que tienen diferentes tiempos de llegada ya que es necesario que finalizan sus operaciones inmediatas anteriores. Se hace uso del método de Retroceso-Avance Modificado. Se calcula la tardanza para cada máquina, que es equivalente a calcular la penalización total con una penalización tardía de 1 y una penalización temprana de 0 en cada trabajo.

Paso 3. La máquina que resulte con la mayor tardanza se selecciona como el cuello de botella, y será llamada la máquina k . Ahora el diagrama de red se modifica agregando conexiones sólidas entre nodos (operaciones) que están utilizando la máquina k según la secuencia obtenida en el paso 2 para la misma máquina.

Paso 4. Recalcular los tiempos de inicio en el diagrama de red considerando las nuevas conexiones agregadas en el paso 3. Se debe tener cuidado de que los nuevos tiempos tengan una secuencia coherente.

Paso 5. Examinar todas las máquinas que se encuentran incluidas en M' . Determinar si las conexiones existentes continúan siendo óptimas, es posible que las conexiones creadas por la máquina k hayan cambiado esta condición.

Seleccionando una máquina a la vez, por ejemplo la máquina j ; se eliminan las conexiones creadas anteriormente por j , y se busca la menor tardanza resolviendo el problema de una sola máquina, obteniendo los datos del diagrama de red modificado por k . Una vez obtenida la mejor secuencia para la máquina j , se crean las conexiones sólidas en la red. Se calcula el tiempo total de fabricación de todo el proyecto con la nueva secuencia de j . Si el nuevo resultado es menor o igual al anterior la nueva secuencia de j es conservada; pero si este no es el caso, entonces se conserva la secuencia original de j .

Paso 6. Agregar la máquina k al conjunto M' . Si M' es igual a M , entonces termina el procedimiento; de otra manera ir al paso 2.

Ventajas y desventajas: La principal ventaja de este algoritmo es que trata de calendarizar las máquinas con mayor conflicto que en este caso son las cuello de botella, pero el problema es que se corre el riesgo de eliminar algún resultado cercano al óptimo o al óptimo.

Ejemplo:

El siguiente ejemplo es extraído de la publicación de [3]. Considere los datos del siguiente problema. Tabla 2.4.

Tabla 2.4 Datos para ejemplo del Método Cambio de Cuello de Botella.

Trabajos	Máquinas	Tiempos de Procesamiento
1	1,3	1,2
2	2,3	3,1
3	3,1,2	2,1,1
4	1,2,3	3,2,1
5	3,1	2,1

Porque sólo se desea determinar las máquinas cuello de botella, las penalizaciones temprana y tardía de todos trabajos son 0 y 1, respectivamente. Los datos son representados en la Figura 2.5, y se aplica el procedimiento de la Ruta Crítica al diagrama de red para determinar el tiempo total de fabricación inicial $C_{máx}$.

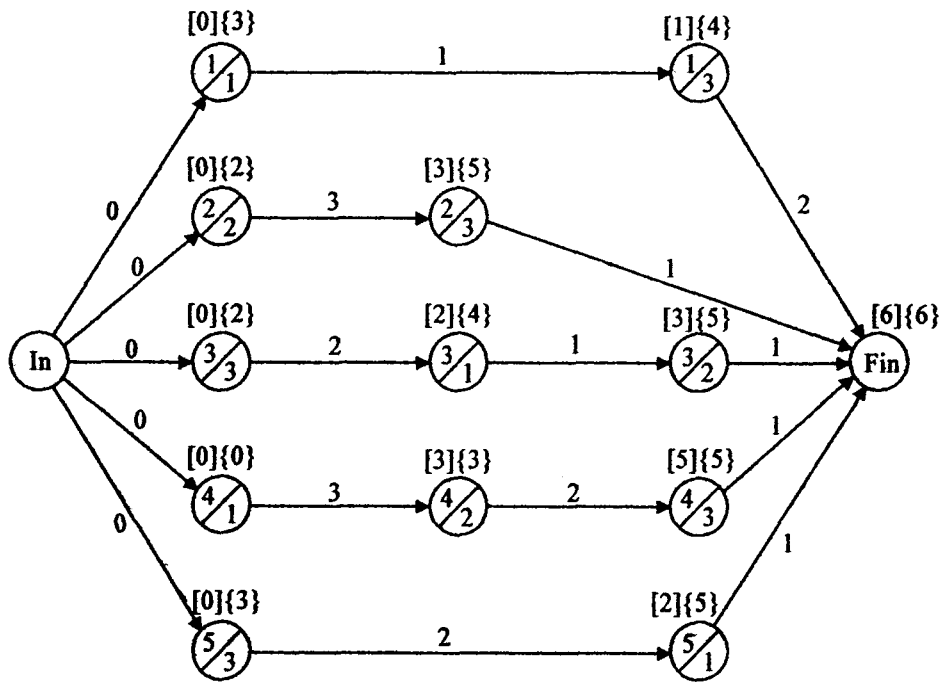


Fig. 2.5 Diagrama de red de un problema tipo Job Shop.

Por ejemplo, el nodo 5/3 nos indica que se trata del trabajo número 5 cuya operación se realiza en la máquina número 3. Su tiempo de procesamiento es de 2, y su tiempo más temprano de procesamiento es de 0, es decir, que se encuentra disponible desde el tiempo 0; mientras que el tiempo más tardío de procesamiento es 3, para evitar que este trabajo retrase el $C_{máx}$. En el caso

de que los tiempos de inicio temprano y tardío sean iguales, significa que no puede existir ningún retraso en la operación, porque atrasaría todo el proyecto.

Las fechas de vencimiento de cada operación equivalen al tiempo más tardío en que puede iniciar la siguiente operación del mismo trabajo.

Iteración 1. Determinar la tardanza mínima de todas las máquinas incluidas en el conjunto M. En el primer ciclo el conjunto M contiene todas las máquinas y M' está vacío. Ya que la penalización por tardanza es de 1 para todos los trabajos, la minimización de la penalización por tardanza es equivalente a la minimización de la tardanza. Entonces se aplica el procedimiento de Avance-Retroceso. Los datos se obtienen de la Figura 2.5.

Máquina 1, Iteración 1. La secuencia óptima es 4,1,3,5; la penalización es de 0 ya que no rebasa el $C_{máx}$.

	Trabajo 1	Trabajo 3	Trabajo 4	Trabajo 5
P_i	1	1	3	1
A_i	0	2	0	2
D_i	4	5	3	6
L_i	1	1	1	1
E_i	0	0	0	0

Los datos se obtienen de la siguiente forma: por ejemplo, para el trabajo 3, se está considerando el nodo 3/1. La flecha saliente de este nodo, representa P_i ; A_i es el tiempo de inicio más temprano del nodo 3/1; D_i es el tiempo de inicio más tardío del nodo 3/2, y solo hay penalización por tardanza que es de 1 para todos los trabajos.

Máquina 2, Iteración 1. La secuencia óptima es 2,4,3; la penalización es de 0.

	Trabajo 2	Trabajo 3	Trabajo 4
P_i	3	1	2
A_i	0	3	3
D_i	5	6	5
L_i	1	1	1
E_i	0	0	0

Máquina 3, Iteración 1. La secuencia óptima es 5,3,2,4,1; la penalización es de 2.

	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4	Trabajo 5
P_i	2	1	2	1	2
A_i	1	3	0	5	0
D_i	6	6	4	6	5
L_i	1	1	1	1	1
E_i	0	0	0	0	0

Se selecciona la máquina que tiene la mayor penalización como cuello de botella. En este caso la máquina 3 tiene la mayor penalización, así que se selecciona. Utilizando la secuencia óptima de la máquina cuello de botella como guía para el procedimiento, se conectan con arcos las operaciones a desarrollarse en la máquina 3. Se anotan los tiempos de procesamiento de cada operación en su arco correspondiente, como se observa en la Figura 2.6. Agregar la máquina 3 al conjunto M' ; ahora $M'=\{3\}$. Recalcular los tiempos más tempranos y tardíos para la red completa utilizando la Figura 2.6. El nuevo $C_{máx}$ es 8.

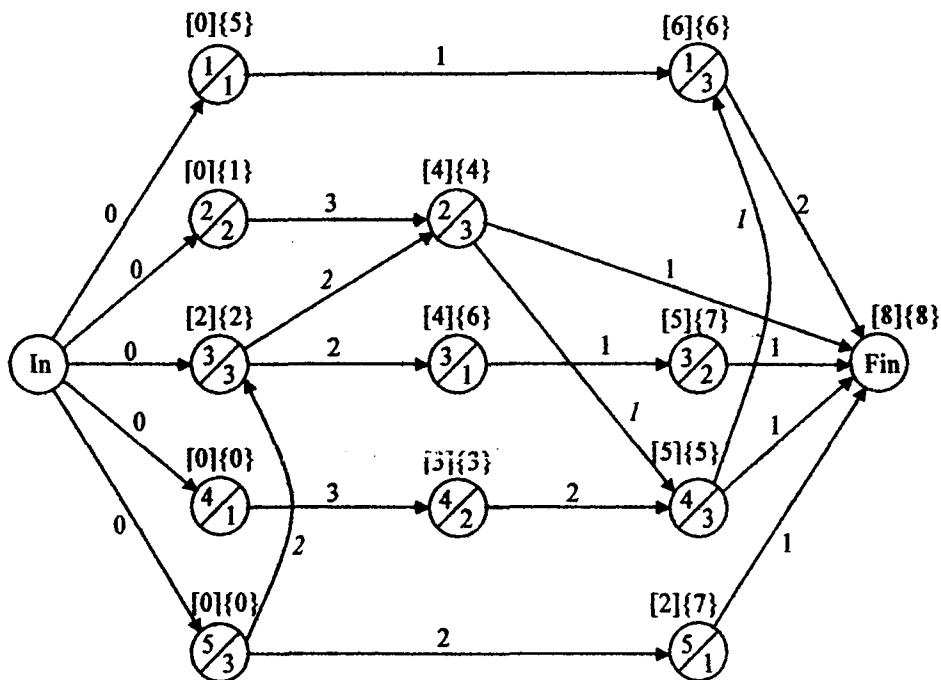


Fig. 2.6 Iteración 1, solución de la máquina 3.

Iteración 2. Ya que $M-M'=\{1,2\}$, se desarrolla la secuencia óptima para las máquinas 1 y 2 que podrían minimizar la penalización por tardanza. Los valores correspondientes a P_i , A_i , y D_i se toman de la Figura 2.6.

Máquina 1, Iteración 2. La secuencia óptima es 4,1,3,5; la penalización es de 0.

	Trabajo 1	Trabajo 3	Trabajo 4	Trabajo 5
P_i	1	1	3	1
A_i	0	4	0	2
D_i	6	7	3	8
L_i	1	1	1	1
E_i	0	0	0	0

Máquina 2, Iteración 2. La secuencia óptima es 2,4,3; la penalización es de 0.

	Trabajo 2	Trabajo 3	Trabajo 4
P_i	3	1	2
A_i	0	5	3
D_i	4	8	5
L_i	1	1	1
E_i	0	0	0

Es posible seleccionar las máquinas 1 o 2 como el cuello de botella porque tienen la misma penalización. Al azar se selecciona la máquina 2. Utilizando la secuencia óptima de la máquina 2 se modifica el diagrama de red como se muestra en la Figura 2.7, conectando la secuencia de operaciones asociadas con la máquina 2. Recalcular los tiempos de inicio temprano y tardío y tardío. C_{\max} permanece en 8.

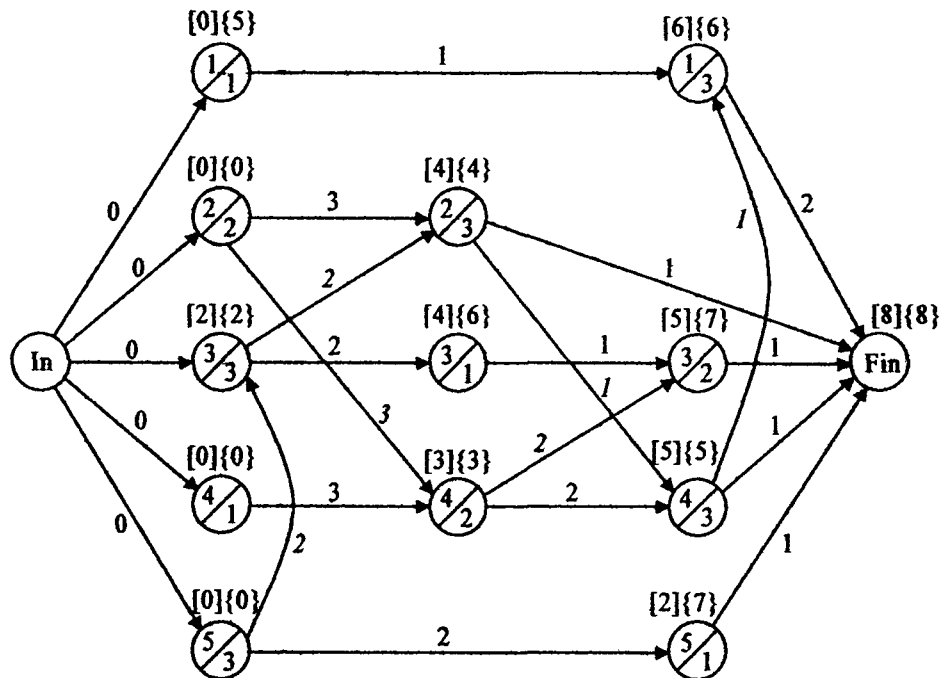


Fig. 2.7 Iteración 2, solución de las máquinas 2 y 3.

Aplicando el paso 5, $M'=\{3\}$. La secuencia óptima para la máquina 3 en la última iteración fue 5,3,2,4,1; como se ve en la Figura 2.7. La red se modifica retirando las conexiones de la máquina 3 como se ve en la Figura 2.8. Recalcular los tiempos de inicio temprano y tardío de la Figura 2.7. Los datos de la solución son utilizados como datos de entrada para la reevaluación de la máquina 3.

Máquina 3, Iteración 2. La secuencia óptima es 5,3,2,4,1; la penalización es de 2.

	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4	Trabajo 5
P_i	2	1	2	1	2
A_i	1	3	0	5	0
D_i	6	6	4	6	5
L_i	1	1	1	1	1
E_i	0	0	0	0	0

Ya que se obtuvo la misma secuencia que antes, se unen las operaciones de la máquina 3 de nuevo, y la red permanece igual que la Figura 2.7. No hay mejora porque C_{\max} continúa siendo 8. Agregar la máquina 2 a M' , así que ahora $M'=\{2,3\}$.

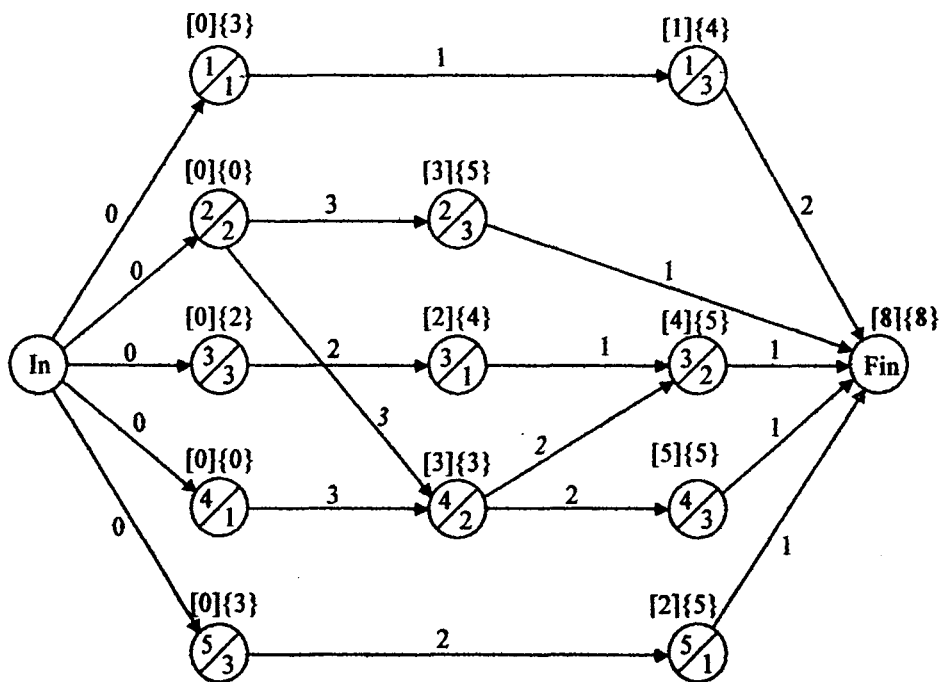


Fig. 2.8 Iteración 2, Máquina 2 resuelta, máquina 3 retirada.

Iteración 3. La máquina 1 es la única que sobra en $M-M'$, así que se resuelve buscando la menor penalización. Los datos de la máquina 1 se toman de la Figura 2.7.

Máquina 1, Iteración 3. La secuencia óptima es 4,1,3,5; la penalización es de 0.

	Trabajo 1	Trabajo 3	Trabajo 4	Trabajo 5
P_i	1	1	3	1
A_i	0	4	0	2
D_i	6	7	3	8
L_i	1	1	1	1
E_i	0	0	0	0

Modificar el diagrama de red como se muestra en la Figura 2.9. agregando líneas sólidas entre las operaciones de la máquina 1. Recalcular los tiempos de inicio temprano y tardío nos da como resultado $C_{\max} = 8$. En este punto, ya que $M' = \{2,3\}$, obtenemos nuevamente las secuencias para las máquinas 2 y 3 buscando menor penalización total y ver si hay alguna mejora. Comenzamos removiendo las líneas entre operaciones de la máquina 3, como se ve en la Figura 2.10., y determinar la menor penalización para la máquina 3 de la condición resultante.

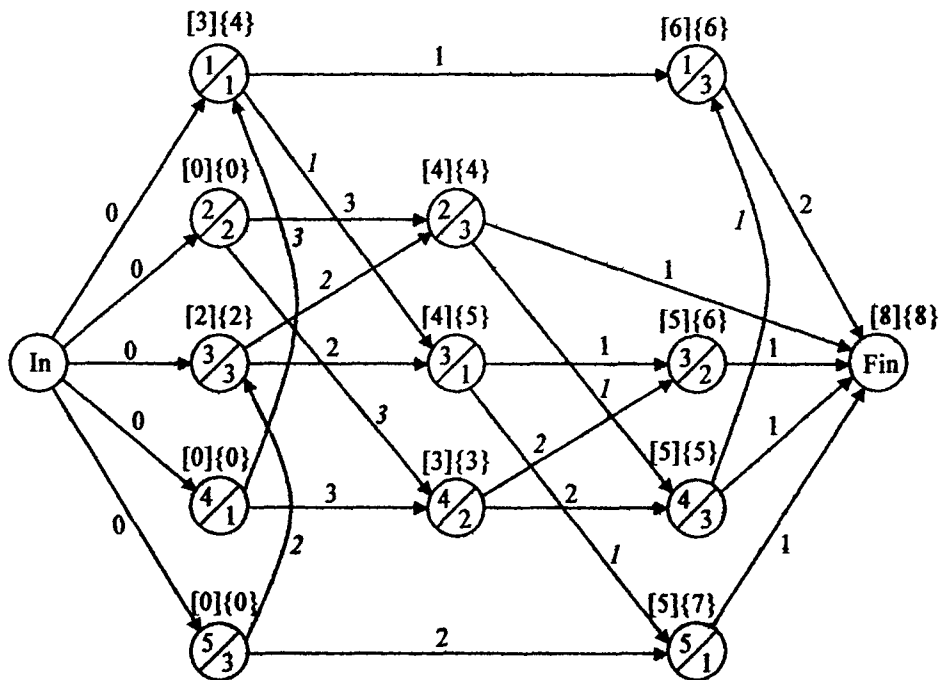


Fig. 2.9 Iteración 3, solución de las máquinas 1, 2 y 3.

Máquina 3, Iteración 3. La secuencia óptima es 5,3,2,4,1; la penalización es de 2.

	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4	Trabajo 5
P_i	2	1	2	1	2
A_i	4	3	0	5	0
D_i	6	6	4	6	5
L_i	1	1	1	1	1
E_i	0	0	0	0	0

Ya que la secuencia de trabajos no cambia, el $C_{máx}$ permanece en 8. Regresando a la red de la Figura 2.9, se retiran las líneas entre operaciones de la máquina 2, como se muestra en la Figura 2.11. Recalcular los tiempos de inicio temprano y tardío. Se busca una solución para la máquina 2 tratando de minimizar la penalización utilizando los datos de la Figura 2.11.

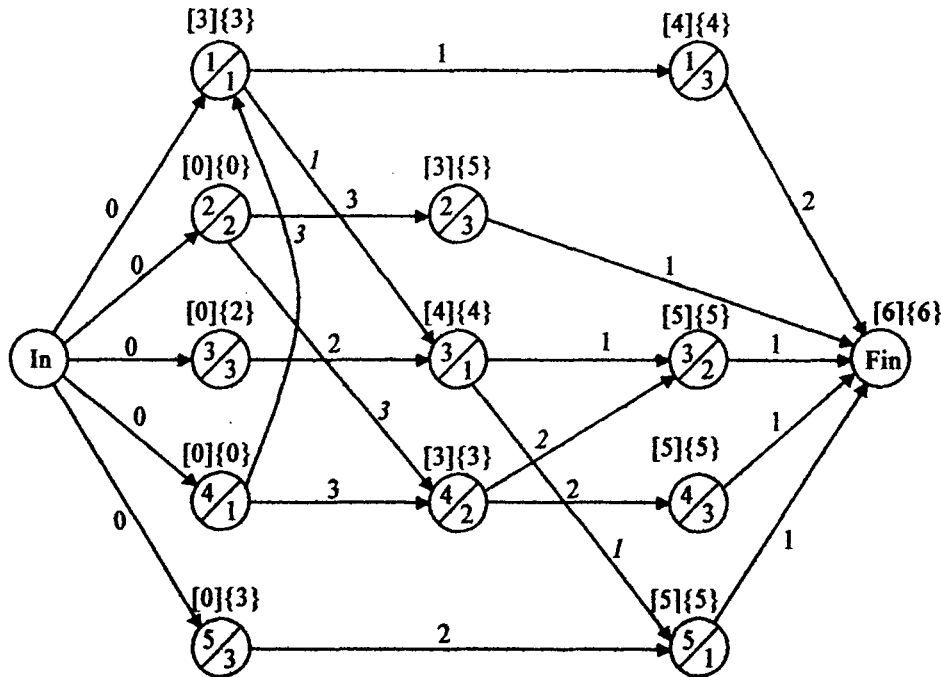


Fig. 2.10 Iteración 3, solución de las máquinas 1 y 2, máquina 3 retirada.

Máquina 2, Iteración 3. La secuencia óptima es 2,4,3; la penalización es de 0.

	Trabajo 2	Trabajo 3	Trabajo 4
P_i	3	1	2
A_i	0	5	3
D_i	4	8	5
L_i	1	1	1
E_i	0	0	0

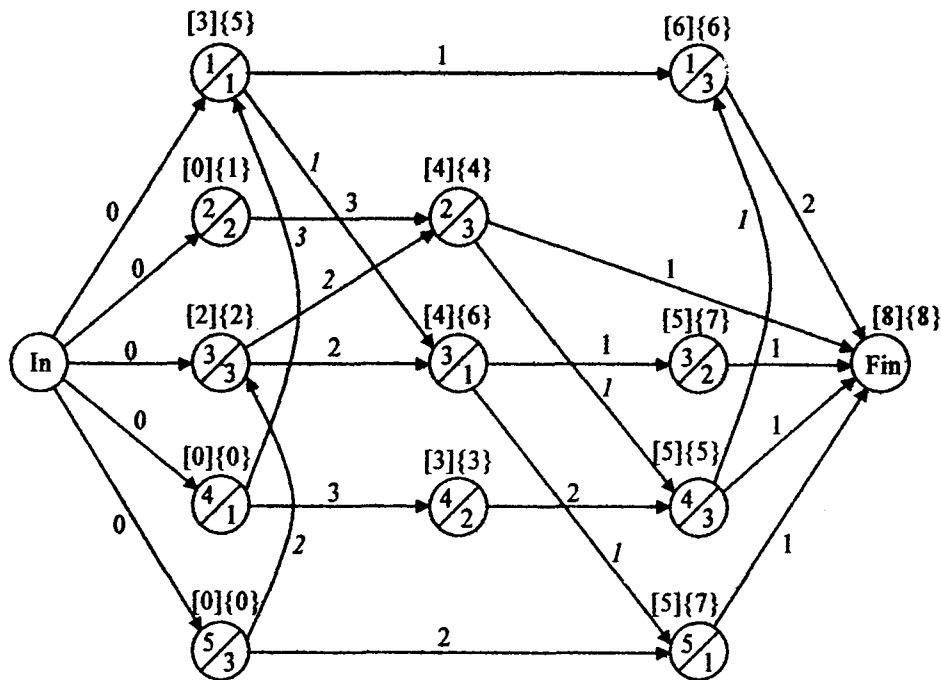


Fig. 2.11 Iteración 3, solución de las máquinas 1 y 3, máquina 2 retirada.

Una vez más la secuencia no cambia y el $C_{máx}$ permanece en 8. Regresamos al diagrama de red como en la Figura 2.9. Todas las máquinas están en el conjunto M' , así que se ha alcanzado la secuencia óptima para cada máquina en la Figura 2.9. el óptimo $C_{máx}$ es 8, que es el tiempo de finalización de la red. La secuencia en cada máquina se obtiene siguiendo la secuencia de trabajos conectados para cada máquina. Para la máquina 1 la secuencia es 4,1,3,5; para la máquina 2 es 2,4,3; y para la máquina 3 es 5,3,2,4,1. Así que la carga correspondiente a cada máquina es:

Máquina 1 0 (4/3) 3 (1/1) 4 (3/1) 5 (5/1) 6

Máquina 2 0 (2/3) 3 (4/2) 5 (3/1) 6

Máquina 3 0 (5/2) 2 (3/2) 4 (2/1) 5 (4/1) 6 (1/2) 8

2.2.3 ALGORITMO CODICIOSO HEURÍSTICO

En los últimos años, se han desarrollado diferentes algoritmos heurísticos para la calendarización de actividades, como el propuesto por [6], para resolver $N/M/G/C_{\text{máx}}$ cuyo método al que le denomina “codicioso heurístico”. Trata de resolver un problema de N trabajos y M máquinas como uno de dos trabajos, es decir, transformarlo en un problema $2/M/G/C_{\text{máx}}$. Los trabajos se consideran uno después del otro de acuerdo con una secuencia fija. El algoritmo propuesto es el siguiente:

1. Seleccionar una secuencia de trabajos al azar, considérese la secuencia J_1, J_2, \dots, J_N .
2. Asignar las máquinas al primer trabajo J_1 de acuerdo con algunas reglas de prioridad.
3. Hacer $J_{\text{com}} = J_1$.
4. Comienza un ciclo: Para $i = 2$ hasta N
 - a. Resolver como problema de 2 trabajos con J_{com} y J_i .
 - b. Construir un trabajo combinado J' de los trabajos J_{com} y J_i .
 - c. Hacer $J_{\text{com}} = J'$.
5. La secuencia final y $C_{\text{máx}}$ puede ser determinada de la combinación de trabajos J_{com} .

Ventajas y desventajas: La ventaja de este método es que trata de simplificar el modo de obtener la solución de problemas tipo *Job Shop*; sin embargo, se puede apreciar que este algoritmo depende de la secuencia de trabajos escogida en el primer paso, además de que no asegura obtener una solución aceptable.

2.2.4 ALGORITMO DE GIFFLER AND THOMPSON

El Algoritmo de Giffler and Thompson puede ser considerado como una base común de los algoritmos heurísticos basados en reglas de despacho. Sea $Q(t)$ el conjunto de todas las operaciones no calendarizadas en el tiempo t ; r_i y c_i denoten el tiempo más temprano de inicio de la operación y el tiempo más temprano de finalización, respectivamente, de la operación i . El algoritmo asigna operaciones disponibles a las máquinas, es decir, operaciones que pueden comenzar a ser procesadas. Los conflictos son resueltos al azar.

$t = 0; Q(t) = \{1, \dots, n-1\};$

Repetir

Entre todas las operaciones no calendarizadas en $Q(t)$, hacer j^* la operación con el menor tiempo de finalización, es decir, $c_{j^*} = \min \{c_j | j \in Q(t)\}$.

Hacer m^* la máquina en que tiene que ser procesada la operación j^* .

Escoger al azar una operación i si hay conflicto $\{j \in Q(t) | j \text{ tiene que ser procesada en la máquina } m^* \text{ y } r_j < c_{j^*}\}$.

$Q(t) = Q(t) \setminus \{i\}$; Modificar c_j para todas las operaciones $j \in Q(t)$. Ajustar t hasta la próxima asignación posible de operación a una máquina.

Hasta que $Q(t)$ esté vacío.

Ventajas y desventajas: El uso de reglas de despacho favorece en la elección de alguna operación cuando se encuentra en conflicto con otra basándose en cierta prioridad. La principal desventaja de este método es que al elegir al azar una operación donde $r_j < c_{j^*}$, es posible generar tiempos de ocio en las máquinas y disminuir su utilización.

Ejemplo

Considere un problema de 3 trabajos y 3 máquinas, M denota la máquina y T el tiempo de procesamiento de la operación, como se muestra a continuación:

M1T3	M3T5	M2T4
M2T3	M3T2	M1T3
M1T4	M2T7	M3T2

Cada operación está acompañada de sus tiempos de inicio y finalización lo más temprano posible. En el tiempo cero, es posible iniciar todos los trabajos, pero el algoritmo dice que la operación con menor tiempo de finalización se toma como referencia, en este caso las operaciones O_{11} y O_{21} tienen el menor valor ($c_{j^*} = 3$). Se selecciona al azar una operación con $r_j \leq 3$. Se coloca la siguiente operación del trabajo seleccionado, y se actualizan los tiempos de inicio

y finalización de cada operación. Así se repite sucesivamente hasta terminar con todas las operaciones. El procedimiento se muestra en la Tabla 2.5.

Tabla 2.5 Solución de un problema de 3x3 por el Método de Giffler & Thompson.

t	Trabajos						Máquinas			Parámetros		Regla
	1		2		3		1	2	3	j^*	m^*	
1	M1T3	0/3	M2T3	0/3	M1T4	0/4	0	0	0	O_{11} O_{21}	1	Azar
2	M3T5	3/8	M2T3	0/3	M1T4	3/7	3	0	0	O_{21}	2	
3	M3T5	3/8	M3T2	3/5	M1T4	3/7	3	3	0	O_{22}	3	Azar
4	M3T5	5/10	M1T3	3/5	M1T4	3/7	3	3	5	O_{23}	1	Azar
5	M3T5	5/10	M1T3	7/10	M2T7	7/14	7	3	5	O_{12} O_{23}	3	
6	M2T4	10/14	M1T3	7/10	M2T7	7/14	7	3	10	O_{23}	1	
7	M2T4	10/14	--		M2T7	7/14	10	3	10	O_{13} O_{32}	2	Azar
8	--		--		M2T7	14/21	10	14	10	O_{32}	2	
9	--		--		M3T2	21/23	10	21	10	O_{33}	3	
10	--		--		--		10	21	23			

La gráfica de Gantt queda de la siguiente manera (Figura 2.12), con un $C_{m\acute{a}x}$ de 23 unidades de tiempo:

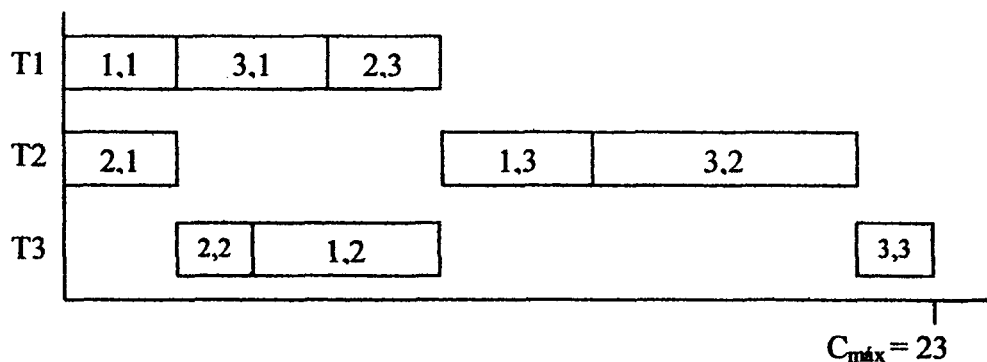


Fig. 2.12 Gráfica de Gantt de solución obtenida por Giffler & Thompson.

2.2.5 ALGORITMO DE GENERACIÓN DE PLANES SIN RETRASO

[7] Hace mención de un algoritmo que permite la generación de muchos planes sin retraso como se deseé. Básicamente, se construye un plan asignando una operación a la vez. En cualquier punto en el tiempo cada trabajo puede ser terminado o puede tener una operación a ser calendarizada, simplemente se avanza en el tiempo, y cuando una o más operaciones están listas a ser calendarizadas y la máquina está disponible, se calendariza una operación. El algoritmo es el siguiente:

Consideremos que en el punto t ,

- S_t sea el plan parcial de $(t - 1)$ operaciones calendarizadas.
- A_t sea el conjunto de operaciones calendarizables en el punto t , es decir, todas las operaciones predecesoras están en S_t .
- e_k sea el tiempo más temprano en que la operación $k \in A_t$ pueda ser calendarizada, es decir, las operaciones predecesoras estén completas y la máquina esté disponible.

Entonces:

Paso 1. Inicializar. Establecer $t = 1$, $S_1 = \emptyset$. A_t incluye la primera operación de cada trabajo listo.

Paso 2. Seleccionar operación. Encontrar $e^* = \min_{k \in A_t} e_k$. Si existen varios e^* , seleccionar uno arbitrariamente. Hacer que m^* sea la máquina necesitada por e^* . Escoger cualquier $k \in A_t$ que requiere m^* y tiene $e_k = e^*$.

Paso 3. Incrementar. Agregar la operación seleccionada k a S_t para crear S_{t+1} . Remover k de A_t y agregar la próxima operación para su trabajo a menos que ese trabajo esté completo; esto crea A_{t+1} . Establecer $t = t+1$. Si $t = MN$ detener; de otra manera ir al paso 2.

Ventajas y Desventajas: Este algoritmo tiene la ventaja sobre el de Gliffer and Thompson en que se reducen los tiempos de ocio ya que no se hacen esperar las operaciones, además de que permite el uso de reglas de despacho, sin embargo dentro del método utiliza la regla de Azar para seleccionar operaciones, lo que hace incierto el resultado.

Ejemplo

Considere el mismo problema utilizado en el método de Giffler & Thompson de 3 trabajos y 3 máquinas.

M1T3	M3T5	M2T4
M2T3	M3T2	M1T3
M1T4	M2T7	M3T2

Al inicio, en tiempo 0, la primera operación de cada trabajo se encuentran disponibles. Para la máquina 1 existen dos operaciones disponibles, así que es necesario aplicar la regla SPT, es decir, se selecciona la operación O_{11} que es la de menor tiempo de procesamiento. Se coloca la siguiente operación del trabajo seleccionado, es decir, O_{12} ; y se actualizan los tiempos de inicio y finalización de cada operación. Ahora se repite el mismo proceso hasta que se seleccionan todas las operaciones. Los resultados se muestran en la Tabla 2.6.

Tabla 2.6 Solución de un problema de 3x3 por el Método Generación de Planes sin Retraso.

t	Trabajos						Máquinas			Parámetros		Regla
	1		2		3		1	2	3	k_c^*	m^*	
1	M1T3	0/3	M2T3	0/3	M1T4	0/4	0	0	0	$O_{11} O_{21} O_{31}$	1	SPT
2	M3T5	3/8	M2T3	0/3	M1T4	3/7	3	0	0	O_{21}	2	
3	M3T5	3/8	M3T2	3/5	M1T4	3/7	3	3	0	$O_{12} O_{22} O_{31}$	3	SPT
4	M3T5	5/10	M1T3	5/8	M1T4	3/7	3	3	5	O_{31}	1	
5	M3T5	5/10	M1T3	7/10	M2T7	7/14	7	3	5	O_{12}	3	
6	M2T4	10/14	M1T3	7/10	M2T7	7/14	7	3	10	$O_{23} O_{32}$	1	
7	M2T4	10/14	--	--	M2T7	7/14	10	3	10	O_{32}	2	SPT
8	M2T4	14/18	--	--	M3T2	14/16	10	14	10	$O_{13} O_{32}$	1	
9	--	--	--	--	M3T2	14/16	10	18	10	O_{33}	3	
10	--	--	--	--	--	--	10	18	16			

La gráfica de Gantt queda de la siguiente manera (Figura 2.13), con un $C_{\text{máx}}$ de 18 unidades de tiempo:

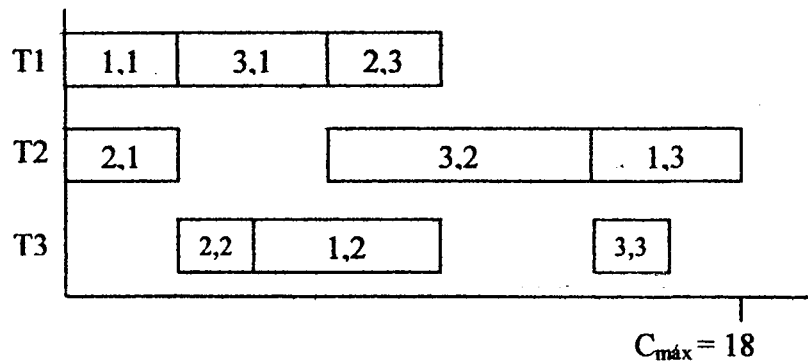


Fig. 2.13 Gráfica de Gantt de solución obtenida por Método Generación de Planes sin Retraso

2.2.6 ALGORITMO DE GENERACIÓN DE PLANES SIN RETRASO MODIFICADO

Recientemente, en un trabajo de Tesis [8] se desarrolló un algoritmo heurístico para la calendarización sin demora de sistemas de producción tipo *Job Shop*, en el cual se presenta una mejora a un algoritmo básico presentado en [7] introduciendo nuevos criterios. El algoritmo utiliza algunas reglas de despacho para priorizar los trabajos como se observa a continuación:

Paso 1. Inicialización. Se considera la primera operación de cada trabajo en turno A_t .

Paso 2. Se selecciona la operación con la próxima fecha de vencimiento (EDD). $e^* = \min_{k \in A_t} c_k$.

Paso 3. Se calcula la suma del tiempo más temprano de iniciación de la operación que pertenece al trabajo i , denotado por e_i más la ponderación del tiempo de procesamiento en la máquina j , más el tiempo de procesamiento en la máquina sucesora. $\min_{i \in A_t} \{e_i + F_{i(j+1)}\}$. En caso de empate se utiliza la regla SPT, Menor tiempo de procesamiento.

Paso 4. Se agrega la operación seleccionada k a S_t , para crear S_{t+1} . k se elimina de A_t y se coloca la operación siguiente del mismo trabajo y crear A_{t+1} . Establecer $t=t+1$. Si $t = MN$ detener, de lo contrario continuar con el paso 1.

Ventajas y desventajas: La ventaja del algoritmo es que trata en primer lugar de evitar los retrasos en la finalización del producto, y en segundo lugar de reducir el tiempo total de fabricación. La principal desventaja es que este algoritmo fue diseñado para reducir el tiempo total de fabricación, pero utiliza una regla de despacho (EDD) que no considera los tiempos de los trabajos ni número de operaciones.

Ejemplo

Considere el mismo problema utilizado en el método de Giffler & Thompson de 3 trabajos y 3 máquinas.

M1T3	M3T5	M2T4
M2T3	M3T2	M1T3
M1T4	M2T7	M3T2

Al inicio, en tiempo 0, la primera operación de cada trabajo se encuentran disponibles. La operación O_{21} es la única en solicitar a la máquina 2 así que es posible calendarizarla. Se asigna la nueva operación y se actualizan tiempos de inicio y finalización de todas las operaciones. En la segunda etapa, para la máquina 1 existen dos operaciones disponibles, así que es necesario aplicar la primera regla " $e_i + F_{i,i+1}$ " para seleccionar una.

$$O_{11} \rightarrow 0 + 3 + 5 = 8$$

$$O_{31} \rightarrow 0 + 4 + 7 = 11$$

La operación seleccionada es la O_{11} . Los resultados completos se muestran en la Tabla 2.7.

Tabla 2.7 Solución de un problema de 3x3 por el Método Dorantes-Gálvez.

t	Trabajos						Máquinas			Parámetros		Regla
	1		2		3		1	2	3	k_e^*	m^*	
1	M1T3	0/3	M2T3	0/3	M1T4	0/4	0	0	0	O ₁₁ O ₂₁ O ₃₁	2	$e_r + F_{i,i+1}$
2	M1T3	0/3	M3T2	3/5	M1T4	0/4	0	3	0	O ₁₁ O ₃₁	1	$e_r + F_{i,i+1}$
3	M3T5	3/8	M3T2	3/5	M1T4	3/7	3	3	0	O ₁₂ O ₂₂ O ₃₁	3	$e_r + F_{i,i+1}$
4	M3T5	5/10	M1T3	5/8	M1T4	3/7	3	3	5	O ₃₁	1	
5	M3T5	5/10	M1T3	7/10	M2T7	7/14	7	3	5	O ₁₂	3	
6	M2T4	10/14	M1T3	7/10	M2T7	7/14	7	3	10	O ₂₃ O ₃₂	1	$e_r + F_{i,i+1}$
7	M2T4	10/14	--		M2T7	7/14	10	3	10	O ₃₂	2	
8	M2T4	14/18	--		M3T2	14/16	10	14	10	O ₁₃ O ₃₃	3	
9	M2T4	14/18	--		--		10	14	16	O ₁₃	2	
10	--		--		--		10	18	16			

La gráfica de Gantt queda de la siguiente manera (Figura 2.14), con un $C_{máx}$ de 18 unidades de tiempo:

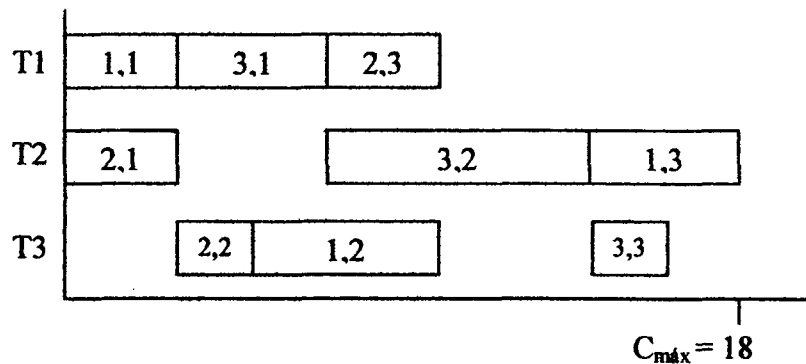


Fig. 2.14 Gráfica de Gantt de solución obtenida por Dorantes-Gálvez.

2.2.7 ALGORITMO GENÉTICO

Como su nombre lo dice, el algoritmo genético es motivado por la teoría de evolución pero que ha sido diseñado como estrategia de búsqueda general y como método de optimización.

El algoritmo genético considera una población de soluciones viables que permitan identificar y explorar las propiedades que las buenas soluciones tienen en común.

[4] muestra el siguiente esquema general para definir el funcionamiento de un algoritmo genético de enumeración. Figura 2.15.

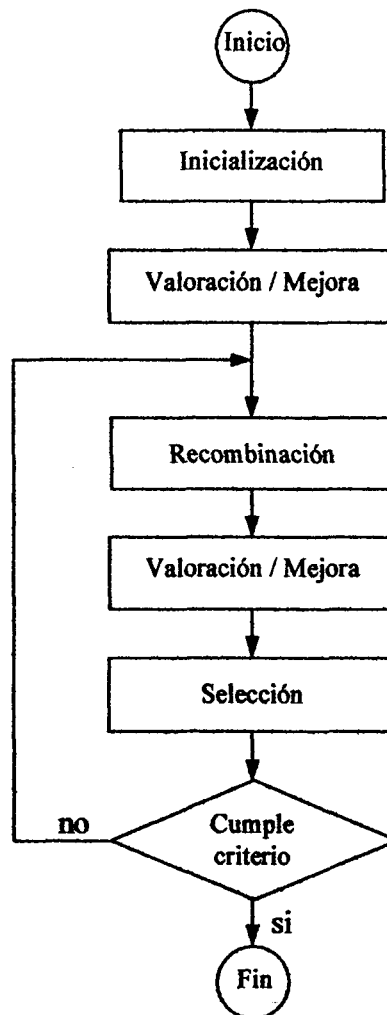


Fig. 2.15 Enumeración genética

Donde:

Inicialización. Se construye una población inicial de individuos, cada uno de los cuales es una cadena de reglas de decisión locales.

Valoración / Mejora. Evalúa cada individuo en la población actual introduciendo conocimiento específico del problema por métodos heurísticos de propósito especial (tal como investigación local) que son guiados por una secuencia de decisiones locales.

Recombinación. Extiende la población actual agregando individuos obtenidos por transformaciones unitarias y binarias (cruzamiento, mutación) en uno o dos individuos de la población actual.

Selección. Reduce la población extendida a su tamaño original de acuerdo con reglas de selección.

En un árbol de enumeración, como es el caso del método de Rama y Límite, se pueden considerar todas las posibles secuencias de decisiones, y una solución del problema estaría representada por la ruta desde la raíz hasta una de sus hojas. De la misma forma el algoritmo genético puede guiar un proceso de búsqueda con el fin de aprender y encontrar las decisiones más prometedoras dentro de un tiempo razonable.

Según [4] un algoritmo genético no trabaja muy bien en los problemas de calendarización para obtener soluciones cercanas a la óptima sin una mejora heurística. Propone dos casos diferentes en los que es posible desarrollar un algoritmo genético para la solución de problemas de calendarización tipo *Job Shop*: el primero es el algoritmo genético basado en reglas de despacho, el segundo es el algoritmo genético basado en el heurístico cambio de cuello de botella.

2.2.8 ALGORITMO GENÉTICO BASADO EN REGLAS DE DESPACHO (P-GA).

En este algoritmo, cada individuo es una cadena de $n-1$ entradas $(p_1, p_2, \dots, p_{n-1})$, donde $n-1$ es el número de operaciones del problema original. Una entrada p_i representa una regla de despacho. La entrada en la i -ésima posición significa que un conflicto en la i -ésima iteración del algoritmo de Giffler y Thompson debe ser resuelta utilizando la regla de prioridad p_i , es decir, una operación de un conjunto de operaciones en conflicto tiene que ser seleccionada por la regla p_i , en caso de empate, éste se rompe por una elección al azar.

El trabajo del algoritmo genético es determinar la mejor secuencia de reglas de despacho. Es obvio que se puede utilizar el cruzamiento más simple donde sub-cadenas de dos cadenas diferentes son intercambiadas, ya que el orden de las reglas no tiene restricciones y siempre se obtiene una descendencia factible. Figura 2.16.

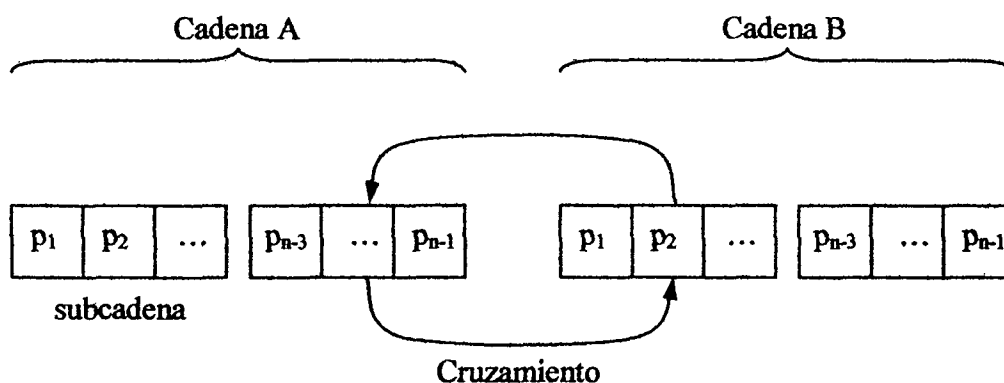


Fig. 2.16 Cruzamiento simple entre dos cadenas

Según los experimentos de [4] se recomienda el uso de reglas de despacho que son parcialmente complementarias como SPT y LPT, LOPNR y MOPNR, etc. Con el fin de que cada miembro en conflicto pueda ser elegido, de otra forma se puede obtener una convergencia prematura y excluir una solución óptima. El operador de mutación aplicado con una muy pequeña probabilidad simplemente intercambia una posición de cadena a otra; es decir, la regla de una cadena seleccionada al azar es reemplazada por una nueva regla elegida al azar entre las reglas aún no utilizadas. Ver Figura 2.17. Según los experimentos, el número máximo de reglas se puede restringir a k en lugar de $k!$ posibles permutaciones, si k es el número de operaciones que

pueden estar en conflicto. Es posible omitir esta aproximación cuando los resultados obtenidos no sean buenos.

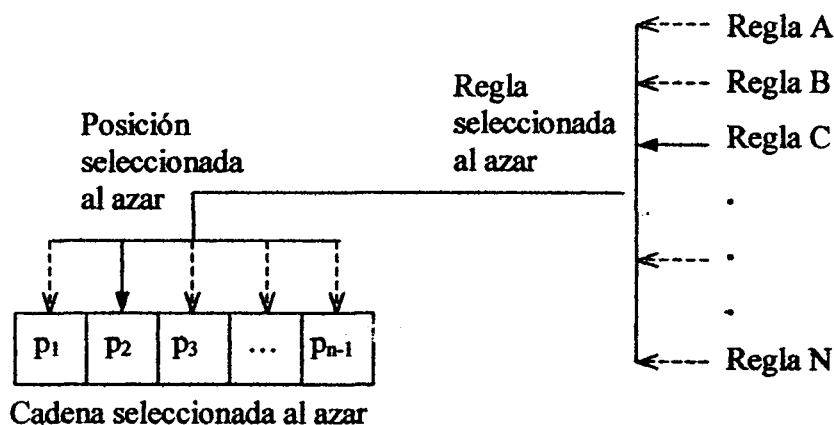


Fig. 2.17 Mutación de una cadena.

Ventajas y desventajas: La ventaja del algoritmo genético es que es capaz de encontrar una solución cercana a la óptima pero sacrificando el tiempo para el cálculo. La calidad del resultado de este algoritmo depende del tamaño de la población y de la cantidad de iteraciones que se le permitan.

2.2.9 ALGORITMO GENÉTICO BASADO EN EL HEURÍSTICO CAMBIO DE CUELLO DE BOTELLA (SB-GA).

Para poder realizar un algoritmo genético basado en el algoritmo heurístico de cambio de Cuello de Botella es necesario que el algoritmo heurístico tenga unos cambios que menciona [4]. Considerando un árbol de enumeración similar al del método de Rama y Límite, cada nodo representa el conjunto de máquinas contenidas en M' en el orden que fueron asignadas. La raíz del árbol corresponde a $M' = \{ \}$. Cada rama corresponde a la inclusión de una máquina m en el conjunto M' , así que una rama conduce a un nodo representando al conjunto $M' \cup \{m\}$, como se aprecia en la Figura 2.18. Para cada nodo le corresponde un paso simple del algoritmo cambio de Cuello de Botella.

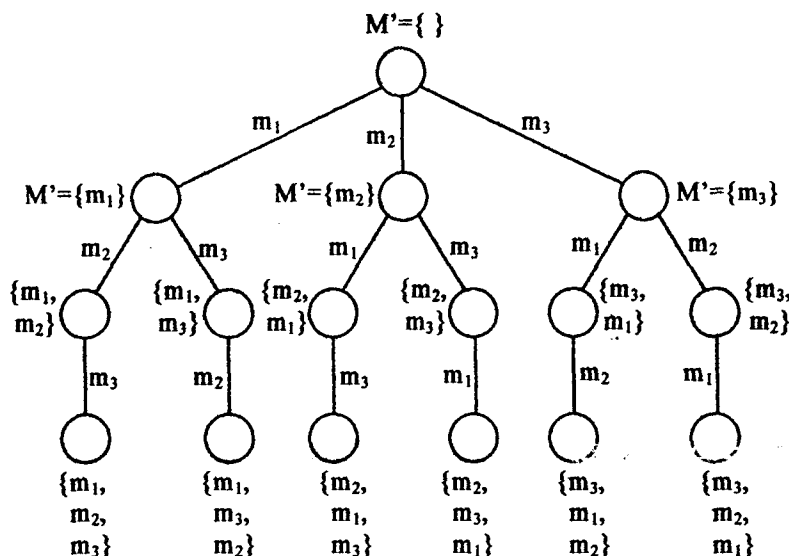


Fig. 2.18 Árbol de enumeración de M' .

Obviamente una enumeración completa del árbol es inaceptable. Por lo tanto se proponen dos rangos de búsqueda divididos por una profundidad l . En el primer rango, la búsqueda de nodos corresponde a asignar todas las ramas posibles que resulta en la inclusión de máquinas $m \notin M'$. Por lo tanto, los nodos sucesores del nodo M' corresponden al conjunto de máquinas $M' \cup \{m\}$ para todas las $m \in M \setminus M'$. El valor utilizado por [4] para l es 3, más allá de esta profundidad se aplica el criterio de botella (segundo rango); es decir, solo un nodo sucesor correspondiente a la inclusión de la máquina cuello de botella.

El algoritmo genético controla la selección de nodos en el árbol de enumeración. La longitud de la cadena que representa a un individuo en la población iguala al número de máquinas del problema, y que es igual a la profundidad del árbol de enumeración. Un individuo es codificado sobre el alfabeto desde 1 hasta el número de máquinas, y una cadena parcial desde la primera hasta la k -ésima entrada solo describe la secuencia en que las máquinas individuales son consideradas en el algoritmo heurístico. Para reducir el tiempo de cómputo durante el experimento, el algoritmo heurístico, en lugar de comenzar el ciclo de reoptimización cuando dos máquinas son secuenciadas, se inicia la reoptimización cuando menos de seis máquinas aún no se asignan a M' . El valor de 6 se obtuvo empíricamente como el mejor. La diferencia entre el método Heurístico del cambio de Cuello de Botella y el algoritmo genético es que el cuello de botella ya no es un criterio de decisión para la elección de la próxima máquina.

Ventajas y desventajas: La ventaja es que el algoritmo genético trata de optimizar la selección de nodos en el algoritmo de Cambio de Cuello de Botella. La desventaja es que el tiempo de cálculo es mucho mayor el del genético comparado con el heurístico y el resultado de los dos métodos es muy parecido.

2.3 MÉTODO ESTOCÁSTICO DE COMPARACIÓN DE ALGORITMOS

La prueba y comparación de algoritmos heurísticos ha sido un tema de mucha discusión en años recientes. Una clásica aproximación de pruebas de algoritmos es demostrar que un algoritmo propuesto es mejor, por lo menos en algunos aspectos.

Pruebas empíricas han sido el foco de investigación en varios contextos. En el contexto de programación matemática, la conducción de experimentos computacionales se ha llevado al cabo desde finales de los 70's.

En casos extremos, un algoritmo heurístico especializado puede desarrollarse excepcionalmente bien en casos particulares mientras en otros falle al tratar de producir una solución aceptable. [9] propone un método formal para la comparación y selección de algoritmos heurísticos equivalentes, o el mismo algoritmo con diferentes configuraciones. Utiliza una aproximación con técnicas estadísticas, la cual emplea técnicas de reducción de varianza para reducir el tamaño de los experimentos computacionales cuando se analizan algoritmos al azar.

Se propone un método estocástico de optimización diseñado para comparar algoritmos o configuraciones de algoritmos en un estilo eficiente cuando se presenta un conjunto de problemas. El propósito de este método es: 1) proporcionar una media de comparación de algoritmos, y 2) proporcionar un mecanismo de auto-selección de algoritmos heurísticos para identificar apropiadamente los parámetros de un algoritmo dadas las condiciones de un problema. Supóngase que se desean comparar varios algoritmos heurísticos cada uno con diferentes parámetros. Hay un total de k diferentes combinaciones de parámetros. Por conveniencia, a estas combinaciones en diferentes algoritmos se indexarán por i , donde $i = 1, 2, \dots, k$. El objetivo es

encontrar el algoritmo que se desempeñe mejor sobre un problema en particular de acuerdo a un rango específico de variación. Denotando $h_i(w)$ como el resultado de aplicar el algoritmo i dadas las variaciones de un problema caracterizado por w . $h_i(w)$ es una variable azarosa caracterizada por la variación del problema. Específicamente:

$$h_i(w_j) = E_w[h_i(w)] + \varepsilon_i(w_j)$$

donde $\varepsilon_i(w_j)$ puede ser visto como una estimación incierta o ruido, donde se puede implicar que tiene media de cero. Entonces el mejor algoritmo i se puede escoger basado en la medida: $E_w[h_i(w)] = \arg \min_i E_w[h_i(w)]$. Pero es imposible comparar con un número infinito de ejemplos, por lo que se toma una muestra n . Sin embargo en la estimación, $E_w[h_i(w)]$ converge muy lentamente conforme n tiende al infinito. Es por esto que se hace uso de la optimización ordinal, que se refiere a una aproximación general que selecciona un conjunto de alternativas del espacio de diseño basada en ciertos criterios y en un nivel de confianza específico. Suponga que se selecciona un algoritmo b utilizando el siguiente criterio:

$$b \equiv \arg \min_i E_w[h_i(w)]$$

Si se define la selección correcta (CS) como el evento que define que el algoritmo b es actualmente el mejor algoritmo, entonces la probabilidad de confianza $P\{CS\} \equiv P$. El hacer $P\{CS\}$ lo suficientemente alto, hace que la estimación de $E_w[h_i(w)]$ pueda converger lentamente. El siguiente problema es estimar el valor de $P\{CS\}$. Para problemas de la vida real, el número de diseños bajo consideración puede ser largo. Utilizando un modelo Bayesiano se puede desarrollar una técnica de estimación para cuantificar el nivel de confianza cuando el número de diseños es grande.

El método anterior se puede observar en el siguiente algoritmo.

Paso 0. Especificar un nivel de confianza satisfactorio P^* .

Realizar n_0 pruebas de todos los algoritmos.

$l \leftarrow 0$,

$N_1^l = N_2^l = \dots = N_k^l = n_0$.

Paso 1. Hacer una estimación (APCS) del valor de $P\{CS\}$ como APCS ($N_1^l = N_2^l = \dots = N_k^l$).

Si APCS ($N_1^l = N_2^l = \dots = N_k^l$) $\geq P^*$ entonces parar, de otra forma ir al paso 2.

Paso 2. Realizar τ pruebas al algoritmo i . $i = 1, \dots, k$.

$$N_i^{l+1} = N_i^{l+\tau}. \text{ Para } i = 1, \dots, k.$$

$l \leftarrow l+1$, ir al paso 1.

Ventajas y desventajas: La ventaja de este método es que permite obtener el mejor resultado de un algoritmo considerando sus variantes y compararlo con otros. Pero su desventaja radica en que es necesario desarrollar un programa para poder realizar la comparación de manera efectiva ya que por estar basado en la estadística es necesario tener una población de cada algoritmo y consume mucho tiempo.

2.4 CONCLUSIONES DEL CAPÍTULO

Los algoritmos genético han demostrado obtener resultados próximos al óptimo a cambio de consumir gran cantidad de tiempo en su procesamiento. En contraste, los algoritmos heurísticos pueden obtener muy buenas soluciones aunque no necesariamente próximas a la óptima pero con un tiempo muy reducido, además tienen mayor flexibilidad para adaptarse a diversas situaciones y objetivos presentes en la industria. En la actualidad los algoritmos heurísticos son los más utilizados en la industria, principalmente los basados en reglas de despacho por su fácil implementación y su baja complejidad.

Es posible desarrollar un algoritmo heurístico basado en reglas de despacho para reducir el tiempo de procesamiento tomando algunos puntos de otros algoritmos.

El algoritmo de Giffler & Thompson es una base importante para el desarrollo de algoritmos que utilizan reglas de despacho, sin embargo es muy rígido y da mayor facilidad para dejar tiempos de ocio. El algoritmo de Generación de Planes sin Retraso evita estos tiempos de ocio de las operaciones y deja abierta la posibilidad de seleccionar diferentes reglas de despacho, característica que es utilizada en el algoritmo de Dorantes-Gálvez creando una reglas que considera los tiempos de la operación actual y la siguiente para la toma de decisiones. El algoritmo de Intercambio de Cuello de Botella sugiere que la secuencia de operaciones de la

máquina cuello de botella es muy importante en el resultado obtenido en $C_{\text{máx}}$, y es una característica que puede ser incluida en futuros algoritmos.

El método estocástico de comparación de algoritmos es muy interesante cuando se trata de buscar un método que mejor se adecue a alguna aplicación específica, sin embargo es necesario desarrollar un programa que obtenga soluciones para cada algoritmo, lo que hace su aplicación no muy sencilla.

3. PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS.

3.1 PLANTEAMIENTO DEL PROBLEMA

Hace veinte o treinta años, la calendarización de sistemas complejos era principalmente un ejercicio académico. Antes de la aparición del Justo a tiempo (JIT), se consideraba razonable tener grandes cantidades de inventario de trabajo en proceso que absorbía los errores de calendarización en el sistema. Ahora, es ampliamente aceptado que estos inventarios sean eliminados progresivamente por una variedad de razones como: el incremento de la complejidad y la obsolescencia de los productos, mayor flexibilidad y más rápida reacción para los cambios deseados en un producto, mayor capacidad de reacción a los problemas de emergencia en el piso, etc.

La calendarización ha sido un problema que no se ha podido resolver del todo, principalmente donde el sistema de producción se vuelve más complejo y la variedad de productos y máquinas aumenta. Se han desarrollado una gran cantidad de métodos que tratan de ofrecer una mejor solución a un problema específico. [2] sugieren algunas aproximaciones como: resolver los problemas con sólo un objetivo a la vez, desarrollar curvas que relacionen objetivos, y combinar objetivos asignando costos a los deseos de los clientes y falta de utilización.

Por el conflicto que existe entre los diferentes objetivos que se buscan cumplir con la calendarización, es imposible satisfacerlos todos por completo, y es necesario priorizarlos. Es por esto que día a día se tratan de encontrar mejores algoritmos que resuelvan satisfactoriamente los problemas presentados en sistemas de producción tipo *Job Shop*, y cumplir con la relación costo-beneficio.

La calendarización de las actividades es un trabajo extremadamente importante al que se le debe dedicar mucho tiempo y estudio. El utilizar un método heurístico no nos asegura obtener la solución óptima, pero si una solución rápida, confiable y realizable para una infinidad de problemas en piso. Se pueden tener muchos métodos con ideas diferentes tratando de solucionar un problema NP, y que, dependiendo del problema unos son mejores que otros, pero la mayoría ofrece una parte de la solución.

Las reglas de despacho se han originado gracias a la experiencia en el piso de taller, observando como reacciona el sistema cuando se le da preferencia a un trabajo sobre otros y sacando conclusiones de acuerdo a las características actuales de piso.

Entonces la pregunta es ¿Se puede crear un algoritmo heurístico que permita obtener muy buenos resultados realizables en la minimización del tiempo máximo total de fabricación y en un tiempo reducido, para la calendarización de un sistema de producción tipo *Job Shop*?

Así que podemos definir el problema de la siguiente manera

“Se desea crear un algoritmo heurístico para la calendarización de actividades en sistemas de producción del tipo $N/M/G/C_{\max}$, mediante el uso de grupos de reglas de despacho; que ofrezca una nueva solución a los problemas reales de calendarización industrial”.

El tema de la presente Tesis fue propuesto como parte de un proyecto interno de investigación del Dr. Dante Jorge Dorantes González, Director del Departamento de Mecatrónica y Automatización del ITESM-CEM, donde se busca obtener un mejor método de calendarización de actividades en sistemas de producción tipo taller llamado “*Job Shop*”.

3.2 OBJETIVOS

- Desarrollar un algoritmo heurístico basado en el uso de reglas de despacho cuyo resultado sea satisfactorio en la reducción del tiempo total de fabricación.
- Desarrollar un software en lenguaje Visual Basic que automatice el algoritmo heurístico desarrollado, así como su interfaz visual de los resultados.
- Realizar una recopilación de las principales reglas de despacho utilizadas en la calendarización de actividades industriales.
- Enlistar y describir las reglas de despacho recopiladas y generar grupos de éstas cuya combinación tenga como objetivo el reducir el tiempo máximo de fabricación.
- Comparar las alternativas de solución del algoritmo heurístico y sugerir con base en este análisis los grupos de reglas de despacho que mejor se desempeñen.

4. RECOPIACIÓN DE REGLAS DE DESPACHO

A lo largo de los años se han presentado numerosos problemas de calendarización donde las reglas de despacho existentes no han podido satisfacer los objetivos solicitados; de tal manera que mucha gente ha dedicado su tiempo a generar nuevas reglas de despacho incrementando la variedad de estas. Se han escrito varios artículos donde se recopilan reglas de despacho; sin embargo, el principal problema es que en su mayoría se citan las mismas reglas. Así que en esta sección se presenta una recopilación de reglas de despacho más amplia así como su descripción de cada una.

La investigación de reglas de despacho ha estado activa por varias décadas [11], y muchas reglas diferentes han sido estudiadas en la literatura. Estas reglas han sido clasificadas de varias formas. Por ejemplo, una distinción puede ser entre reglas estáticas y dinámicas. Las reglas estáticas no dependen del tiempo, solo están en función del trabajo y/o de los datos de la máquina. Las reglas dinámicas dependen del tiempo, esto implica que en algún punto en el tiempo el trabajo j puede tener una mayor prioridad que el trabajo k , y que, más tarde el trabajo j y el trabajo k puedan tener la misma prioridad.

Una segunda forma de clasificar las reglas es de acuerdo con la información de la que están basadas. Una regla local solo utiliza información perteneciente o a la cola donde el trabajo está esperando o a la máquina donde el trabajo está formado. Una regla global puede ser utilizada para obtener información de otras máquinas, tal como tiempo de procesamiento del trabajo en la próxima máquina de su ruta, etc.

A continuación se presenta una lista de reglas de despacho comúnmente utilizadas.

- SPT (Shortest Processing Time)
- LPT (Longest Processing Time)
- EDD (Earliest Due Date)
- FCFS (First Come, First Served)
- FISFS (First In System, First Served)
- S/RO (Slack per Remaining Operation)
- COVERT (Cost Over Time)
- LWKR (Least Work Remaining)
- MWKR (Most Work Remaining)
- LTWK (Least Total Work)
- MTWK (Most Total Work)
- MOPNR (Most Operations Remaining)
- LOPNR (Least Operations Remaining)
- LORPT (Longest Operation Remaining Processing Time)
- SORPT (Shortest Operation Remaining Processing Time)
- WINQ (Work In Next Queue)
- LPUL (Largest Penalty Per Unit Length)
- SWPT (Shortest Weighted Processing Time)
- LW&LPUL (Largest Weight and Largest Penalty Per Unit Length)
- CR (Critical Ratio)
- WLWKR (Highest value of Weight divided by Least Work Remaining)
- WTWK (Largest value of Weight divided by Total Work)
- MST (Minimum Slack Time)
- OPNDD (Earliest Operational Due Date)
- SCR (Smallest Critical Ratio)
- A/OPN (Smallest Remaining Allowance per Remaining Operations)
- SPT & AWTL (Shortest Processing Time and Absolute Waiting Time Limit)
- SPT & MWTL (Shortest Processing Time and Mean Waiting Time Limit)
- FSQ (First In First Out or Shortest Processing Time according to Queue Length)

- MOD (Modified Operation Due Date)
- SPT-T (Truncated Shortest Processing Time)
- SL + SPT (Slack + Shortest Processing Time)
- CR + SPT (Critical Ratio + Shortest Processing Time)
- SPT* (Splitted Queue & Shortest Processing Time)
- MST* (Splitted Queue & Minimum Slack Time)
- CEXSPT(Conditionally Expedited by Shortest Processing Time)
- WLS (Weighted Loss of Slack)
- LOS (Longest Operation Successor)
- SOS (Shortest Operation Successor)
- MIVS (Minimum Inventory Variability Scheduling)
- FSMCT (Fluctuation Policies for Mean of the Cycle Time)
- FSVCT (Fluctuation Policies for the Variance of Cycle Time)
- RANDOM (Al azar)

Tomando en cuenta las siguientes cantidades:

- Tiempo de llegada a la cola t_{ai} .
- Duración de la operación p_i .
- Fecha de entrega de la operación o_i .
- Fecha de entrega del trabajo correspondiente d_i .
- Número de operaciones restantes en el trabajo n_i^* .
- Tiempo de procesamiento restante en el trabajo w_i^* .
- Tiempo de inactividad t_{si} .
- Tiempo presente t .

Ahora se presenta la explicación de cada una de las reglas:

SPT (Shortest Processing Time)

Menor Tiempo de Procesamiento.

Si una máquina m tiene esperando n trabajos en cola, éstos se arreglan en orden ascendente de acuerdo a sus tiempos de procesamiento p_i .

El tiempo de procesamiento es el tiempo que tarda la máquina en finalizar un trabajo. En el caso de *Job Shop*, donde el volumen de producción es muy bajo, el tiempo de procesamiento se puede considerar como: el tiempo de ajuste de la máquina para el nuevo trabajo, más el tiempo de manejo del material, más el tiempo de operación.

Considérese el siguiente ejemplo; se tienen 5 trabajos con sus respectivos tiempos de procesamiento, fechas de vencimiento y penalizaciones como se muestra a continuación:

Tabla 4.1 Datos para ejemplo de calendarización.

Trabajo	Tiempo de Procesamiento p_i	Fecha de Vencimiento d_i	Penalización L_i
1	8	17	2
2	14	53	2
3	27	49	1
4	19	32	4
5	3	8	6

Estableciendo los trabajos en orden ascendente de sus tiempos de procesamiento obtenemos la siguiente secuencia de calendarización: {5,1,2,4,3}.

LPT (Longest Processing Time)

Mayor Tiempo de Procesamiento.

Al comparar entre varios trabajos que compiten por el uso de una misma máquina, se le da prioridad al trabajo cuya operación en esa máquina requiera de mayor tiempo. El resultado es inverso a SPT. De la Tabla 4.1, la secuencia sería {3,4,2,1,5}.

EDD (Earliest Due Date)

Próxima Fecha de Vencimiento.

Los trabajos se ordenan de acuerdo a su proximidad a la fecha de vencimiento o fecha de entrega. El primero en vencerse es el primero en calendarizarse.

Considerando como ejemplo los datos de la Tabla 4.1 determinamos la secuencia de calendarización seleccionando los trabajos en orden ascendente según su fecha de vencimiento: {5,1,4,3,2}.

FCFS (First Come, First Served)

Primero en llegar, primero en servir.

En la cola de espera de una máquina, el trabajo que se calendarizará primero, será el que ha estado esperando por más tiempo (menor t_{ai}).

Considere la Figura 4.1. Los trabajos se encuentran en la cola de espera en orden de llegada, y el número entre paréntesis es el tiempo transcurrido desde que el pedido fue solicitado. Utilizando la regla FCFS el trabajo que se debe seleccionar es el T2 que se encuentra en primer lugar, posteriormente T4, T1 y finalmente T3, quedando: {2,4,1,3}.

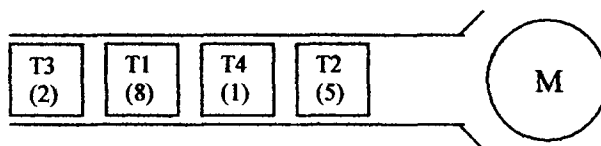


Fig. 4.1 Reglas FCFS y FISFS

FISFS (First In System, First Served)

Primero en el sistema, primero en servir.

Se selecciona el trabajo que ha estado más tiempo en el piso de trabajo.

Retomando el ejemplo de la Figura 4.1. Utilizando la regla FISFS encontramos que el trabajo T1 es el que lleva más tiempo en el sistema, por lo que debe ser el primero en calendarizarse, la secuencia sería {1,2,3,4}.

S/RO (Slack per Remaining Operation)

Menor inactividad por operaciones restantes.

Se selecciona el trabajo con la menor relación de inactividad entre operaciones restantes a ser realizadas.

Donde la inactividad se calcula:

$$t_{ai} = d_i - t - w_i^*$$

De la Tabla 4.2, considerando que el tiempo actual es 2, es posible determinar el valor de S/RO para cada trabajo:

$$S/RO_1 = (17-2-7) / 6 = 1.33$$

$$S/RO_2 = (53-2-11) / 1 = 40$$

$$S/RO_3 = (49-2-26) / 3 = 7$$

$$S/RO_4 = (32-2-19) / 11 = 1$$

$$S/RO_5 = (8-2-3) / 9 = 0.33$$

Ahora se calendarizan los trabajos en orden ascendente del valor obtenido de S/RO. La secuencia final es {5,4,1,3,2}.

Tabla 4.2 Datos para ejemplo de calendarización.

Trabajo	Tiempo de Procesamiento restante w_i^*	Fecha de Vencimiento d_i	No. de operaciones restantes n_i^*
1	7	17	6
2	11	53	1
3	26	49	3
4	19	32	11
5	3	8	9

COVERT (Cost Over Time)

Costo sobre tiempo.

Se debe considerar lo siguiente:

L_i : Penalización

p_i : Tiempo de procesamiento

d_i : Fecha de vencimiento

t_t : La suma de todos los tiempos de procesamiento

t_r : La suma de los tiempos de procesamiento de los trabajos aún no calendarizados.

s_i : Tiempo de inicio para el siguiente trabajo calendarizado, cero para el primer trabajo.

c : Coeficiente

Pr : Prioridad

Paso I: Calcular PR (Para los trabajos aún no calendarizados)

Caso 1: Si $d_i < (s_t + p_i)$ entonces $Pr = 1$

Caso 2: Si $d_i > (s_t + p_i)$ y $d_i < tt$ entonces $Pr = (tt - d_i) / (t_r - p_i)$

Caso 3: Si $tt \leq d_i$ entonces $Pr = 0$

Paso II: Calcular c para el trabajo i .

$$c_i = Pr \cdot (L_i / p_i)$$

Paso III: El trabajo con el mayor valor de c se calendariza primero.

Tomando los valores de la tabla 4.1

Iteración 1

$$tt = 8 + 14 + 27 + 19 + 3 = 71$$

$$t_r = 71$$

$$s_t = 0$$

Trabajo 1: $17 > (0+8)$, y $17 < 71$, entonces $Pr_1 = (71 - 17) / (71 - 8) = 0.857$

Trabajo 2: $53 > (0+14)$, y $53 < 71$, entonces $Pr_2 = (71 - 53) / (71 - 14) = 0.315$

Trabajo 3: $49 > (0+27)$, y $49 < 71$, entonces $Pr_3 = (71 - 49) / (71 - 27) = 0.5$

Trabajo 4: $32 > (0+19)$, y $32 < 71$, entonces $Pr_4 = (71 - 32) / (71 - 19) = 0.75$

Trabajo 5: $8 > (0+3)$, y $8 < 71$, entonces $Pr_5 = (71 - 8) / (71 - 3) = 0.926$

Porque el trabajo 5 tiene el mayor valor de Pr , es el primero en calendarizarse y se elimina de la lista de trabajos no calendarizados.

Iteración 2

$$tt = 71$$

$$t_r = (71 - \text{Tiempo de procesamiento del trabajo 5}) = (71 - 3) = 68$$

$s_t = 3$ (Tiempo de procesamiento de los trabajos calendarizados)

Trabajo 1: $17 > (3+8)$, y $17 < 71$, entonces $Pr_1 = (71 - 17) / (68 - 8) = 0.9$

Trabajo 2: $53 > (3+14)$, y $53 < 71$, entonces $Pr_2 = (71 - 53) / (68 - 14) = 0.333$

Trabajo 3: $49 > (3+27)$, y $49 < 71$, entonces $Pr_3 = (71 - 49) / (68 - 27) = 0.536$

Trabajo 4: $32 > (3+19)$, y $32 < 71$, entonces $Pr_4 = (71 - 32) / (68 - 19) = 0.795$

El trabajo 1 tiene el mayor valor de Pr, y es el siguiente en la lista de calendarizados.

Iteración 3

$$tt = 71$$

$$t_r = 68 - 8 = 60$$

$$s_t = 11$$

Trabajo 2: $53 > (11+14)$, y $53 < 71$, entonces $Pr_2 = (71 - 53) / (60 - 14) = 0.391$

Trabajo 3: $49 > (11+27)$, y $49 < 71$, entonces $Pr_3 = (71 - 49) / (60 - 27) = 0.666$

Trabajo 4: $32 > (11+19)$, y $32 < 71$, entonces $Pr_4 = (71 - 32) / (60 - 19) = 0.951$

El trabajo 4 tiene el mayor valor de Pr, y es el siguiente en la lista de calendarizados.

Iteración 4

$$tt = 71$$

$$t_r = 60 - 19 = 41$$

$$s_t = 30$$

Trabajo 2: $53 > (30+14)$, y $53 < 71$, entonces $Pr = (71 - 53) / (41 - 14) = 0.391$

Trabajo 3: $49 < (30+27)$, entonces $Pr = 1$

El trabajo 3 tiene el mayor valor de Pr, y es el siguiente en la lista de calendarizados, por lo que el trabajo 2 será el último en seleccionarse, quedando la lista de la siguiente manera: {5,1,4,3,2}.

LWKR (Least Work Remaining)

Menor tiempo de procesamiento restante.

Se selecciona el trabajo con el menor tiempo total de procesamiento de las operaciones restantes sin considerar el tiempo de la operación actual.

Por ejemplo, en la Tabla 4.3 el trabajo 1 requiere de 11 unidades de tiempo para ser finalizado, sin embargo la operación que está compitiendo por la máquina tiene una duración de 3, por lo que restarían 8 unidades. Si a todos los trabajos se les realiza la misma operación a los trabajos 1, 2, 3, 4 y 5 les restarían 8, 9, 6, 10 y 7 respectivamente. Seleccionándolos de menor a mayor el orden de trabajos quedarían como {3,5,1,2,4}.

Tabla 4.3 Datos para ejemplo de calendarización.

Trabajo	Tiempo de operaciones ya procesadas	Tiempo de la operación actual p_i	Tiempo de operaciones restantes $w_i^* - p_i$	Número de operaciones totales	Número de operaciones restantes n_i^*
1	5	3	8	4	2
2	3	2	9	6	4
3	6	5	6	3	1
4	4	6	10	5	3
5	8	1	7	9	6

MWKR (Most Work Remaining)

Mayor tiempo de procesamiento restante.

Se selecciona el trabajo con el mayor tiempo de procesamiento restante sin incluir el tiempo de la operación que compite por la máquina.

En la tabla 4.3 los trabajos 1, 2, 3, 4 y 5 tienen como tiempo restante 8, 9, 6, 10 y 7 respectivamente. Ordenando los trabajos de mayor a menor, la secuencia de prioridades queda {4,2,1,5,3}.

LTWK (Least Total Work)

Menor tiempo total de procesamiento.

Se selecciona el trabajo con el menor tiempo total de procesamiento.

De la Tabla 4.3 el tiempo total de procesamiento es la suma de las tres columnas de tiempo para cada trabajo. Para los trabajos 1, 2, 3, 4 y 5 los tiempos totales son 16, 14, 17, 20 y 16 respectivamente. Ya que los trabajos 1 y 5 tienen el mismo tiempo, es posible tener dos secuencias diferentes {2,1,5,3,4} ó {2,5,1,3,4}.

MTWK (Most Total Work)

Mayor Tiempo Total de Procesamiento.

Se selecciona el trabajo que requiere menor tiempo de procesamiento del total de sus operaciones. En el caso de LTWK las dos secuencias serían {4,3,5,1,2} o {4,3,1,5,2}.

MOPNR (Most Operations Remaining)

Mayor número de operaciones restantes.

Se selecciona el trabajo con el mayor número de operaciones restantes en su secuencia de procesamiento.

De la Tabla 4.3, la última columna muestra la cantidad de operaciones restantes para cada trabajo. Seleccionando los trabajos con mayor número de operaciones hasta el menor, la secuencia queda {5,2,4,1,3}

LOPNR (Least Operations Remaining)

Menor número de Operaciones Restantes.

Se selecciona el trabajo con el menor número de operaciones restantes en su secuencia de procesamiento. El resultado es inverso a MOPNR; es decir, la secuencia para los datos de la Tabla 4.3 es {3,1,4,2,5}.

LORPT (Longest Operation Remaining Processing Time)

Mayor Tiempo de Procesamiento de Operación Actual y Restantes .

Se le da preferencia al trabajo que requiere mayor tiempo de procesamiento para ser finalizado, incluyendo el tiempo de la operación que está compitiendo por la máquina. De la Tabla 4.3, los trabajos 1, 2, 3, 4 y 5 tienen como tiempos 11, 11, 11, 16 y 8 respectivamente. Los trabajos 1, 2 y 3 pueden ser colocados en cualquier orden. Una posible solución sería {4,1,2,3,5}.

SORPT (Shortest Operation Remaining Processing Time)

Menor Tiempo de Procesamiento de Operación Actual y Restantes .

Se le da preferencia al trabajo que requiere menor tiempo de procesamiento para ser finalizado, que incluye el tiempo de la operación que compite por la máquina más el tiempo de las operaciones restantes. El resultado es inverso a LORPT, y una de las posibles soluciones a los datos de la Tabla 4.3 sería {5,3,2,1,4}.

WINQ (Work In Next Queue)

Trabajo en la próxima cola.

Se selecciona el trabajo cuya máquina subsecuente tiene la menor cola.

Tomemos el caso de la Figura 4.2. Tres trabajos se encuentran en la cola de espera de la máquina 1, cada uno con su secuencia de máquinas o centros de trabajo. Utilizando la regla WINQ seleccionaremos el orden de preferencia en la máquina m_1 . Ya que la máquina m_3 tiene menos trabajos en espera, entonces el trabajo que pasará por m_1 y luego por m_3 es el que tienen la preferencia, en este caso es el trabajo T_2 , después le seguiría T_3 y finalmente T_1 .

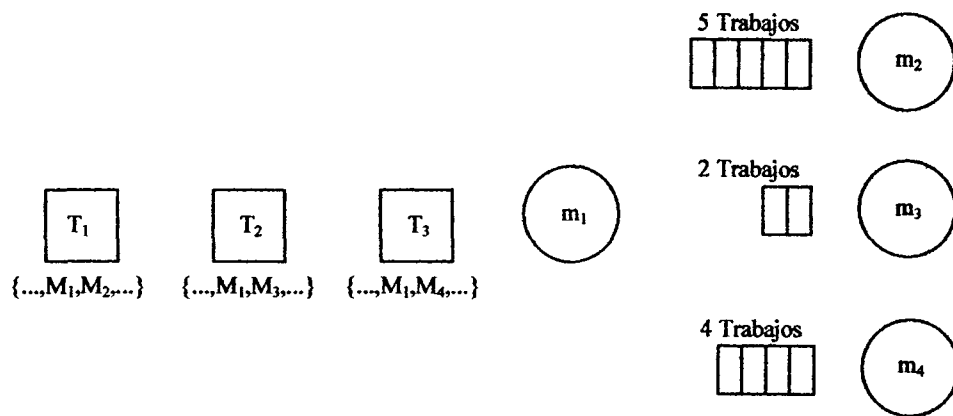


Figura 4.2 Regla WINQ

LPUL (Largest Penalty Per Unit Length)

Mayor penalización por tiempo de trabajo.

Para cada trabajo, se calcula la relación $U_i = L_i / p_i$ (penalización / tiempo de procesamiento). Se calendarizan los trabajos en orden descendente de U_i . En el caso de un empate, el trabajo con el menor tiempo de procesamiento es seleccionado.

De la Tabla 4.1

Para cada producto se calcula $U_i = L_i / p_i$

$$U_1 = 2/8 = 0.25$$

$$U_2 = 2/14 = 0.1428$$

$$U_3 = 1/27 = 0.0370$$

$$U_4 = 4/19 = 0.2105$$

$$U_5 = 6/3 = 2$$

La secuencia quedaría {5,1,4,2,3}

SWPT (Shortest Weighted Processing Time)

Menor tiempo de procesamiento penalizado.

Se calcula la relación $S_i = p_i / L_i$ (tiempo de procesamiento / penalización) para cada trabajo. Los trabajos se calendarizan en orden ascendente de S_i .

Por ejemplo, utilizando los datos de la Tabla 4.1

$$S_1 = 8/2 = 4$$

$$S_2 = 14/2 = 7$$

$$S_3 = 27/1 = 27$$

$$S_4 = 19/4 = 4.75$$

$$S_5 = 3/6 = 0.5$$

La secuencia resultante es {5,1,4,2,3}

LW&LPUL (Largest Weight and Largest Penalty Per Unit Length)

Mayor Penalización y LPUL.

Los trabajos se ordenan de forma descendente según su penalización y se calendarizan en este orden. En caso de empate se utiliza la regla LPUL.

Con los valores de la Tabla 4.1, los trabajos 1, 2, 3, 4 y 5 tienen un costo por retraso de 2, 2, 1, 4 y 6. Entonces la secuencia sería {5,4,2,1,3} o {5,4,1,2,3}.

CR (Critical Ratio)

Relación crítica.

Calcular el valor de T , el cual es la suma de los tiempos de procesamiento de todos los trabajos que han sido calendarizados. Calcular la relación Cr_i para cada trabajo i aún no calendarizado como $Cr_i = (d_i - T) / p_i$ [(fecha de vencimiento - T) / tiempo de procesamiento]. El trabajo con el menor valor de CR es el próximo en calendarizarse. Tomando los valores de la Tabla 4.1.

Iteración 1; $T = 0$

$$Cr_1 = (17-0)/8 = 2.125$$

$$Cr_2 = (53-0)/14 = 3.785$$

$$Cr_3 = (49-0)/27 = 1.814$$

$$Cr_4 = (32-0)/19 = 1.684$$

$$Cr_5 = (8-0)/3 = 2.666$$

La secuencia queda {4, , , }

Iteración 2; $T = 19$

$$Cr_1 = (17-19)/8 = -0.25$$

$$Cr_2 = (53-19)/14 = 2.857$$

$$Cr_3 = (49-19)/27 = 1.111$$

$$Cr_5 = (8-19)/3 = -3.666$$

La secuencia queda {4,5, , }

Iteración 3; $T = 22$

$$Cr_1 = (17-22)/8 = -0.625$$

$$Cr_2 = (53-22)/14 = 2.214$$

$$Cr_3 = (49-22)/27 = 1$$

La secuencia queda {4,5,1, , }

Iteración 4; $T = 30$

$$Cr_2 = (53-30)/14 = 1.642$$

$$Cr_3 = (49-30)/27 = 0.703$$

La secuencia final queda {4,5,1,3,2}

WLWKR (Highest value of Weight divided by Least Work Remaining)

Mayor valor de penalización dividido por el menor tiempo de trabajo restante.

Se calcula la relación de penalización del trabajo i entre el menor tiempo restante de procesamiento del mismo. El trabajo que dé el mayor valor en la relación es el primero en calendarizarse. Considere la Tabla 4.4.

Tabla 4.4 Datos para ejemplo de calendarización.

Trabajo	Tiempo total de Procesamiento	Tiempo Restante	Fecha de Vencimiento	Penalización L_i
1	8	5	17	2
2	14	11	53	2
3	27	15	49	1
4	19	6	32	4
5	3	1	8	6

$$T_1 = 2/5 = 0.4$$

$$T_2 = 2/11 = 0.181$$

$$T_3 = 1/15 = 0.066$$

$$T_4 = 4/6 = 0.666$$

$$T_5 = 6/1 = 6$$

La secuencia obtenida por esta regla es {5,4,1,2,3}

WTWK (Largest value of Weight divided by Total Work)

Mayor valor de penalización dividido por el trabajo total.

El primer trabajo en seleccionarse es el que resulte del mayor valor de la relación Penalización / Tiempo total. De la Tabla 4.4 se puede calcular:

$$Pr_1 = 2/8 = 0.25$$

$$Pr_2 = 2/14 = 0.142$$

$$Pr_3 = 1/27 = 0.037$$

$$Pr_4 = 4/19 = 0.21$$

$$Pr_5 = 6/3 = 2$$

La secuencia de operaciones queda {5,1,4,2,3}.

MST (Minimum Slack Time)

Menor tiempo de inactividad.

Recordando que la inactividad está dada por:

Inactividad = Fecha de Vencimiento – Tiempo presente – Tiempo de procesamiento restante.

El trabajo que tenga más ajustado el tiempo para su fabricación será el que tenga la preferencia.

Si el Tiempo presente es cero, tomando los datos de la Tabla 4.4.

$$I_1 = 17 - 0 - 5 = 12$$

$$I_2 = 53 - 0 - 11 = 42$$

$$I_3 = 49 - 0 - 15 = 34$$

$$I_4 = 32 - 0 - 6 = 26$$

$$I_5 = 8 - 0 - 1 = 7$$

La secuencia obtenida es: {5,1,4,3,2}

OPNDD (Earliest Operational Due Date)

Próxima fecha de vencimiento operacional.

Se selecciona el trabajo con el menor valor de: Fecha de vencimiento de la próxima operación – Tiempo presente (hora o fecha actual).

Es el mismo ejemplo que EDD, solo que la fecha de vencimiento es de la próxima operación a realizarse y no del trabajo total. La forma de obtener la fecha de vencimiento de cada operación, es partiendo de la fecha de entrega del trabajo e ir en retroceso restándole el tiempo de la última operación a la primera. El diagrama de red puede ser de mucha ayuda.

SCR (Smallest Critical Ratio)

Menor relación crítica.

Se calcula la relación: Tiempo de tolerancia restante / Tiempo de procesamiento restante.

El trabajo con el menor valor se calendariza primero.

Si el tiempo presente es cero, tomando los datos de la Tabla 4.4.

$$SCR_1 = (17 - 0 - 5)/8 = 1.5$$

$$SCR_2 = (53 - 0 - 11)/14 = 3$$

$$SCR_3 = (49 - 0 - 15)/27 = 1.259$$

$$SCR_4 = (32 - 0 - 6)/19 = 1.368$$

$$SCR_5 = (8 - 0 - 1)/3 = 2.333$$

La secuencia obtenida es {3,4,1,5,2}

A/OPN (Smallest Remaining Allowance per Remaining Operations)

Menor relación de tiempo de tolerancia restante por operaciones restantes.

Si la tolerancia se considera como el máximo tiempo de inactividad que puede tener un trabajo antes de la fecha de vencimiento, entonces el trabajo con menor valor de la relación: tiempo de tolerancia restante / operaciones restantes, será el primero de elegir.

Si los trabajos 1, 2, 3, 4 y 5 tienen como operaciones restantes 4, 2, 8, 3 y 1 respectivamente. Tomando los valores de la tabla 4.4, la secuencia quedaría {1,3,5,4,2}.

$$A/OPN_1 = (17 - 0 - 5)/4 = 3$$

$$A/OPN_2 = (53 - 0 - 11)/2 = 21$$

$$A/OPN_3 = (49 - 0 - 15)/8 = 4.25$$

$$A/OPN_4 = (36 - 0 - 6)/3 = 10$$

$$A/OPN_5 = (8 - 0 - 1)/1 = 7$$

SPT & AWTL (Shortest Processing Time and Absolute Waiting Time Limit)

Menor Tiempo de Procesamiento y Límite Absoluto de Tiempo de Espera.

Los trabajos se ordenan de acuerdo a la regla SPT; sin embargo, si hay algún trabajo que ha esperado más tiempo que un límite específico, el o los trabajos son ordenados de acuerdo con la regla FIFO en frente de los lotes que aún no exceden el límite.

SPT & MWTL (Shortest Processing Time and Mean Waiting Time Limit)

Menor Tiempo de Procesamiento y Límite Medio de Tiempo de Espera.

Los trabajos se ordenan de acuerdo a la regla SPT; sin embargo, si hay algún trabajo que ha esperado más que un múltiplo del tiempo de espera promedio, los trabajos son ordenados de acuerdo con la regla FIFO en frente de los lotes que aún no exceden el límite.

FSQL (First In First Out or Shortest Processing Time according to Queue Length)

Primero en Llegar Primero en Salir o Menor Tiempo de Procesamiento de acuerdo a la longitud de la cola.

Mientras el número de trabajos en cola es menor a un límite, se utiliza FCFS o FIFO, pero cuando se rebasa este límite, se utiliza SPT.

MOD (Modified Operation Due Date)*Fecha de Entrega Modificada de la Operación.*

Se selecciona el trabajo que obtiene el menor valor en el siguiente cálculo donde se comparan dos valores y se selecciona el mayor.

$$\text{MOD} = \text{máx}(t + p_i, o_i)$$

Por ejemplo; si consideramos el tiempo actual $t = 0$, y hay dos operaciones compitiendo por la misma máquina, donde las operaciones 1 y 2 tienen un tiempo de procesamiento de 5 y 3, y su fecha de vencimiento es 4 y 6 respectivamente; se calcula de la siguiente forma:

$$\text{MOD}_1 = \text{máx}(0 + 5, 4) \quad \Rightarrow \quad \text{MOD}_1 = 5$$

$$\text{MOD}_2 = \text{máx}(0 + 3, 6) \quad \Rightarrow \quad \text{MOD}_2 = 6$$

La operación que tiene el menor valor es la 1, de tal forma que es la primera en ser calendarizada.

SPT-T (Truncated Shortest Processing Time)*Menor Tiempo de Procesamiento Truncado.*

Al igual que MOD, se realiza la comparación entre dos valores y se selecciona el menor.

Se realiza el siguiente cálculo para cada operación y se selecciona la que obtenga el menor valor:

$$\text{SPT-T} = \text{mín}(p_i + r, t_{si}/n_i^*)$$

Donde r es un parámetro libre que influye en el peso del componente SPT con respecto al componente de inactividad.

SL + SPT (Slack + Shortest Processing Time)*Inactividad + Menor Tiempo de Procesamiento*

Siguiendo la misma técnica que MOD, se comparan dos valores de los que se selecciona el mayor. Se realiza el cálculo para cada operación que compite y se selecciona la que resulte menor entre todas:

$$\text{SL+SPT} = \text{máx}(p_i, t_{si}/w_i^*)$$

CR + SPT (Critical Ratio + Shortest Processing Time)*Relación Crítica + Menor Tiempo de Procesamiento*

Se selecciona primero el trabajo que resulte menor en el siguiente cálculo:

$$CR+SPT = \max(t + p_i, t + p_i \cdot (d_i - t) / w_i^*)$$

SPT* (Splitted Queue & Shortest Processing Time)*División de cola y Menor Tiempo de Procesamiento.*

La operación en la cual $t_{si} - u$ es negativo, forma una cola de alta prioridad que es procesada primero. El orden en las dos colas es de acuerdo a la regla SPT. u es un parámetro de diseño que define cuando una operación se vuelve urgente.

MST* (Splitted Queue & Minimum Slack Time)*División de cola y Menor Tiempo de Procesamiento.*

Lo mismo que SPT* excepto que la cola de alta prioridad es ordenada de acuerdo a MST en lugar de SPT.

CEXSPT(Conditionally Expedited by Shortest Processing Time)*Condicionalmente Expedido por Menor Tiempo de Procesamiento.*

Similar a SPT*; tres colas son consideradas enfrente de cada estación. La primera contiene las operaciones con inactividad negativa t_{si} (ya atrasados, tardanza inevitable), la segunda con inactividad local negativa (ha pasado la fecha de entrega operacional), y la tercera las operaciones restantes (no críticas). Todas las colas son priorizadas de acuerdo a SPT. La primera cola tiene la mayor prioridad, así que las operaciones de esta cola son las únicas calendarizadas, si ninguna de las operaciones restantes tuvieran inactividad negativa, operaciones de la segunda cola son las únicas calendarizadas, sino, alguna operación de la tercera cola tendrá una inactividad local negativa después.

WLS (Weighted Loss of Slack)*Pérdida Relativa de Inactividad*

La regla WLS compara la situación de las otras operaciones restantes en una cola si la operación j fuera calendarizada. Esta decisión causaría una pérdida de inactividad de todas las operaciones consideradas. Sin embargo, solamente es crítica si una operación tuviera inactividad

negativa t_{si} después de asignar j . Así que se define como la pérdida de inactividad de la operación i debida a la calendarización de la operación j antes que i :

$$\Delta t_{si} = \begin{cases} p_i & \text{si } t_{si} \leq 0 \\ p_i - t_{si} & \text{si } p_i > t_{si} \\ 0 & \text{si } t_{si} \geq p_i \end{cases}$$

El número de prioridad PN_j es calculado como:

$$PN_j = \sum e^{-c \cdot \min(0, t_{si})} [e^{c \cdot \Delta t_{si}} - 1]$$

La cola es dividida de nuevo en dos sub-colas. La cola de alta prioridad contiene aquellas operaciones para las cuales $t_{si} < \max_{i \neq j} p_i$; es decir, las operaciones que tienen inactividad negativa debida a la calendarización de alguna otra operación antes que la operación j . Estas operaciones críticas son ordenadas de acuerdo con la fórmula anterior. Las operaciones no críticas son calendarizadas de acuerdo con la regla OPNDD.

LOS (Longest Operation Successor)

Operación Sucesora más Larga.

Cuando varios trabajos compiten por los servicios de una misma máquina, se le da preferencia al trabajo cuya operación sucesora a la actual tenga mayor tiempo de procesamiento.

SOS (Shortest Operation Successor)

Operación Sucesora más Corta.

Cuando varios trabajos compiten por los servicios de una misma máquina, se le da preferencia al trabajo cuya operación sucesora a la actual tenga menor tiempo de procesamiento.

MIVS (Minimum Inventory Variability Scheduling)

Calendarización de Menor Variabilidad del Inventario. (Ver [12])

FSMCT (Fluctuation Policies for Mean of the Cycle Time)

Políticas de Fluctuación para la Media del Tiempo de Ciclo. (Ver [13])

FSVCT (Fluctuation Policies for the Variance of Cycle Time)

Políticas de Fluctuación para la Varianza del Tiempo de Ciclo. (Ver [13])

RANDOM (Al azar)

La secuencia no depende de un valor en particular.

5. DESARROLLO

Se pretende desarrollar un algoritmo heurístico basado en reglas de despacho que permita la calendarización de operaciones en máquinas sin demora. Posteriormente se realiza un análisis de las distintas reglas de despacho para identificar cuáles tienen características que puedan favorecer en la reducción del tiempo total de fabricación, y se presenta una nueva regla que persigue el mismo objetivo. Una vez realizada la selección de reglas, se procede a generar grupos de éstas.

El siguiente paso es utilizar el algoritmo heurístico propuesto, y realizar una comparación entre los distintos grupos de reglas para determinar cuáles tienen mejor desempeño en la reducción del tiempo total de fabricación. Para su ejecución es necesario desarrollar un programa que permita variar los datos, y a su vez observar de forma clara los resultados obtenidos.

5.1 DESCRIPCIÓN DEL ALGORITMO HEURÍSTICO

Como ya se mencionó anteriormente, el problema de calendarización para un sistema de producción tipo *Job Shop* es encontrar la mejor secuencia de operaciones en cada máquina sin alterar la secuencia de operaciones de los trabajos para cumplir con un objetivo dado. Ya que muchos objetivos pueden contradecirse entre sí, es imposible cumplir con todos ellos, de tal forma que se le debe dar prioridad a alguno.

Uno de los objetivos más importantes y de los más utilizados es el de reducir el tiempo total de fabricación. Entre los beneficios que proporciona es el de liberar a todas las máquinas del taller lo más pronto posible, evitar el tiempo de ocio de las máquinas, disminuir el tiempo de flujo promedio y reducir el costo por tardanza; en los dos últimos no necesariamente el resultado es el mejor. El algoritmo que se propone tiene como objetivo reducir el tiempo total de fabricación.

Para poder realizar el método se deben tomar las siguientes consideraciones:

- Todos los trabajos se encuentran disponibles desde el inicio de la calendarización; es decir, que están disponibles desde el tiempo cero.
- Un trabajo no puede ser procesado de forma simultanea por dos máquinas o más.
- El tiempo de procesamiento de cada trabajo es conocido y determinístico.
- Los tiempos de ajuste y transporte son independientes de la secuencia y están incluidos en el tiempo de procesamiento de los trabajos.
- Los trabajos son procesados tan pronto como sea posible.
- Todos los trabajos tienen la misma importancia, a excepción de que se indique lo contrario.
- Todas las máquinas pueden o no estar disponibles al inicio del periodo de planeación. Sólo podrán asignarse operaciones a una máquina en particular hasta que ésta se encuentre disponible.
- Como máximo, sólo puede ser procesado un trabajo en una máquina específica en un tiempo dado.
- Sólo hay una máquina de cada tipo en el taller.
- Se permite el inventario en proceso.

Una vez que se han considerado los puntos anteriores es posible crear el siguiente algoritmo basado en el de Calendarización sin Retraso con algunas modificaciones tomadas del heurístico Cambio de Cuello de Botella.

Ahora el algoritmo no necesariamente debe tener $N \cdot M$ operaciones, es posible que sean menos. Al igual que el algoritmo modificado [8] se utilizan reglas de despacho para romper el empate cuando dos operaciones solicitan los servicios de una misma máquina, la diferencia es que ahora no es obligatorio que todas las máquinas tengan la misma secuencia de reglas. Los pasos son los siguientes:

- S_t - sea el plan parcial de t operaciones calendarizadas.
- A_t - sea el conjunto de operaciones calendarizables en el punto t , es decir, todas las operaciones predecesoras están en S_t .
- e_k - es el tiempo más temprano en que la operación k pueda ser calendarizada, donde $k \in A_t$.
- e^* - es el tiempo más temprano en que la máquina m^* se encuentra disponible.
- m^* - la máquina que está disponible más temprano en el tiempo.
- m_i - una máquina del conjunto de i máquinas.

Paso 1. Establecer la secuencia general de reglas de despacho que se aplicará en caso de empate entre dos operaciones. Indicar si alguna máquina utilizará una secuencia de reglas diferente.

Paso 2. Contar el número de operaciones del problema. Es posible que unos trabajos tengan menos operaciones que otros. El valor obtenido se almacena en $OpTotal$, y no es obligatorio que $OpTotal = N \cdot M$. Establecer $t = 1$, $S_1 = \emptyset$.

Paso 3. A_t contiene la primera operación de cada trabajo preparado. El conjunto M_t contienen todas las máquinas que pueden ser utilizadas por A_t .

Paso 4. Seleccionar de M_t la máquina que se desocupa primero m^* . Si r máquinas cumplen esta condición (donde $r > 1$), seleccionar una arbitrariamente $m_i^* \in \{m_1, m_2, \dots, m_r\}$, $i = 1, 2, 3, \dots, r$.

Paso 5. Seleccionar la o las operaciones que utilizan la máquina m^* y que su tiempo de inicio e_k sea menor o igual al tiempo e^* en que se desocupa la máquina. Si ninguna operación cumple con

esto, entonces seleccionar la operación con menor e_k siempre y cuando al asignarla, la máquina m^* aún siga siendo la menor. En caso contrario, seleccionar otra máquina m_i^* y repetir el paso 5. Si todas las opciones de i fueron consideradas y no se asignó ninguna operación, incrementar el tiempo de todas las máquinas m_i^* hasta el siguiente tiempo menor en M_t . En caso de que $m_i^* = M_t$, entonces incrementar hasta $\min_{k \in A_t} e_k$. Regresar al paso 4.

Paso 6. Asignar al plan S_t la operación seleccionada. En caso de tener empate entre dos o más operaciones, utilizar la secuencia de reglas asignadas a la máquina m^* para romper el empate. Primero se aplicará la regla con mayor importancia, si el empate continúa utilizar la segunda regla y así sucesivamente.

Paso 7. Actualizar los tiempos de finalización tanto de la máquina m^* como del trabajo cuya operación fue calendarizada. Hacer $t = t + 1$. Si $t > OpTotal$ detener el algoritmo; de otra manera ir al paso 3.

5.2 ESTUDIOS PREVIOS DE REGLAS DE DESPACHO.

En los últimos años se han hecho estudios de comparación entre reglas de despacho. Una de las reglas más utilizadas es la de SPT que según el autor puede variar su nombre como SPTF “Shortest Processing Time First” [14] o como SOP “Shortest Operation Time” [4]. Sin importar el nombre que reciba, la razón de su uso es que esta regla es conocida por reducir los tiempos de ciclo; por ejemplo, en la industria de la fabricación de semiconductores, donde los precios y la tecnología utilizada han alcanzado un cierto nivel, y la capacidad de cumplir con la fecha de entrega junto con la reducción de tiempo de ciclo es probablemente el factor decisivo para competir por un lugar en el mercado global.

[14] discute algunas variantes de SPT para tratar de reducir el tiempo de ciclo en una etapa de un sistema de producción donde se pueden manejar hasta 200 productos diferentes con más de 600 operaciones cada uno como en la fabricación de semiconductores. Primero se comparan SPT, FIFO y SPT-AWTL con tiempos de espera de 5 y 10 horas. Los resultados se

muestran en la Figura 5.1.a donde SPT muestra el mejor desempeño en cuanto a tiempo de ciclo y FIFO el peor. Las curvas de SPT-AWTL se mantiene entre estas dos curvas ordenadas de acuerdo a su tiempo límite.

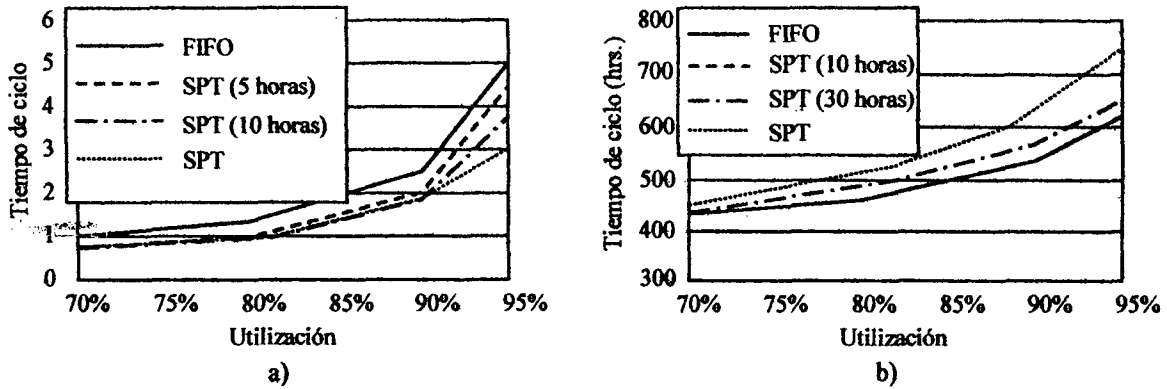


Fig. 5.1 Tiempos de ciclo promedio de SPT-AWTL

Sin embargo cuando se aplicó a un modelo completo, el tiempo de ciclo se incrementó considerablemente para SPT y se redujo para FIFO invirtiendo el orden de las curvas como se aprecia en la Figura 5.1.b. Sin embargo, cuando se varía el tiempo de procesamiento promedio de los productos, SPT vuelve a mostrar el mejor resultado y FIFO el peor.

La segunda comparación fue entre SPT, FIFO y SPT-MWTL. Recordamos que la regla se comporta exactamente igual a FIFO cuando el tiempo de espera es 0x, en las pruebas realizadas, el comportamiento de la regla SPT-MWTL es muy similar a SPT con un tiempo de espera de 10x. Los resultados se observan en la Figura 5.2.

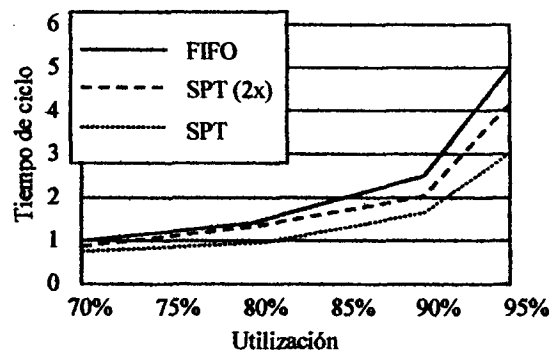


Fig. 5.2 Tiempos de ciclo promedio de SPT-AWTL

La regla SPT proporciona los mejores resultados en la reducción del tiempo de ciclo, mientras que FIFO los peores. Cuando se realizaron las pruebas con el modelo completo los resultados fueron similares a los obtenidos por SPT-AWTL, la regla que proporciona el mejor resultado depende del tiempo de procesamiento promedio de los trabajos.

La tercera y última comparación SPT, FIFO y FSQ se aplicó solamente en el modelo completo. Para una longitud de cola límite de 0 la regla se comporta como SPT, y para grandes valores el resultado se parece a FIFO. Una vez más, los resultados fueron similares al caso de SPT-AWTL. De aquí la conclusión del autor, las reglas modificadas varían entre las reglas clásicas SPT y FIFO dependiendo de los parámetros considerados. Los modelos de una etapa tienen una reducción considerable en tiempos de ciclo si la regla de despacho es cambiada de FIFO a SPT. Los parámetros de las variantes de SPT pueden ser utilizados para controlar la magnitud de este cambio. Para un modelo completo, se observa que el cambio de regla de FIFO a SPT no fue predecible. Los efectos de utilizar SPT dependen de las secuencias de operaciones y de la combinación de productos, así que es prácticamente imposible predecir la dirección y magnitud de los cambios en el tiempo de ciclo.

[15] realiza una comparación entre cinco reglas de despacho para tratar de reducir la media y la variación de los tiempos de ciclo. Las reglas son MIVS, FSMCT, FSCVT EDD y FIFO. Se utilizan dos tipos diferentes de modos de fabricación de semiconductores donde los procesos son diferentes, los residuos y reprocesos, la transportación, los tiempos de ajuste, los lotes y fallas de las máquinas. Para los dos modos de fabricación, los resultados se muestran en la Figura 5.3.

En el primer grupo de datos FSVCT obtiene los mejores resultados, seguida de MIVS, FIFO, FSMCT y EDD en orden descendente para la media, y en la varianza FSVCT continúa siendo la mejor, seguida de EDD, FSMCT, MIVS y FIFO. Los datos del segundo modo muestran una imagen completamente diferente. La menor media es obtenida por MIVS, y la mejora relativa con FIFO es de apenas 2.5%. EDD, FSMCT y FSVCT obtuvieron los peores resultados. Comparando la varianza FSVCT mejora a las otras reglas. La mejora relativa de FSVCT comparada con FIFO es de alrededor del 22%. MIVS se desempeña casi igual que FIFO.

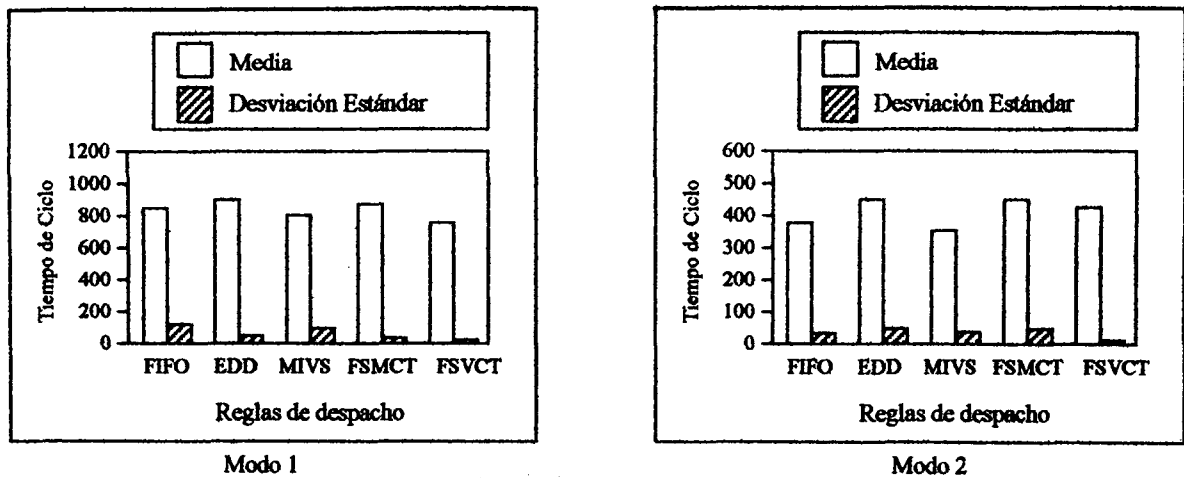


Fig. 5.3 Medias y varianza de las reglas en dos modos de fabricación.

En una recopilación de reglas de despacho realizada por [16] se resumen algunos resultados obtenidos por otros autores. Aquí se presentan éstos:

En el caso de una sola máquina,

- Si todas las operaciones pueden ser calendarizadas tal que cada una sea terminada antes de su fecha de entrega, entonces EDD logra esto.
- Si el plan que es generado por EDD es tal que solo una operación es terminada tarde, entonces EDD minimiza la tardanza media T_{media} .
- Si todos los tiempos de llegada o todos los tiempos de operación son iguales, entonces EDD minimiza la tardanza máxima $T_{máx}$.
- Si todas las fechas de entrega son iguales, entonces SPT minimiza T_{media} y FIFO minimiza $T_{máx}$.
- Si todas las operaciones son terminadas inevitablemente tarde, entonces SPT también minimiza T_{media} porque minimiza el tiempo de flujo promedio.
- Si solo existen dos operaciones que tienen el mismo tiempo de llegada, entonces MOD minimiza T_{media} . Esto ya no aplica para tres operaciones o más en cola.
- Con respecto a T_{media} y $T_{máx}$, ODD o reglas similares se pueden esperar buenos resultados donde la mayoría de los trabajos pueden ser terminados a tiempo (baja carga).
- Con respecto a T_{media} , SPT y reglas que utilizan SPT para trabajos urgentes se puede esperar un buen resultado en situaciones donde la mayoría de los trabajos se terminan tarde.

Cuando se aplican a un Sistema Flexible de Manufactura tomando como criterios T_{media} , T_{rms} (tardanza relativa), y $T_{m\acute{a}x}$; donde:

$$T_{rms} = \left[\frac{1}{N} \sum_{j=1}^N T_j^2 \right]^{1/2}$$

- ODD es el mejor en todos los criterios bajo alta presi3n en tiempo.
- SL/OPN es el mejor en todos los criterios en baja presi3n de tiempo.
- CR es mejor en tardanza media y relativa, en el segundo caso bajo media presi3n.
- CR+SPT es 3ptima con respecto a T_{media} en todas las situaciones.

5.3 ANÁLISIS DE REGLAS DE DESPACHO

Como ya se observ3 en secciones anteriores, existe una gran cantidad de reglas de despacho, cada una de las cuales le da preferencia a un trabajo sobre de otro con el fin de lograr un objetivo. Muchos autores proponen desde su punto de vista las reglas m3s utilizadas en la industria, sin embargo, las que fueron mencionadas en m3s ocasiones son SPT, EDD, FCFS y CR.

En esta secci3n se trata de dividir las reglas en tres grupos importantes que son: 1) Tiempos y operaciones, 2) Fecha de entrega y, 3) Costo, La raz3n es que los objetivos perseguidos en la calendarizaci3n generalmente tratan de controlar tiempos, costos o fechas. Los resultados se pueden observar en la Figura 5.4.

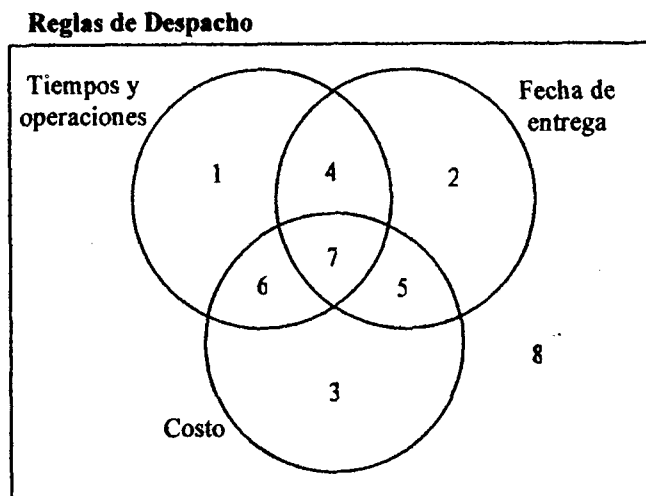


Fig. 5.4 Diagrama de Venn de reglas de despacho.

En la Tabla 5.1 se aprecia, que del total de reglas investigadas, los grupos con mayor cantidad de reglas 1,4,6 se encuentran localizados en los dominios de Tiempos y operaciones. Esto muestra el interés que se tiene en controlar el tiempo de fabricación. (En el capítulo 4 se presenta una mayor variedad de reglas ya que fueron encontradas cuando la investigación estaba adelantada, por lo que sólo se presentan para enriquecer la fuente).

Tabla 5.1 Reglas incluidas en cada conjunto de la Figura 5.4.

1	2	3	4	5	6	7	8
SPT SPT&AWTL SPT&MWTL FSQL FCFS FISFS LWKR LTWKR MOPNR MWKR WINQ LORPT	EDD OPNDD	LW	S/RO CR MST SCR A/OPN		LPUL SWPT WT&LPUL WLWKR WTWK	COVERT	RANDOM

Partiendo de este punto, para tratar de obtener el menor tiempo total de fabricación en el algoritmo propuesto, es necesario descartar todas las reglas que utilizan los criterios de fecha de entrega y costos, ya que la información que tienen para la elección de la operación no se encuentra basada en su totalidad en tiempos ni cantidad de operaciones, lo que puede influir en contra del objetivo que se refiere completamente a tiempo.

Las reglas que quedan son las del grupo 1. Sin embargo, FCFS y FISFS no toman en cuenta ni el tiempo que tarda la operación que están eligiendo, ni la cantidad de operaciones de éste y otros trabajos. La regla WINQ tiene el mismo problema que FCFS y FISFS, además de que una operación crítica puede ser detenida porque la máquina sucesora tiene mayor carga de trabajo y que puede esperar sin afectar el tiempo total de fabricación. De tal forma que también son descartadas. La regla LTWKR pierde importancia en sus datos conforme se van calendarizando operaciones ya que considera información pasada que ya no tiene tanta validez.

SPT generalmente se desempeña mejor en términos de minimizar el tiempo de flujo promedio, el promedio de trabajos en el sistema y el tiempo total de fabricación. También resulta en un bajo inventario en proceso, y proporciona un menor promedio de tardanza. La mayor desventaja de SPT es que tiende a hacer esperar a los trabajos más largos.

FCFS tiene como desventaja que trabajos muy largos pueden retrasar a otros. Sin embargo, para sistemas de servicio donde el consumidor está directamente involucrado es por mucho la regla de prioridad dominante.

La tendencia de EDD es minimizar la tardanza, pero su principal desventaja es que no toma en cuenta el tiempo de procesamiento de los trabajos.

La regla CR, tiende a minimizar la tardanza, pero empeora otros aspectos; cuando un trabajo se encuentra cerca de su fecha de entrega, o la fecha ha pasado, esta regla funciona casi igual a EDD.

Por las razones expuestas por [14], sólo se utiliza la regla SPT en representación de SPT&AWTL, SPT&MWTL y FSQ ya que el resultado de estas tres reglas oscila entre los obtenidos por SPT y FIFO, recordando que SPT obtuvo generalmente el mejor resultado. Para

efectos de comparación de resultados, se consideran las reglas principales o las más representativas de los grupos Fecha de Entrega y Costos que son EDD y LW, respectivamente.

5.4 REGLA “MTOR”

Considerando que el cuello de botella es un factor importante para el resultado del tiempo total de fabricación y que la mayoría de las reglas se aplican de forma general a cualquier máquina, se propone una nueva regla de despacho ideada para hacer un balance entre tiempos y cantidad de operaciones, tanto del trabajo a comparar como de todo el problema y tratar de dar prioridad a los trabajos que tienen más operaciones rezagadas y de mayor duración. Las ventajas que proporcionará esta regla son que la decisión se tomará basada en dos criterios principales. El primero calcula que tan significativa es la cantidad de operaciones restantes del trabajo a comparar con relación a la cantidad de operaciones de las que consta el problema; entre mayor sea esta relación, más alta será la prioridad. El segundo criterios calcula que tan significativo es el tiempo de procesamiento restante del trabajo a comparar con relación al tiempo acumulado de todas las operaciones del problema, al igual que en el primer criterio, entre mayor sea la relación, más alta será la prioridad.

La regla recibe el nombre de MTOR, Mayor número de Tiempo y Operaciones Restantes (Most Time and Operations Remaining). Su descripción es la siguiente:

- Se contabiliza el número total de operaciones que conforman el problema, OT.
- Se calcula el tiempo real que consumirán todas las operaciones del problema, TT, sin importar en qué máquina se realizarán ni a qué trabajo.
- Al igual que LORPT, se calcula el tiempo restante del trabajo a comparar, TR.
- De la misma manera que MOPNR, se calculan la cantidad de operaciones restantes del trabajo a comparar, OR.

Ahora se realiza el siguiente cálculo:

$$MTOR = \frac{OR}{OT} + \frac{TR}{TT}$$

El resultado oscilará entre 0 y 2. La operación que obtenga el valor más grande será la de mayor prioridad. Esto se puede ver como: el trabajo que tenga en promedio más operaciones restantes con mayor duración que el resto de los trabajos tendrá prioridad por la máquina.

Ejemplo

Considere el mismo problema utilizado en el método de Giffler & Thompson de 3 trabajos y 3 máquinas. Utilizar el método de Calendarización de Máquinas sin demora.

M1T3	M3T5	M2T4
M2T3	M3T2	M1T3
M1T4	M2T7	M3T2

Al inicio, en tiempo 0, la primera operación de cada trabajo se encuentran disponibles. Para la máquina 1 existen dos operaciones disponibles, así que es necesario aplicar la regla MTOR, es decir, se selecciona la operación O_{31} . Se coloca la siguiente operación del trabajo seleccionado, es decir, O_{32} ; y se actualizan los tiempos de inicio y finalización de cada operación. Ahora se repite el mismo proceso hasta que se seleccionan todas las operaciones. Los resultados se muestran en la Tabla 5.2.

Tabla 5.2 Solución de un problema de 3x3 por el Algoritmo Heurístico de Calendarización de Máquinas sin Demora.

t	Trabajos						Máquinas			Parámetros		Regla
	1	2	3	1	2	3	k_e^*	m^*				
0	M1T3	0/3	M2T3	0/3	M1T4	0/4	0	0	0	$O_{11} O_{21} O_{31}$	1	MTOR
1	M1T3	4/7	M2T3	0/3	M2T7	4/11	4	0	0	O_{21}	2	
2	M1T3	4/7	M3T2	3/5	M2T7	4/11	4	3	0	O_{11}	1	
3	M3T5	7/12	M3T2	3/5	M2T7	4/11	7	3	0			
4	M3T5	7/12	M3T2	3/5	M2T7	4/11	7	3	3	O_{22}	3	
5	M3T5	7/12	M1T3	7/10	M2T7	4/11	7	3	5	O_{23}	1	
6	M3T5	7/12	--		M2T7	4/11	10	3	5			
7	M3T5	7/12	--		M2T7	4/11	10	5	5	O_{32}	2	
8	M3T5	7/12	--		M3T2	11/13	10	11	5			
9	M3T5	7/12	--		M3T2	11/13	10	11	7	O_{12}	3	
10	M2T4	12/16	--		M3T2	12/14	10	11	12	O_{33}	3	
11	M2T4	12/16	--		--		10	11	14			
12	M2T4	12/16	--		--		11	11	14			
13	M2T4	12/16	--		--		12	12	14	O_{13}	2	
14	--		--		--		12	16	14			

La gráfica de Gantt queda de la siguiente manera (Figura 5.5), con un $C_{máx}$ de 16 unidades de tiempo:

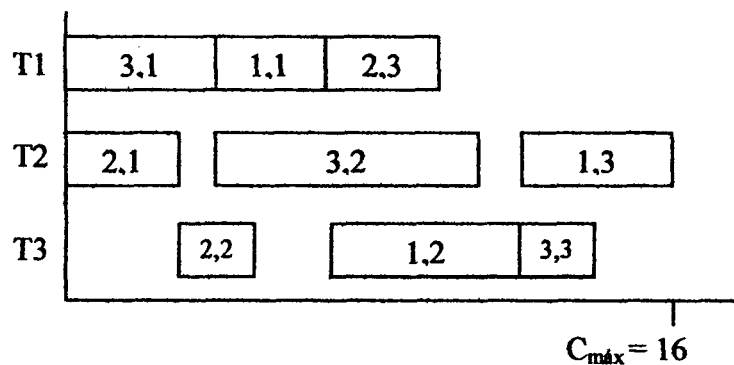


Fig. 5.5 Gráfica de Gantt de solución obtenida por Algoritmo de máquinas sin demora..

5.5 ANÁLISIS DE GRUPOS DE REGLAS Y GENERACIÓN DE SUBGRUPOS

Con el fin de determinar qué combinación de reglas funciona mejor para la solución de problemas tipo *Job Shop*, se proponen grupos de reglas que serán sometidos a prueba a 10 problemas de cada tipo: 2x2, 2x3, 2x4, 2x5, 3x2, 3x3, 3x4, 3x5, 4x2, 4x3, 4x4, 4x5, 5x2, 5x3, 5x4 y 5x5; dando un total de 160 problemas.

Por tratarse de una gran cantidad de reglas, es necesario seleccionar solo algunas para poder reducir la cantidad de soluciones a obtener. Ya que se trata de no permitir que un trabajo se atrase con respecto a los demás, las reglas que pueden aplicarse para cumplir con este objetivo son MOPNR, MWKR y LORPT; sin embargo LORPT hace la misma función que MWKR pero maneja mayor información, así que seleccionamos sólo LORPT. Estas reglas tratan de evitar que algún trabajo se retrase con respecto a los demás. La reglas SPT es seleccionada por recomendación de varios autores, ya que han comprobado que ayuda a reducir el tiempo total de fabricación y el tiempo promedio de flujo. Además, se considera una reglas basada en costos LW y una regla basada en fecha de entrega EDD, simplemente con fines de comparación con las reglas ya seleccionadas que toman en consideración operaciones y tiempos de fabricación.

Ahora se proponen grupos con las tres reglas de operaciones o tiempos SPT, LORPT y MOPNR en todas sus combinaciones, y además se agrega la regla EDD como cuarta opción en caso de que existiera un empate con las tres reglas anteriores.

Los grupos de reglas propuestos son:

Grupo 1A	Grupo 2A	Grupo 3A	Grupo 4A	Grupo 5A	Grupo 6A
SPT	SPT	MOPNR	LORPT	MOPNR	LORPT
MOPNR	LORPT	SPT	SPT	LORPT	MOPNR
LORPT	MOPNR	LORPT	MOPNR	SPT	SPT
EDD	EDD	EDD	EDD	EDD	EDD

Además, del Grupo 1A al Grupo 6A, se utilizó un grupo de reglas especial sólo para la máquina cuello de botella, que incluye las reglas en orden de importancia MTOR, MOPNR y

LORPT. Lo que se busca es observar el comportamiento de la regla MTOR al ser aplicada en el cuello de botella cuando alguno de estos grupos es utilizado como grupo general, ya que el diseño de MTOR trata de darle preferencia a los trabajos más atrasados en operaciones y con mayor duración.

El objetivo de crear la regla MTOR es para que sea aplicada a la máquina o máquinas cuello de botella, pero también se tiene interés de conocer los resultados que se obtendrían si se aplicara dentro del grupo general de reglas; por lo tanto se consideran 6 nuevos grupos en combinación con las reglas anteriores y se muestran a continuación.

Grupo 1B	Grupo 2B	Grupo 3B	Grupo 4B	Grupo 5B	Grupo 6B
MTOR	MTOR	MTOR	MTOR	MOPNR	LORPT
MOPNR	LORPT	SPT	SPT	MTOR	MTOR
LORPT	MOPNR	MOPNR	LORPT	LORPT	MOPNR
SPT	SPT	LORPT	MOPNR	SPT	SPT

5.6 PROGRAMA DE CALENDARIZACIÓN.

Se pretende desarrollar un programa que permite resolver el problema de calendarización para un sistema de producción tipo *Job Shop* mediante reglas de despacho, con el objetivo de poder obtener mayor número de resultados en la comparación de reglas y grupos de reglas que permitan una conclusión más confiable, además de que problemas grandes (mayores a 10x10) necesitan de un software para obtener una solución rápida y a prueba de errores.

El programa correrá en ambiente Windows siendo desarrollado en Visual Basic. Se utilizará el algoritmo propuesto con las mismas consideraciones. Tendrá la capacidad de cambiar las reglas de despacho, además de poder asignar distintas reglas para dos máquinas diferentes además de poder dar preferencia a algunos trabajos con un máximo de cinco. Que sea posible resolver problemas desde 1x1 hasta 100x100, y se pueda repetir una máquina en el mismo trabajo, con una capacidad máxima de operaciones de 10,000.

Los datos de entrada deben ser máquinas, tiempos, fechas de entrega y costos, los datos se guarden en forma de matriz; mientras que la información de salida se muestre en Grafica de Gantt, operaciones en orden de asignación, tiempos de finalización de las máquinas y los trabajos, y un resumen con los resultados de $C_{máx}$, tiempo de flujo promedio, tardanza máxima, tardanza promedio, máquina cuello de botella, tiempo promedio de finalización de las máquinas y utilización promedio relativo a $C_{máx}$. La estructura del programa y sus ventanas se muestra en la Figura 5.6.

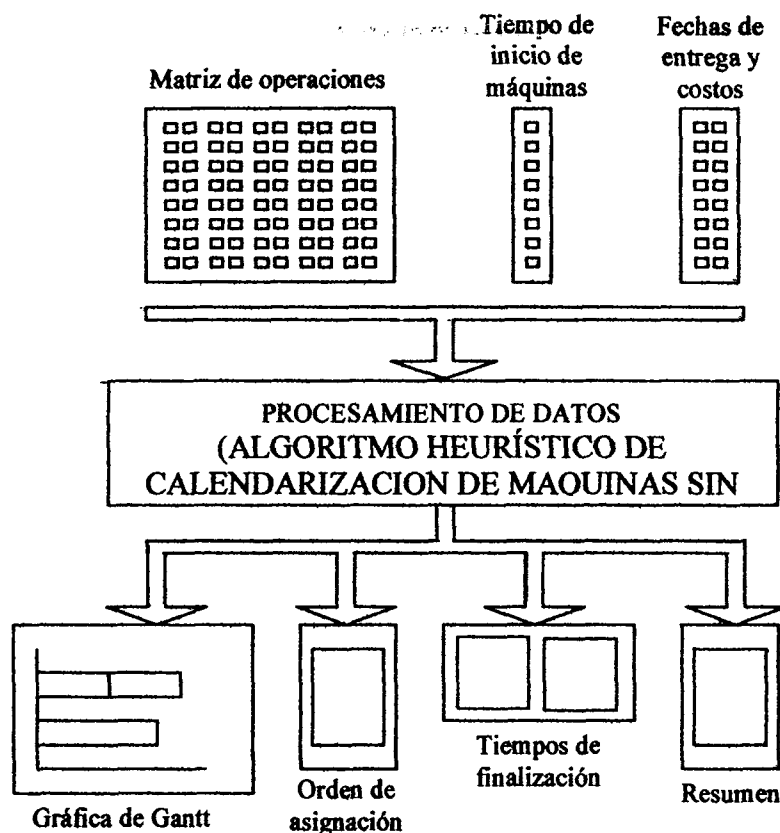


Fig. 5.6 Estructura del programa de calendarización.

El diagrama de flujo y el código del algoritmo se presenta en el Anexo A, acompañado de una pequeña explicación de cada parte.

6. RESULTADOS DEL ANÁLISIS COMPARATIVO ENTRE REGLAS DE DESPACHO

En total, para el análisis comparativo se utilizaron 160 problemas que son el resultado de 10 problemas de cada uno de los 16 tamaños diferentes: (máquinas x trabajos) 2x2, 2x3, 2x4, 2x5, 3x2, 3x3, 3x4, 3x5, 4x2, 4x3, 4x4, 4x5, 5x2, 5x3, 5x4 y 5x5; realizados al azar con trabajos que tienen como característica principal que las cantidades de operaciones pueden ser iguales o diferentes, cada trabajo utiliza una sola vez cada máquina y tienen diferentes secuencias de operaciones con variados tiempos de procesamiento, además de que las máquinas pueden tener diferentes tiempos de inicio (Ver anexo C). Los trabajos también incluyen datos como fecha de vencimiento y costos fijos por entrega tardía para ser utilizados por las reglas EDD y LW respectivamente.

6.1 ANÁLISIS COMPARATIVO ENTRE REGLAS DE DESPACHO INDIVIDUALES

Como primer paso, se resuelve cada uno de los problemas aplicando las reglas (SPT, EDD, LW, MOPNR, LORPT y MTOR) de forma individual. En caso de empate entre 2 o más trabajos, se utiliza la regla de prioridad, en caso de existir aún algún empate, se le da preferencia al trabajo colocado primero en la secuencia.

Una vez que se han resuelto todos los problemas con cada una de las reglas, se realiza un conteo de cuantas veces lograron obtener el mejor resultado de $C_{máx}$; es decir, si de 10 problemas la regla SPT obtuvo 7 veces el mejor resultado comparándose con el resto de las reglas, entonces

tuvo una efectividad del 70%. Puede existir empate entre reglas en el mejor resultado y cuentan de la misma forma. Este procedimiento se realiza para cada uno de los 16 tamaños de problema, para posteriormente calcular la Media y la Desviación Estándar entre todos los tamaños. Las tablas de valores se presentan en el Anexo B.

Los resultados obtenidos se muestran en la Figura 6.1. La regla MTOR se desempeña mejor que el resto de las otras reglas seguida por MOPNR y LORPT. Las reglas que peores resultados obtuvieron fueron EDD y LW como se esperaba, ya que no consideran en ningún momento las características de tiempo u operaciones del problema.

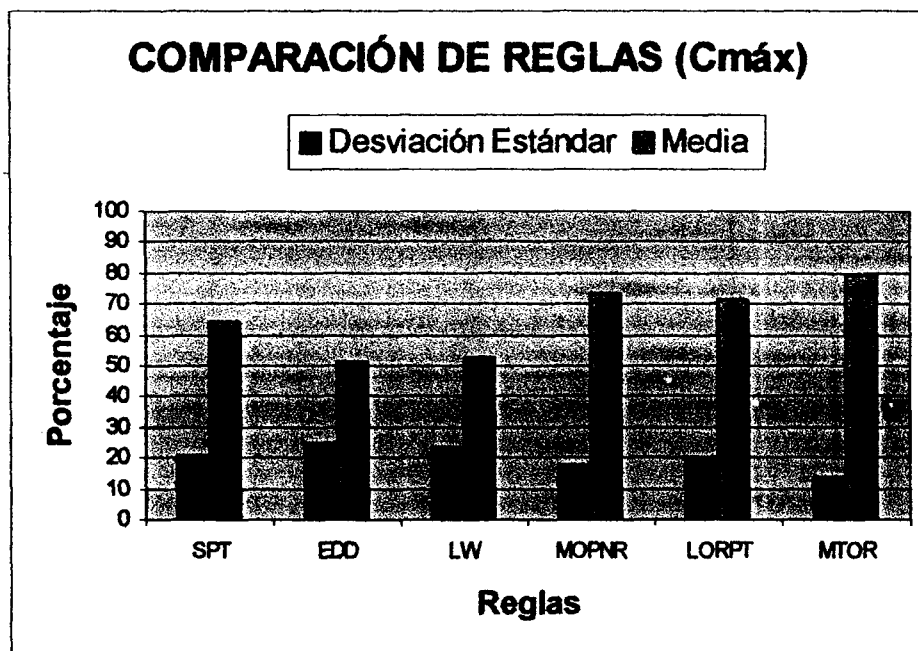


Fig. 6.1 Comparación de $C_{m\acute{a}x}$ entre reglas de despacho individuales.

Además, MTOR presenta la menor desviación estándar lo que nos sugiere una mayor consistencia en sus resultados obtenidos, mientras que EDD y LW además de tener la menor media, también tienen la mayor desviación.

6.2 ANÁLISIS COMPARATIVO ENTRE GRUPOS DE REGLAS DE DESPACHO.

De acuerdo con los resultados obtenidos en el estudio de reglas individuales, se forman los grupos de reglas ya mencionados en el capítulo 5.

Ahora, además de comparar las reglas individuales, se agregan los grupos de reglas A y B a la comparación. Además, se crean los grupos CB (Cuello de Botella), que corresponden a los grupos de reglas 1A, 2A, 3A, 4A, 5A Y 6A, pero en las máquinas cuello de botella se aplica el grupo de la regla MTOR también mencionado en el capítulo 5. Los 160 problemas se resuelven con las 6 reglas y los 18 grupos de reglas y se calcula nuevamente la media y la desviación estándar entre los 16 diferentes tamaños de problemas. Los resultados se presentan en la Figura 6.2.

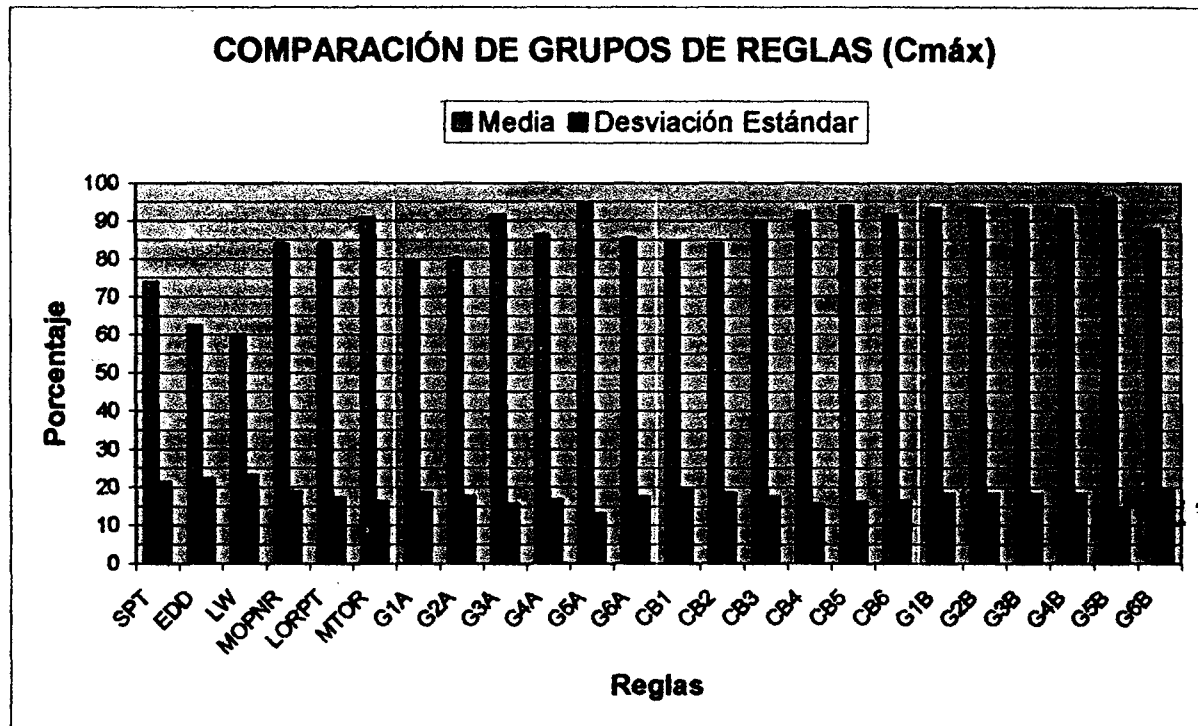


Fig. 6.2 Comparación entre reglas y grupos de reglas de despacho.

Se encuentra una diferencia marcada de las reglas MTOR, MOPNR, LORPT y SPT sobre EDD y LW en el promedio de veces que se obtiene el mejor resultado de $C_{m\acute{a}x}$. Entre los grupos

de reglas A, los grupos 3A y 5A presentan las medias más altas y las desviaciones más bajas, mientras que 1A y 2A obtuvieron resultados muy similares siendo los de peores resultados.

Se observa que cuando se aplica el grupo especial de reglas de despacho donde se incluye a MTOR en los cuellos de botella, hay una mejora; especialmente en aquellos grupos donde sólo una de las dos reglas MOPNR o LORPT aparece en una de las primeras dos posiciones de prioridad, que son los casos de 1A a 4A. Como la regla MTOR toma características de MOPNR y LORPT la mejora no es tan significativa en 5A y 6A, aunque en el caso del grupo 5A empeoró ligeramente.

Una característica importante del uso del grupo especial de reglas de despacho para cuellos de botella es que los resultados de los grupos A se homogeneizaron, y la diferencia entre grupos ya no es tan significativa.

En general, los grupos B fueron los más uniformes y con los mejores resultados comparados con los demás conjuntos de grupos; pero un punto importante es que los grupos B que obtuvieron el mejor y el peor resultado utilizan a la regla MTOR en segundo lugar de prioridad y no en primero. La regla MOPNR como antecesora de MTOR ayuda a obtener mejores resultados, mientras que LORPT produjo el efecto contrario, el resto de los grupos B donde MTOR estaba colocada en primer lugar obtuvieron resultados idénticos entre ellos, a pesar de que la segunda regla de prioridad era diferente.

Cuando se aplica MTOR a los cuellos de botella, el 71% de las veces se reduce el tiempo total de fabricación, de donde el 13% de las soluciones la máquina cuello de botella era la última en finalizar; mientras que el 29% de las veces aumenta el tiempo total de fabricación en donde el 10% de las veces la máquina cuello de botella era la última en finalizar. Entonces, es impredecible saber cuál será el resultado al aplicar el grupo especial de reglas cuando la máquina cuello de botella es la última en finalizar; cuando este no es el caso, es más probable que el resultado sea favorable. Los resultados se pueden apreciar en la Figura 6.3.

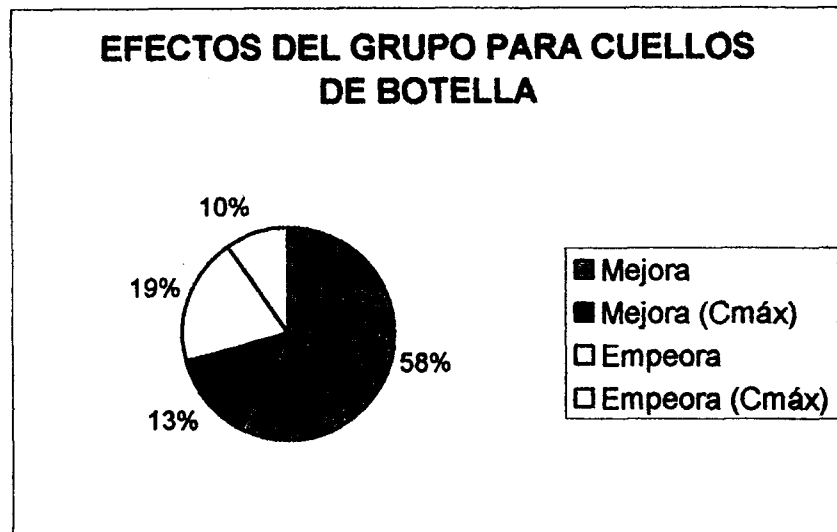


Fig. 6.3 Efectos en los grupos A por el uso del grupo de reglas para cuellos de botella.

Ajustando la gráfica de comparación de reglas y grupos de reglas de despacho, y ordenando de acuerdo a la desviación estándar de menor a mayor, se obtiene la siguiente gráfica.

Figura 6.4.

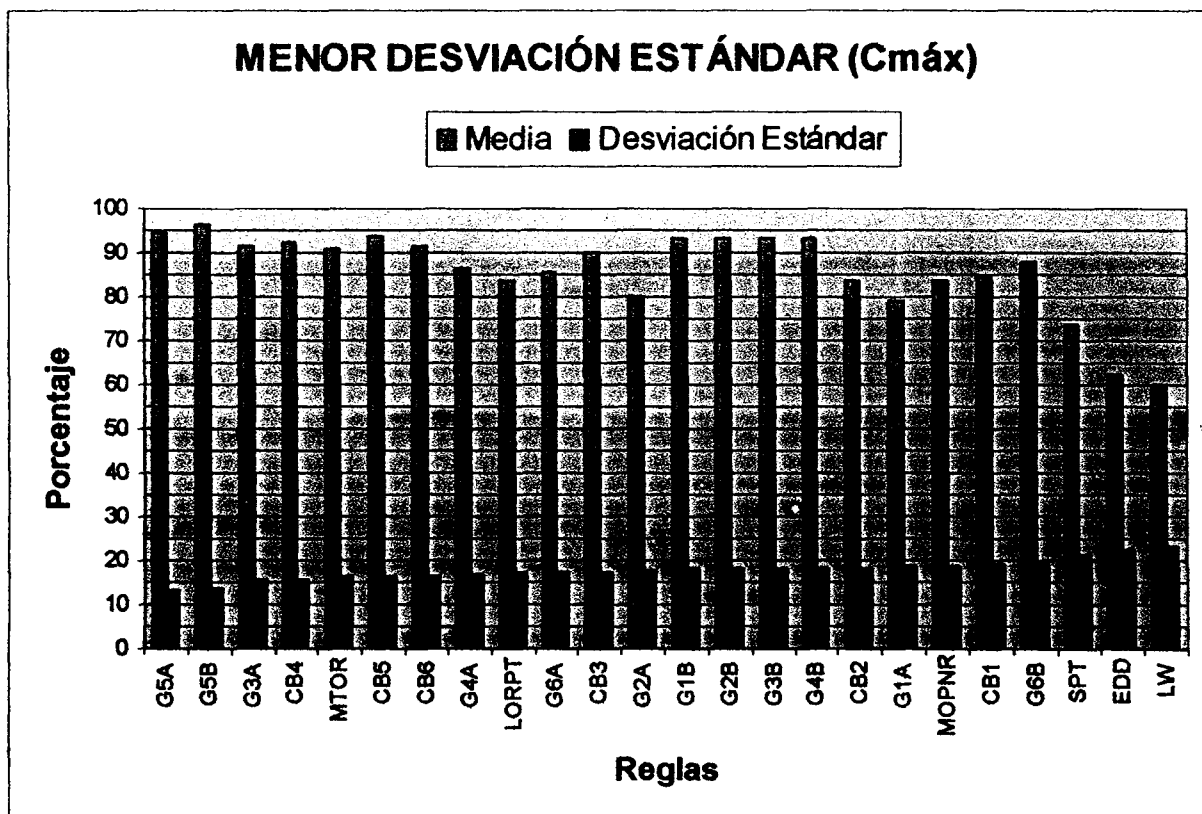


Fig. 6.4 Resultado de $C_{máx}$ ordenados de acuerdo a la desviación estándar.

Los 3 primeros grupos con menor desviación estándar resultan ser aquellos cuya regla de prioridad de mayor jerarquía es MOPNR, sin embargo la regla sola es una de las de mayor variación. El grupo CB4 se localiza en cuarta posición seguido de MTOR, que es la regla individual que mejores resultados obtuvo sobre las otras cinco. Las reglas con mayor variación fueron SPT, EDD y LW.

Ahora, los mismos grupos se ordenan de acuerdo con la media obtenida por ellos de mayor a menor. Los resultados se muestran en la Figura 6.5.

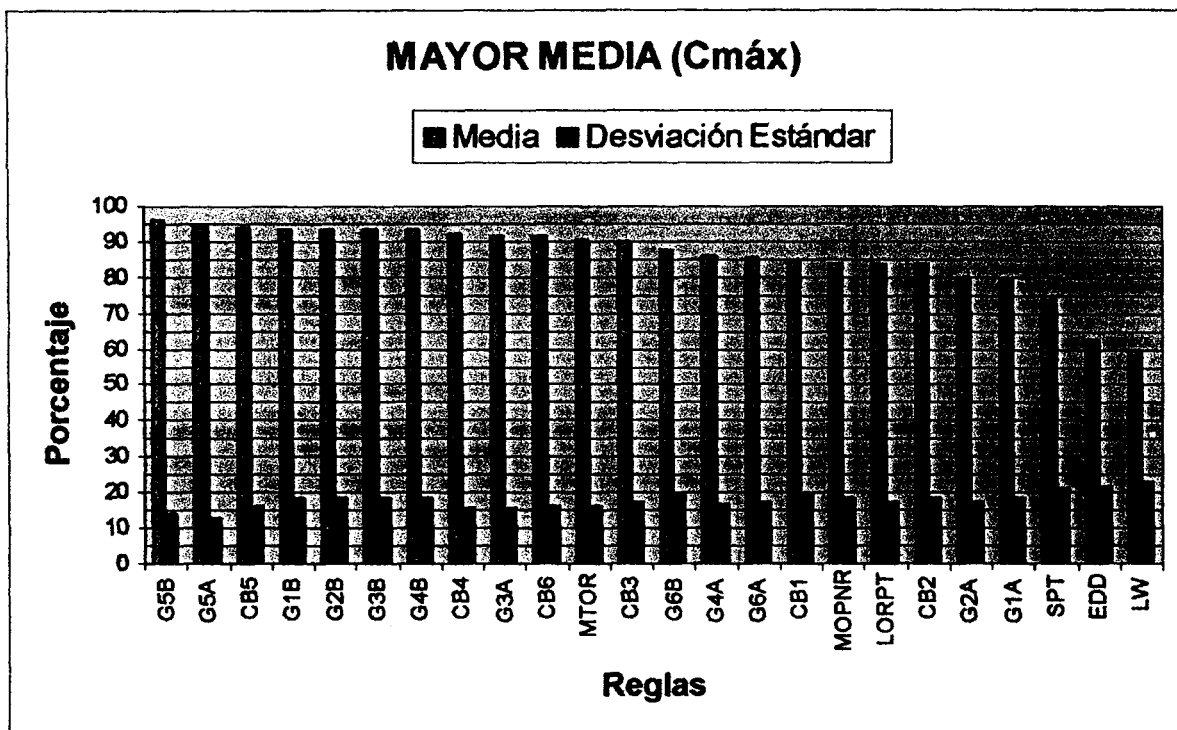


Fig. 6.5 Resultado de $C_{m\acute{a}x}$ ordenados de acuerdo a la media.

Las primeras y últimas posiciones se conservan de la misma forma que la gráfica de desviación estándar, sin embargo se aprecia de una forma más clara que los grupos B se localizan en la mitad de grupos con mejores resultados, y que MTOR sigue siendo la mejor de las reglas individuales aunque es superada por mayor número de grupos.

Analizando de forma individual los resultados obtenidos en la solución de problemas, el 76.8% de los problemas requirieron el uso de reglas de despacho para obtener la solución, mientras que en el 23.2% solamente fue necesaria la estructura principal del algoritmo. De los

problemas que no utilizaron reglas de despacho, el 94.5% se localizaron en problemas de 2 y 3 trabajos, sin importar el número de máquinas. Así que se calculó el total de problemas que requirieron el uso de reglas de despacho y se observó el número de veces que cada regla o grupo de reglas obtuvo el mejor resultado. La Figura 6.6 muestra los detalles.

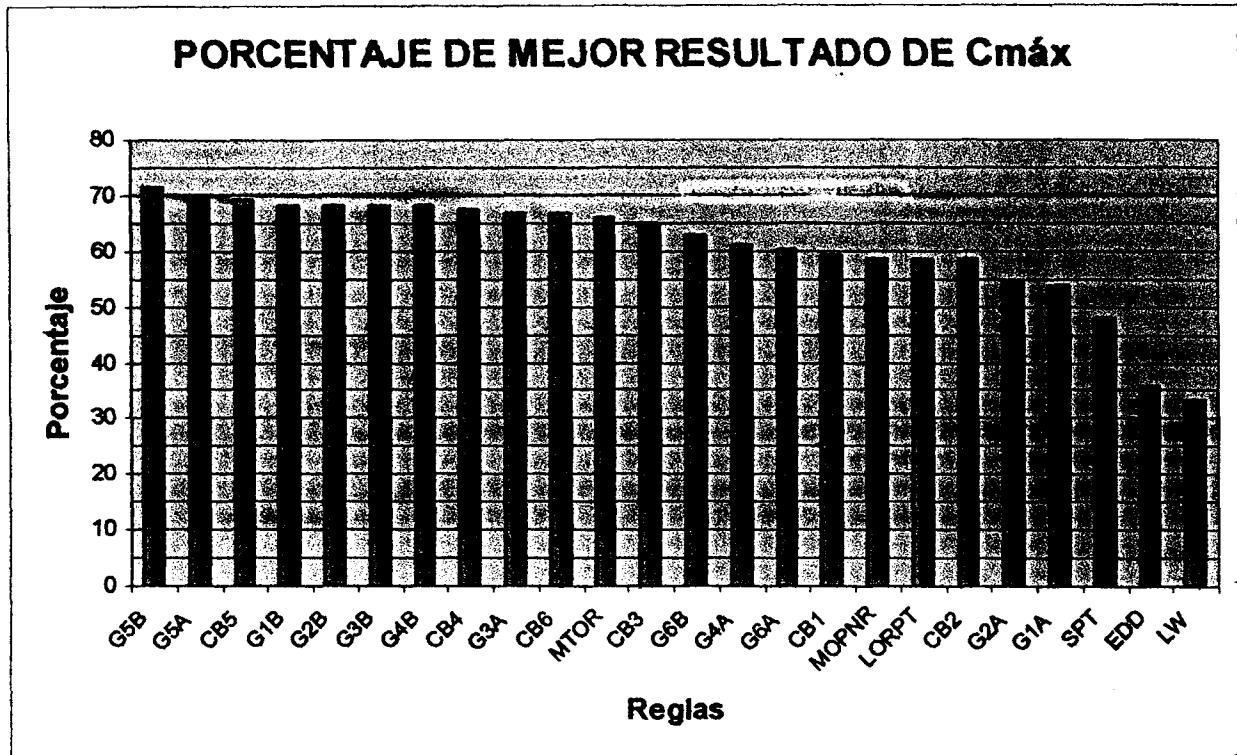


Fig. 6.6 Comparación de porcentajes de mejores resultados de $C_{m\acute{a}x}$.

El grupo 5B obtuvo el mejor resultado con 71.5%, es decir de los 123 problemas que requirieron el uso de reglas de despacho para su solución, en 88 problemas obtuvo el menor resultado de $C_{m\acute{a}x}$.

La figura muestra una ligera variación en los valores, pero conserva los mismos resultados cualitativos que las gráficas anteriores, además se puede apreciar de forma más clara la diferencia existente entre grupos de reglas aunque se conserva la misma tendencia.

G5B demuestra ser el grupo de reglas que mejor se desempeña, seguido de G5A. Los grupos B son los que obtuvieron los mejores resultados en conjunto, comparado con los grupos A y Cuello de Botella. Mientras que MTOR se desempeña mejor que el resto de las reglas aunque siendo mejorada por algunos grupos.

6.3 ANÁLISIS DE MEDIAS Y VARIANZAS

En la sección anterior se realizó una comparación de porcentajes de mejores resultados entre las reglas y grupos de reglas y determinar cuál o cuáles obtuvieron el menor $C_{\text{máx}}$ en mayor cantidad de problemas. En esta sección se presentan los resultados obtenidos al realizar un análisis de varianza. En el Anexo D se pueden apreciar los detalles.

La primera conclusión obtenida es que por la prueba de Bartlett y la prueba de Levene, las varianzas de todos los grupos y reglas son iguales ya que el P-value es mayor de 0.005. Al comparar que tan diferentes son las medias con un intervalo de confianza de 95% es que por lo menos un grupo de reglas es diferentes ya que el P-value = 0; sin embargo, al comparar de forma individual cada uno de los grupos, se observa que muchos de estos son estadísticamente iguales, y que no hay elementos suficientes para decir que son diferentes.

Hay reglas que se observan siendo las peores como EDD, LW y SPT.

Realizando una comparación múltiple con el mejor (MCB), se confirma que EDD, LW y SPT no pueden ser los mejores pero si los peores, ya que su límite superior es cero, y se logra observar que cerca de este límite se encuentran los grupos G1A, G2A, LORPT y MOPNR; ya más adelante se encuentran CB1, CB2, G4A y G6A.

En la Figura 6.7 se aprecian la diferencia de las medias con respecto a una gran media. Los grupos de reglas que se encuentran debajo de la gran media son: CB1, CB2, EDD, G1A, G2A, G4A, G6A, LORPT, LW, MOPNR y SPT; muy cerca se encuentra G6B. La similitud entre grupos se debe a que por lo general sólo se requieren de las dos primeras reglas para poder elegir una operación. Se puede apreciar de forma más clara en las soluciones de los problemas Modelo, donde varios grupos obtuvieron el mismo resultado.

Restan G3A y G5A, que son los dos únicos grupos con medias mayores a la gran media y que no utilizan a la regla MTOR. Estos grupos pueden ser utilizados cuando no existen problemas de cuello de botella y que los trabajos tengan entre ellos una gran diferencia en cantidad de operaciones.

Los grupos CB3, CB4, CB5 y CB6 se pueden utilizar cuando se presentan máquinas cuello de botella que retrasan el tiempo total de fabricación, y que el resto de las máquinas no retrasan a los trabajos con más operaciones.

Los grupos G1B, G2B, G3B, G4B, G5B tienen una gran influencia de MTOR y obtienen resultados similares, ya que es complicado obtener empates con esta regla (sólo en casos donde los trabajos tienen tiempos y cantidad de operaciones muy parecidos). Estos grupos pueden ser de gran utilidad para mantener a los trabajos durante la calendarización con aproximadamente la misma cantidad de operaciones y con un tiempo de procesamiento restante casi similar, reduciendo así el tiempo total de fabricación aunque no se mejora el tiempo de flujo promedio.

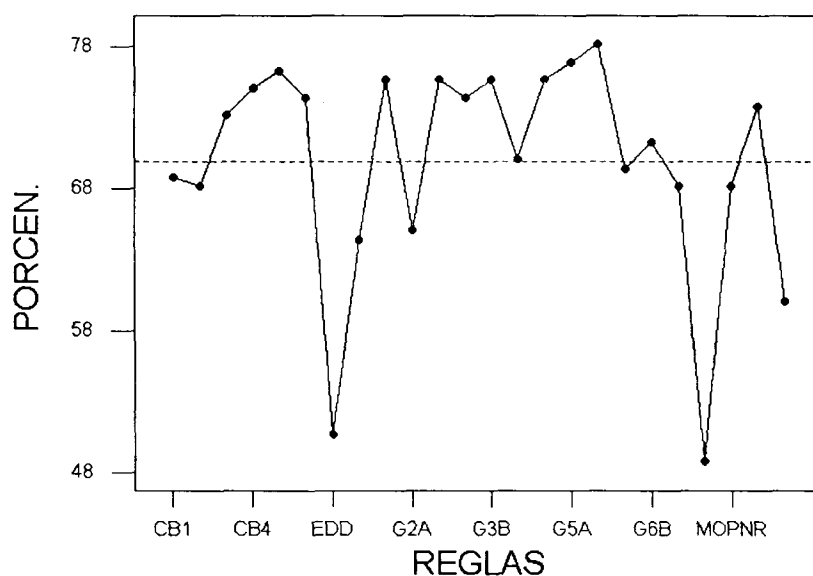


Fig. 6.7. Gráfica de medias de Grupos de Reglas de despacho

7. CONCLUSIONES

En este trabajo de Tesis se buscó un nuevo algoritmo heurístico basado en reglas de despacho tomando algunas características de algoritmos ya existentes. Fue posible obtener un algoritmo que evita el retraso en la asignación de operaciones a las máquinas, además de tener la capacidad de dar prioridad a algún trabajo dependiendo de los objetivos perseguidos, en este caso la reducción del tiempo total de fabricación o $C_{máx}$.

Se logró crear una recopilación de reglas de despacho mediante el estudio de diversas fuentes. Cada regla busca priorizar los trabajos de acuerdo a una característica del problema, ya sea basada en tiempo, cantidad de operaciones, costos, fechas de entrega, etc.

El algoritmo propuesto en esta Tesis permite el uso de una o varias reglas de despacho para dar prioridad a algún trabajo cuando existe un empate. Esta propiedad permitió combinar varias reglas de despacho y resolver un mismo problema con cada combinación para comparar los resultados del tiempo total de fabricación.

Se creó una nueva regla de despacho llamada MTOR que obtuvo mejores resultados que el resto de las reglas con las que se le comparó, ya que considera mayor número de datos del problema, y fue ideada para impedir el retraso de trabajos; sin embargo, fue superada por grupos de combinación de reglas. Una regla de despacho individual no tiene la misma efectividad que un grupo de reglas porque considera menos datos, es decir, cuando dos operaciones tienen la misma característica con la que se está buscando prioridad, es necesario buscar una característica

secundaria que también apoye en el objetivo buscado, una sola regla no tiene la capacidad de decidir con una característica secundaria.

Las reglas EDD y LW obtuvieron los peores resultados respecto al tiempo total de fabricación ya que estas tienen objetivos muy diferentes. EDD trata de evitar la entrega tardía de los trabajos, pero no considera si puede retrasar a otro trabajo y por lo tanto aumentar el $C_{máx}$. La regla LW tiene el mismo efecto que EDD solo que su objetivo es la reducción de costos.

El uso de un grupo especial para ser aplicado en las máquinas cuello de botella mejora los resultados, principalmente cuando no se incluye la combinación de reglas MOPNR y LORPT en los primeros lugares de prioridad. El beneficio que se aporta con el uso de MTOR en las máquinas cuello de botella es que se puede reducir el tiempo total de fabricación y a la vez tratar de cumplir con otro objetivo aplicando distintas reglas en las máquinas restantes.

Cuando se aplicó la regla MTOR en la calendarización de todas las máquinas mejoró los resultados obtenidos por las otras reglas, además, los grupos de reglas que incluyen esta regla fueron los que obtuvieron los mejores resultados. Sin embargo, el hecho de haber sido la regla con mejor desempeño, no significa que será mejor en todas las situaciones.

Se logró desarrollar un programa capaz de resolver problemas de calendarización tipo Job Shop mediante el algoritmo de Calendarización de máquinas sin demora, facilitando el estudio de los grupos de reglas. Además, se muestran diversas formas de presentar los datos, ya sea de forma gráfica, con resumen de resultados o características de cada operación.

Finalmente, con los resultados anteriores, es posible decir que los objetivos establecidos en este trabajo de tesis se alcanzaron satisfactoriamente y se logró hacer un aporte valioso en la investigación de algoritmos heurísticos basados en reglas de despacho.

REFERENCIAS Y BIBLIOGRAFÍA

- [1] MILTENBURG, J. *Manufacturing Strategy: How to formulate and implement a winning plan*. 1995. USA: Productivity Inc. 249-254 p.
- [2] MORTON T. E.; PENTICO D. W., *Heuristic Scheduling Systems: with applications to production systems and project management*. 1993. USA: John Wiley & Sons, Inc. 3-7, 11 p.
- [3] SULE D. R. *Industrial Scheduling*. 1997. USA: PWS Publishing Company, 14-18, 149-154 p.
- [4] DORNDORF, U.; PESCH, E. Evolution Based Learning in a Job Shop Scheduling Environment. *Computers Ops. Res.*, 1995, Vol. 22, N° 1, p. 25-40.
- [5] ADAMS, J.; BALAS, E.; ZAWACK D. The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science*, 1988, Vol. 34, N° 3, p. 391-401.
- [6] MATI, Y.; REZG, N.; XIE, X. An integrated greedy heuristic for a flexible job shop scheduling problem. *Systems, Man, and Cybernetics*, 2001, IEEE, Vol.4, p. 2534 – 2539.
- [7] ASKIN, R.; STANDRIDGE, C. *Modeling and Analysis of Manufacturing Systems*. John Wiley & Sons, Inc. 113-121 p.
- [8] GALVEZ, J. C. Desarrollo de un algoritmo heurístico para calendarización multicriterial sin demoras para sistemas de producción tipo job shop. 2001. México, ITESM-CEM.
- [9] CHUN-HUNG, C.; WU, S.D; DAI, L. Ordinal comparison of heuristic algorithms using stochastic optimization. *Robotics and Automation*, Feb. 1999, IEEE, p. 44 – 56.
- [10] CHASE, R. B.; AQUILANO N. J. *Production and Operations Management: Manufacturing and Service*. USA: Mc. Graw Hill, Octava edición, 1998, 7, 692 p.

- [11] PINEDO, M. Scheduling, *Theory, Algorithms and Systems*. Segunda Ed. 2002. USA: Prentice Hall. 336, 533-543 p.
- [12] LI, S.; T. TANG, COLLINS, D. Minimum inventory Variability Schedule with applications in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing*, p. 145-149.
- [13] LU, S.; RAMASWAMY, D.; KUMAR, P. Efficient Scheduling Policies to reduce Mean and Variation of Cycle-Time in semiconductor manufacturing plants. *IEEE Transactions on Semiconductor Manufacturing*, p. 374-388.
- [14] ROSE, O. The Shortest Processing Time First (SPTF) Dispatching Rule and some variants in Semiconductor Manufacturing. *2001 Winter Simulation Conference*, 2001, IEEE, p. 1220-1224.
- [15] MITTLER, M; SCHOEMING, A. comparison of Dispatching Rules for semiconductor manufacturing using large facility models. *1999 Winter Simulation Conference*, 1999, IEEE, p. 709-913.
- [16] MOSER, M.; ENGELL, S. A survey of Priority Rules for FMS Scheduling and their performance for the Benchmark Problem. *31st Conference of Decision and Control*, 1992, IEEE, p. 392-397.

ANEXO A

A1. LISTA DE VARIABLES

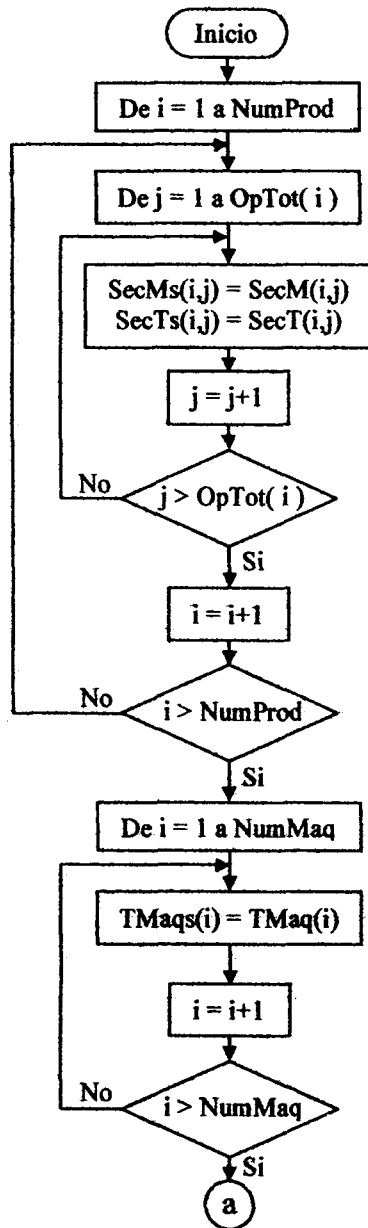
Ac	Variable de número de operación seleccionada.
Acum1	Variable de uso general.
Ar	Variable de número de trabajo seleccionado.
c(w)	Variable de memoria de orden de selección de operaciones (columnas).
Cant	Cantidad de operaciones empatadas.
Comp(i)	Variable de trabajos a comparar.
Cont	Contador de operaciones empatadas.
Costo(i)	Registros para almacenar costos por entregas tardías.
i	Variable de filas en arreglo de trabajos o máquinas. Uso general.
j	Variable de columnas en arreglo de trabajos o máquinas. Uso general.
k	Contador de operaciones calendarizables.
M(Ar,Ac)	Máquinas de las operaciones seleccionadas.
MaqX	Número de máquina con grupo de reglas especial 1.
MaqY	Número de máquina con grupo de reglas especial 2.
MOPNRS(i)	Variable de cálculo para regla MOPNR.
MTORS(i)	Variable de cálculo para regla MTOR.
LORPTS(i)	Variable de cálculo para regla LORPT.
NumMaq	Variable de entrada de número de máquinas.
NumProd	Variable de entrada de número de trabajos.
Op(i)	Contador de cantidad de operaciones seleccionadas.
OpTot(i)	Cantidad de operaciones de cada trabajo.
OpTotal	Cantidad de operaciones de todo el problema.
Perm	Bandera que indica la existencia de una permutación.
r(w)	Variable de memoria de orden de selección de operaciones (filas).
s	Variable utilizada para realizar ciclos.
SecM(i,j)	Arreglo de máquinas del problema.
SecMs(i,j)	Copia para modificar del arreglo de máquinas del problema.
SecT(i,j)	Arreglo de tiempos del problema.
SecTs(i,j)	Copia para modificar del arreglo de tiempos del problema.
SelecM(i)	Columna de máquinas de primera operación de cada trabajo.
SellecT(i)	Columna de tiempos de primera operación de cada trabajo.
Serie(i)	Variables de tiempo de inicio de cada máquina.
TDisp(i)	Variable de tiempo de finalización de la última operación calendarizada.
TF(i,j)	Variable de memoria de tiempos de finalización de cada operación.
TI(i,j)	Variable de memoria de tiempos de inicio de cada operación.
TiEnt(i)	Registros para almacenar tiempos de entrega de cada trabajo.
TiMaq(i)	Copia para modificar de TMaqs(i).
TMaq(i)	Tiempo más temprano de asignación de una máquina.

TMaqs(i)	Tiempo más temprano de asignación de una máquina a modificar.
TT	Suma del tiempo total de todos los trabajos.
w	Contador de ciclo de ejecución del algoritmo.

A2. DIAGRAMA DE FLUJO

El diagrama de flujo del programa se presenta a continuación; acompañado de una pequeña explicación de cada parte.

-----Hacer una copia de los datos-----

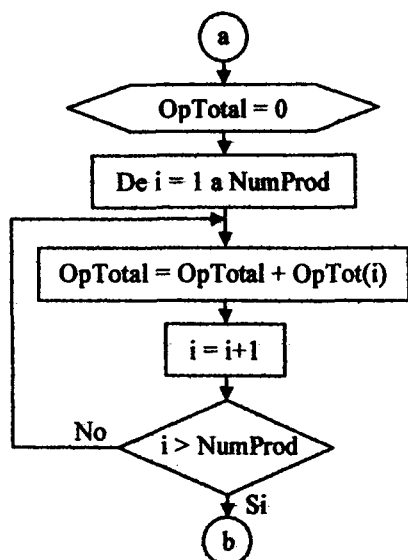


Se realiza una copia de todos los datos de entrada, que se encuentran almacenados en un arreglo rectangular (matriz) para que sean modificados posteriormente.

SecM(i,j) representa la máquina utilizada por el trabajo i durante la operación j. De la misma manera SecT(i,j) representa el tiempo del trabajo i para la operación j.

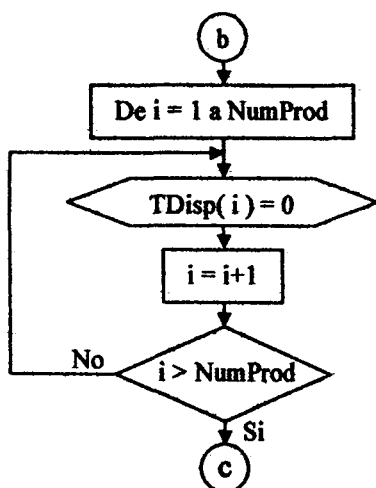
La variable TMaq(i) representa el tiempo en que se encuentra disponible la máquina i.

-----Calcular operaciones totales.-----



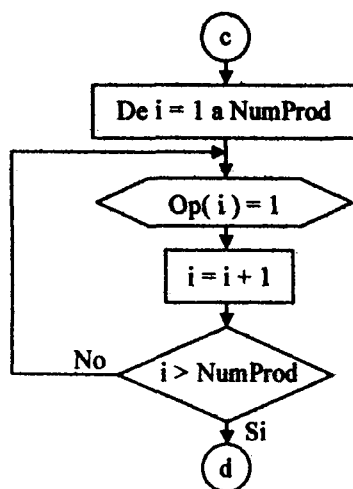
Cada trabajo i tiene una variable con su número de operaciones totales $OpTot(i)$ que se contabilizan en una variable general $OpTotal$.

-----Inicio del tiempo disponible de cada trabajo.-----



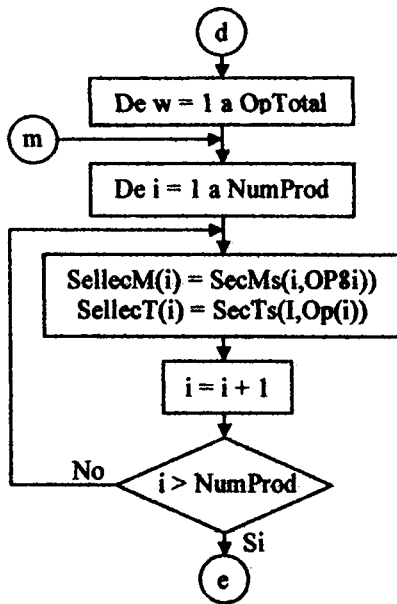
Se determina que todos los trabajos se encuentran disponibles desde el tiempo cero. $TDisp(i)$ indica el tiempo en que estará disponible el trabajo i , que irá variando cuando se asigne cada operación.

-----Iniciar con la primera operación de cada trabajo.-----



La variable $Op(i)$ hace la función de indicador. Establece el número de operación del trabajo i que sigue por ser procesada.

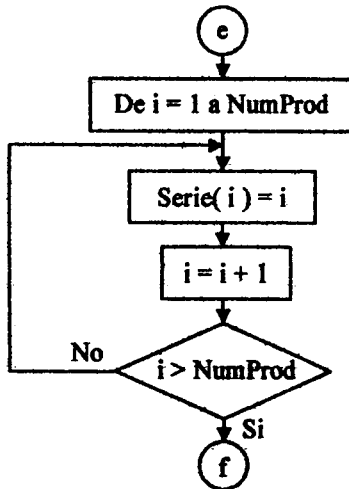
-----Inicia ciclo. Seleccionar primera columna de operaciones.-----



El ciclo w se repetirá hasta asignar el total de operaciones del problema.

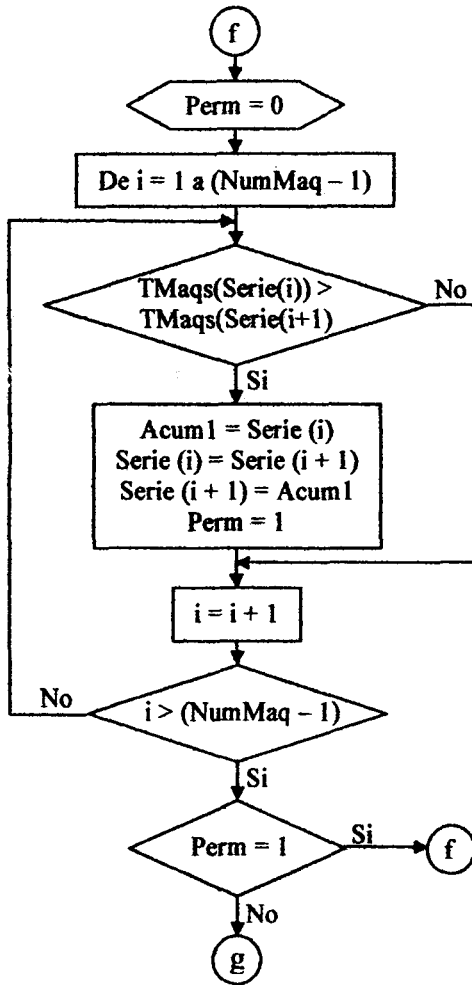
En las variables SellecM(i) y SellecT(i) se guardan las máquinas y tiempos de todas las operaciones que ya pueden ser calendarizadas.

-----Inicializar secuencia de máquinas.-----



La variable Op(i) hace la función de indicador. Establece el número de operación del trabajo i que sigue por ser procesada.

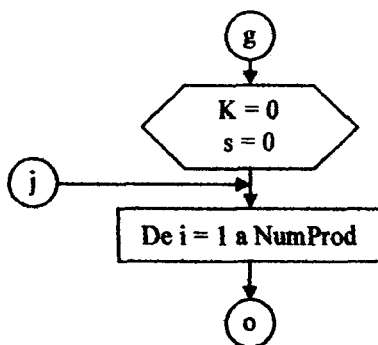
-----Ordenar máquinas en tiempos de menor a mayor.-----



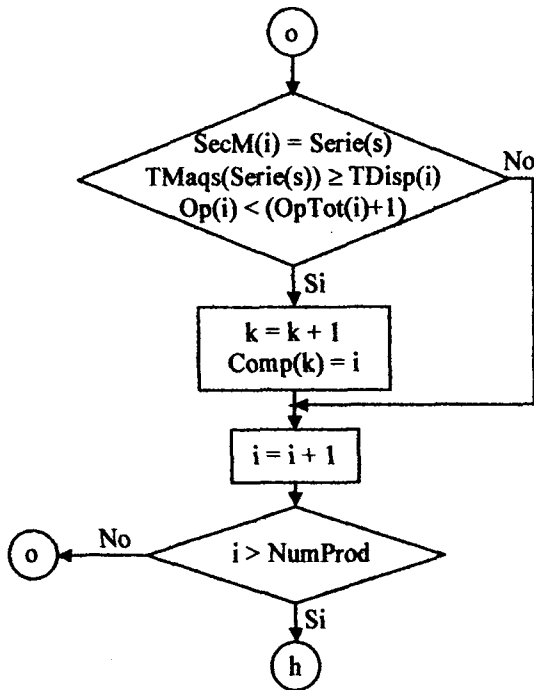
Se colocan las máquinas en orden ascendente de acuerdo al tiempo en que se encuentran o encontrarán libres.

Se compara el tiempo de la primera máquina con el de la segunda, si es mayor, se intercambian posiciones y la bandera Perm cambia de estado a 1 indicando que sí hubo permutación en el orden entre dos máquinas. El proceso se repite hasta que ya no haya permutaciones y las máquinas estén ordenadas desde la primera disponible hasta la última.

-----Seleccionar los trabajos con operaciones a comparar.-----

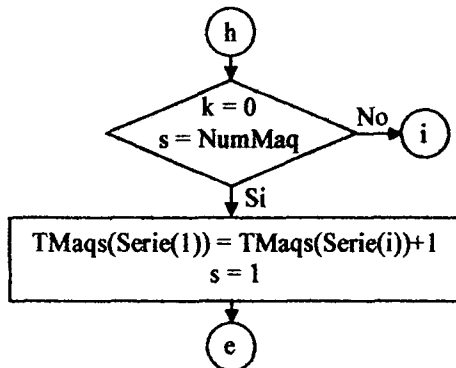


k indicará la cantidad de trabajos que pretenden utilizar una misma máquina al mismo tiempo.



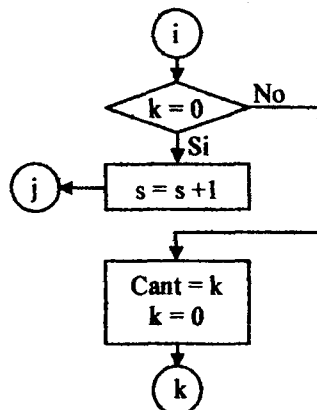
Si la siguiente operación del trabajo i utiliza la máquina Serie(s) y, el tiempo en que estará disponible la máquina Serie(s) es más tarde o igual que el tiempo en que estará disponible el trabajo i y, que el trabajo i todavía tenga operaciones por calendarizar; entonces el trabajo i con su operación actual $Op(i)$ está disponible para competir por la máquina solicitada y se almacena en la variable $Comp(k)$.

-----Incrementar el tiempo en las máquinas.-----



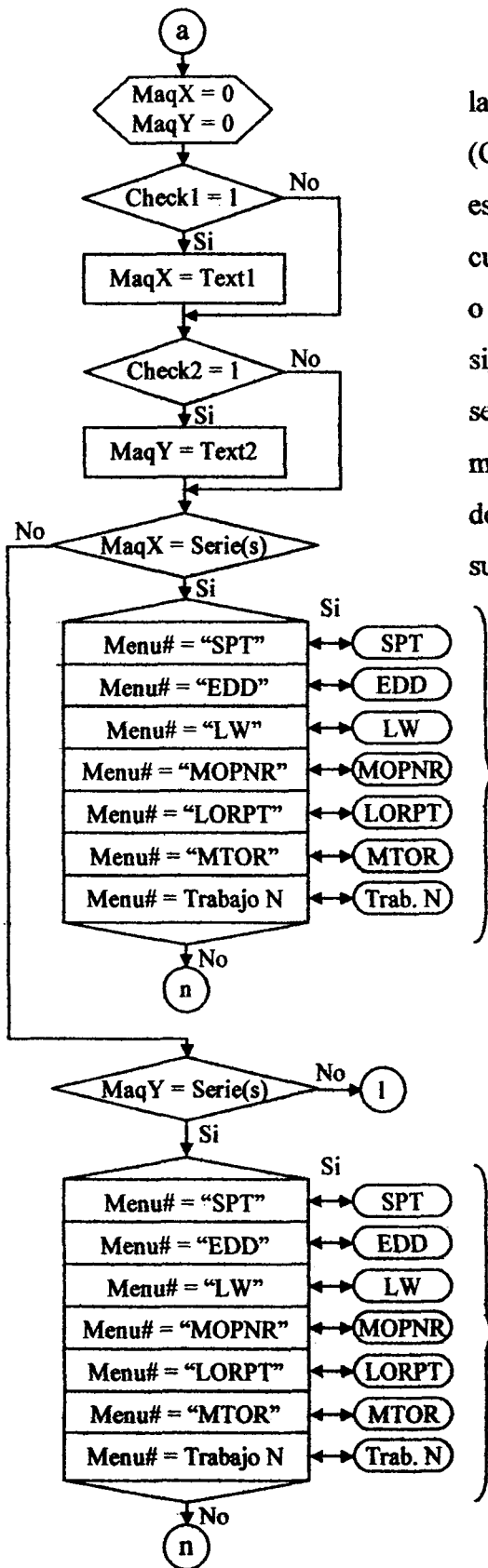
Si ningún trabajo cumplió las condiciones en el ciclo anterior, la máquina que se desocupa primero es incrementada una unidad en el tiempo. Se repiten los ciclos anteriores hasta que un trabajo cumple todas las condiciones anteriores.

-----Tomar la siguiente máquina disponible para la comparación-----



Para seleccionar los trabajos con operaciones a comparar, se comienza tomando la máquina disponible más temprana, si ningún trabajo requiere de sus servicios, se pasa a la siguiente máquina.

-----Selección de reglas de despacho para máquinas especiales.-----



Primero se revisa si están activadas las casillas de Máquina X y Máquina Y (Check1 y Check2 respectivamente). Si están activadas se hace lectura de los cuadros de texto (Text1 y Text2). Si MaqX o MaqY son iguales a la máquina de la siguiente operación a calendarizar, la secuencia de reglas es la de sus propios menús despleables. Se pasa por cada uno de los menús, que indican cuál es la subrutina (Regla) a la que hay que saltar.

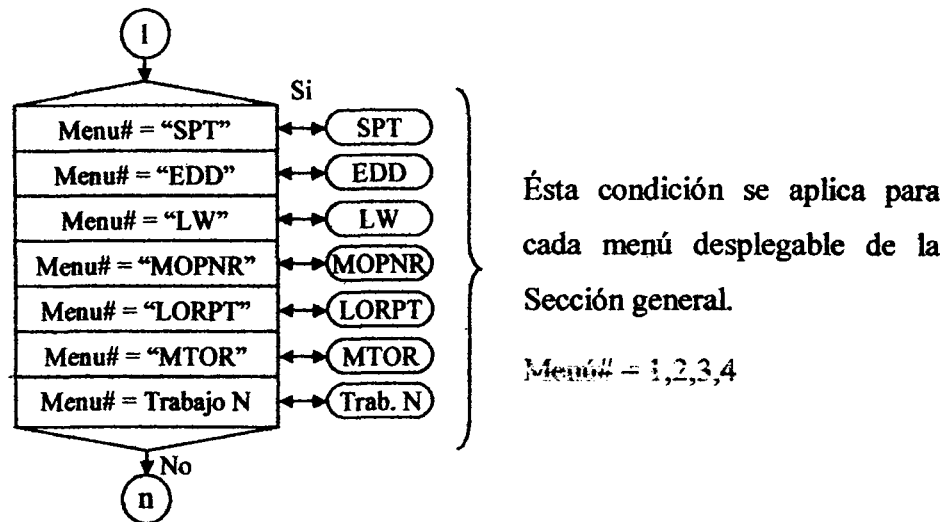
Ésta condición se aplica para cada menú desplegable de la Máquina X en serie.

Menú# = 5,6,7,8

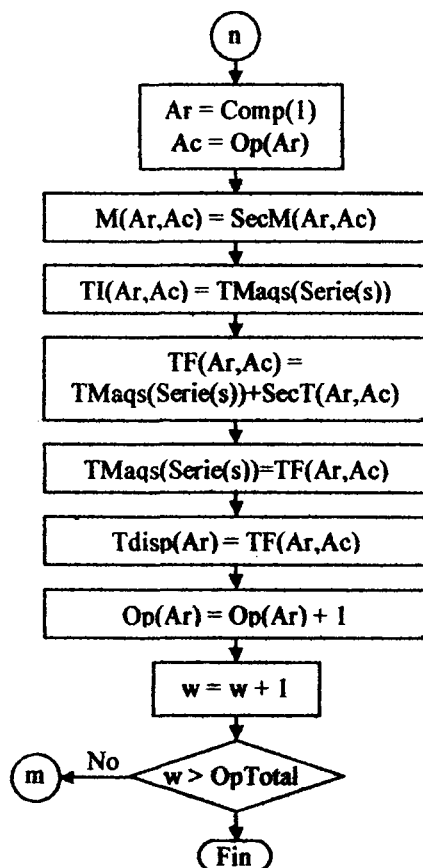
Ésta condición se aplica para cada menú desplegable de la Máquina Y en serie.

Menú# = 9,10,11,12

-----Selección de reglas de despacho generales.-----



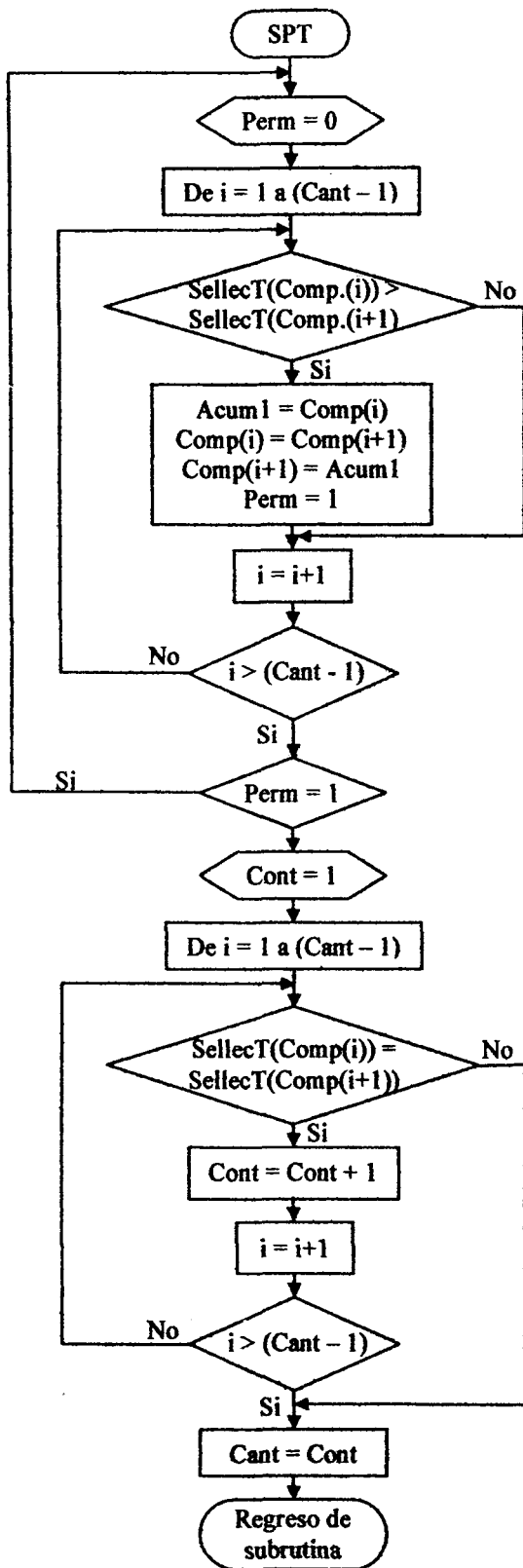
-----Asignar resultados a variables.-----



Ar establece el trabajo seleccionado y Ac la operación calendarizada. Se registra la máquina $M(Ar,Ac)$, el tiempo de inicio de la operación $TI(Ar,Ac)$, el tiempo de finalización $TF(Ar,Ac)$.

Se actualizan los tiempos disponibles para la máquina y el trabajo, además de mover el indicador a la siguiente operación en el mismo trabajo.

Subrutina de SPT

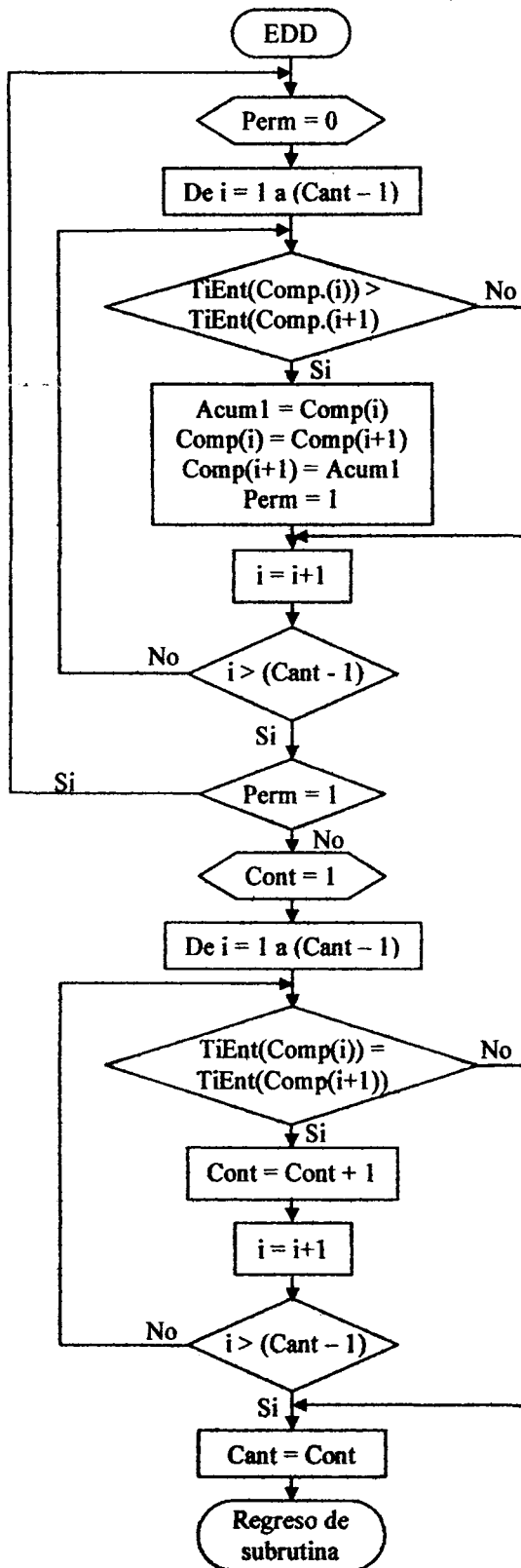


Entrando a la subrutina se inicia un ciclo de permutaciones similar al del ordenamiento de las máquinas en tiempos de menor a mayor, solo que ahora se comparan los tiempos de procesamiento de las operaciones en competencia $Comp(i)$. La cantidad de operaciones empatadas se almacena en la variable $Cant$, de tal forma que si solo es una operación no se realiza el ciclo de permutaciones.

Una vez que estén ordenadas las operaciones ($Comp(1)$, $Comp(2)$, etc.) se realiza un conteo de las operaciones aún empatadas y se almacena en la variable $Cant$.

Al finalizar la subrutina, el apuntador regresa al punto de donde saltó a esta subrutina.

De esta forma se estandarizan los datos de entrada y de salida en cada regla.

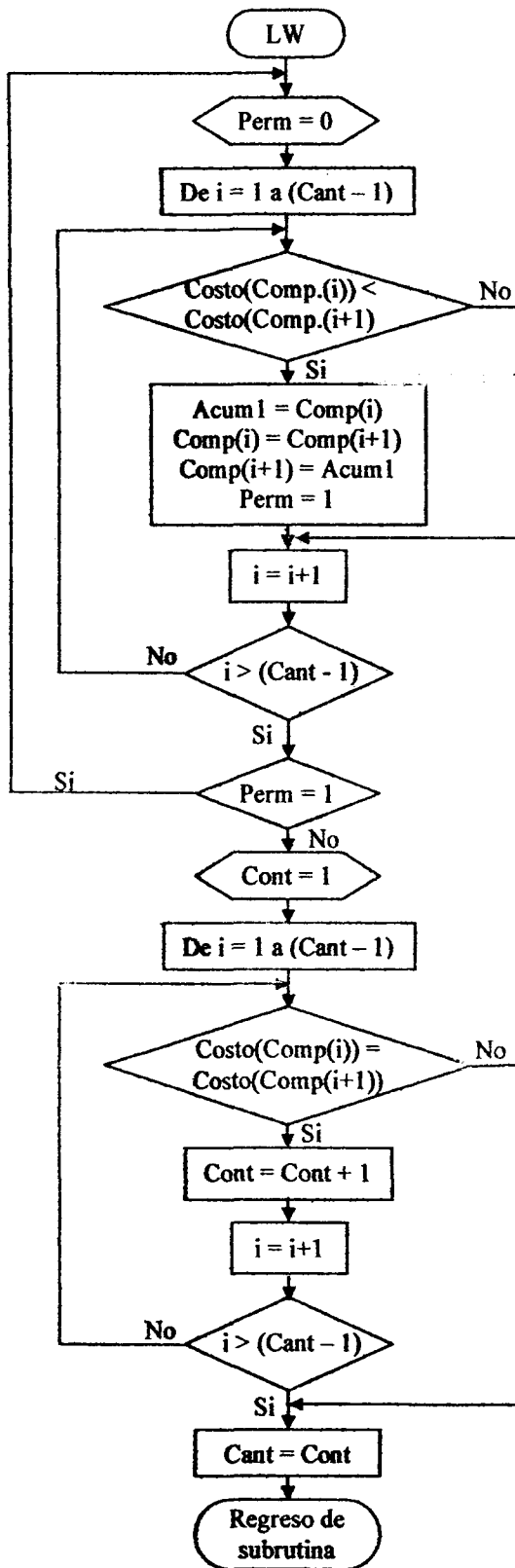


La subrutina es exactamente igual a la de la regla SPT solo que ahora se comparan las fechas de entrega en competencia $Comp(i)$. La cantidad de operaciones empatadas se almacena en la variable $Cant$, de tal forma que si solo es una operación no se realiza el ciclo de permutaciones.

Una vez que estén ordenadas las operaciones ($Comp(1)$, $Comp(2)$, etc.) se realiza un conteo de las operaciones aún empatadas y se almacena en la variable $Cant$.

Al finalizar la subrutina, el apuntador regresa al punto de donde saltó a esta subrutina.

De esta forma se estandarizan los datos de entrada y de salida en cada regla.

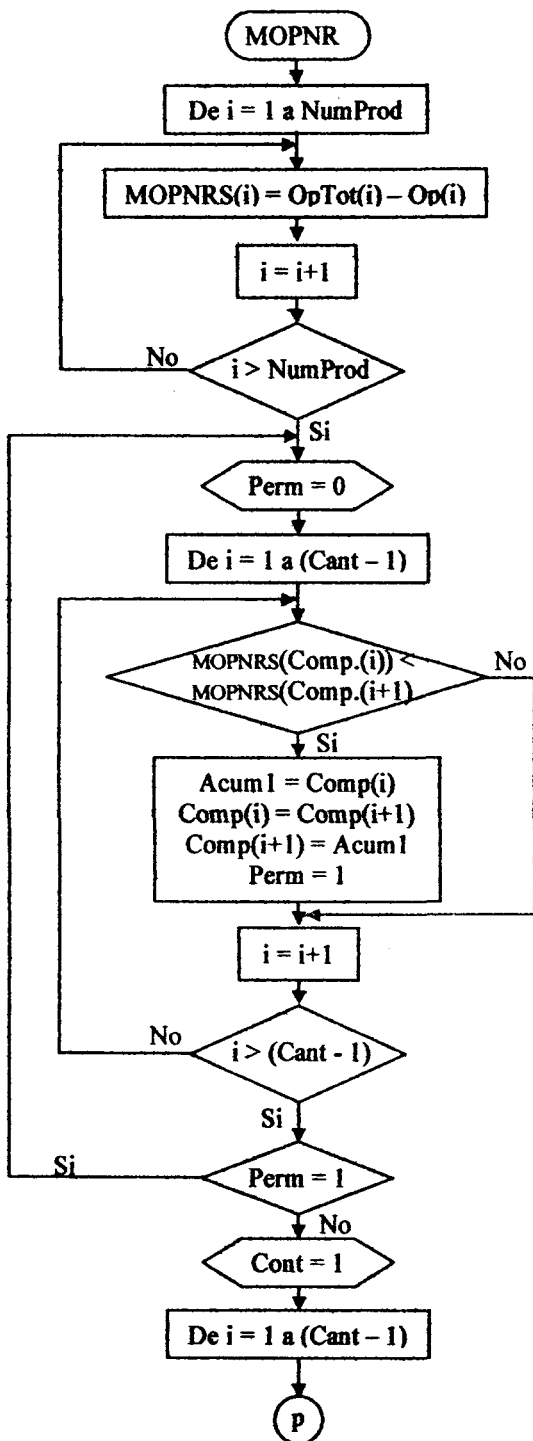


La subrutina es exactamente igual a la de la regla SPT solo que ahora se comparan los costos por entrega tardía $Comp(i)$. La cantidad de operaciones empatadas se almacena en la variable $Cant$, de tal forma que si solo es una operación no se realiza el ciclo de permutaciones.

Una vez que estén ordenadas las operaciones ($Comp(1)$, $Comp(2)$, etc.) se realiza un conteo de las operaciones aún empatadas y se almacena en la variable $Cant$.

Al finalizar la subrutina, el apuntador regresa al punto de donde saltó a esta subrutina.

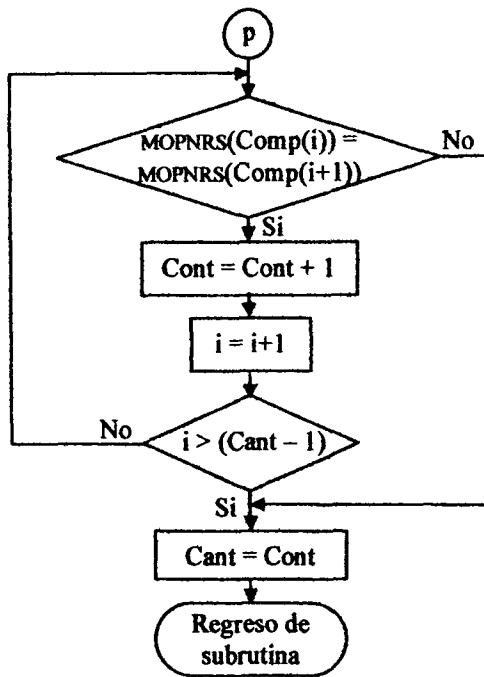
De esta forma se estandarizan los datos de entrada y de salida en cada regla.



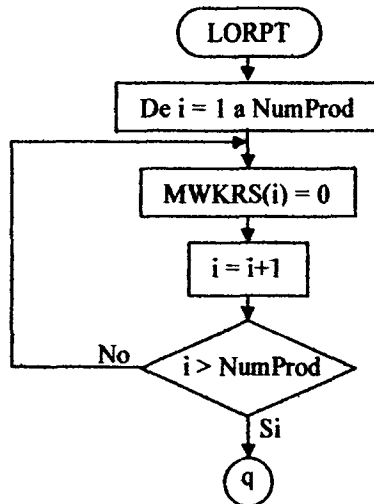
La subrutina inicia calculando el número de operaciones restantes de cada trabajo para, posteriormente, continuar la rutina exactamente igual a la de la regla SPT solo que ahora se comparan los números de operaciones restantes de los trabajos seleccionados $Comp(i)$. La cantidad de operaciones empatadas se almacena en la variable $Cant$, de tal forma que si solo es una operación no se realiza el ciclo de permutaciones.

Una vez que estén ordenadas las operaciones ($Comp(1)$, $Comp(2)$, etc.) se realiza un conteo de las operaciones aún empatadas y se almacena en la variable $Cant$.

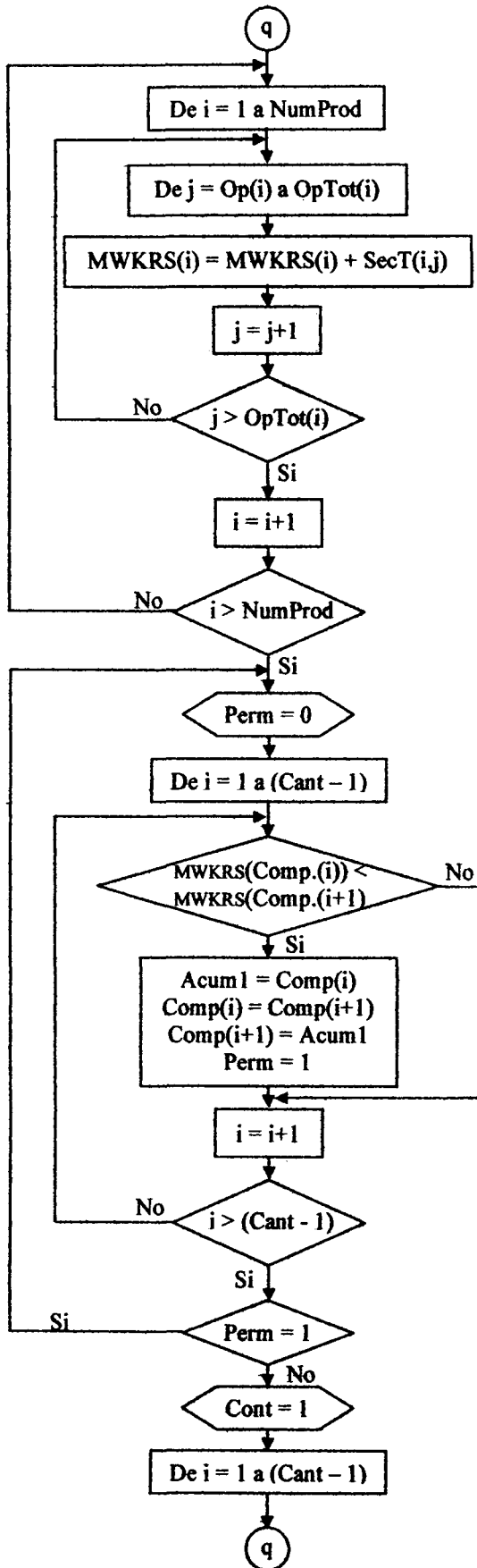
Al finalizar la subrutina, el apuntador regresa al punto de donde saltó a esta subrutina.



-----Subrutina de LORPT-----

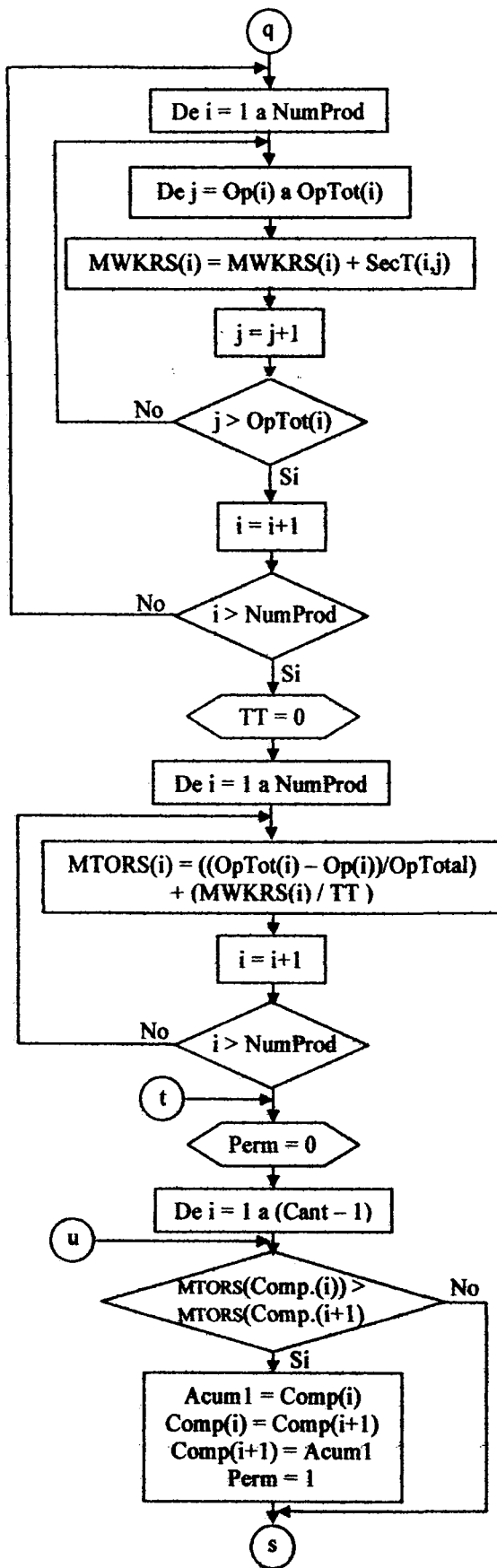


Como primer paso se inicializan las variables $MWKRS(i)$ para poder realizar la suma del tiempo restante de procesamiento para cada trabajo. El resto de la subrutina continua exactamente igual a la de la regla SPT solo que ahora se comparan los tiempos de procesamiento restantes de los trabajos seleccionados $Comp(i)$. La cantidad de operaciones empatadas se almacena en la variable $Cant$, de tal forma que si solo es una operación no se realiza el ciclo de permutaciones.



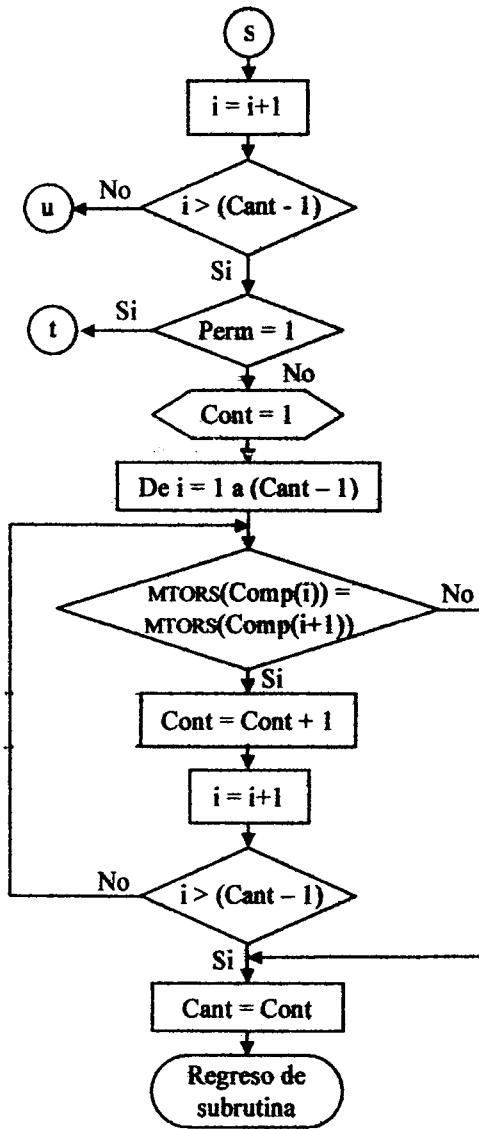
Una vez que estén ordenadas las operaciones (Comp(1), Comp(2), etc.) se realiza un conteo de las operaciones aún empatadas y se almacena en la variable Cant.

Al finalizar la subrutina, el apuntador regresa al punto de donde saltó a esta subrutina.

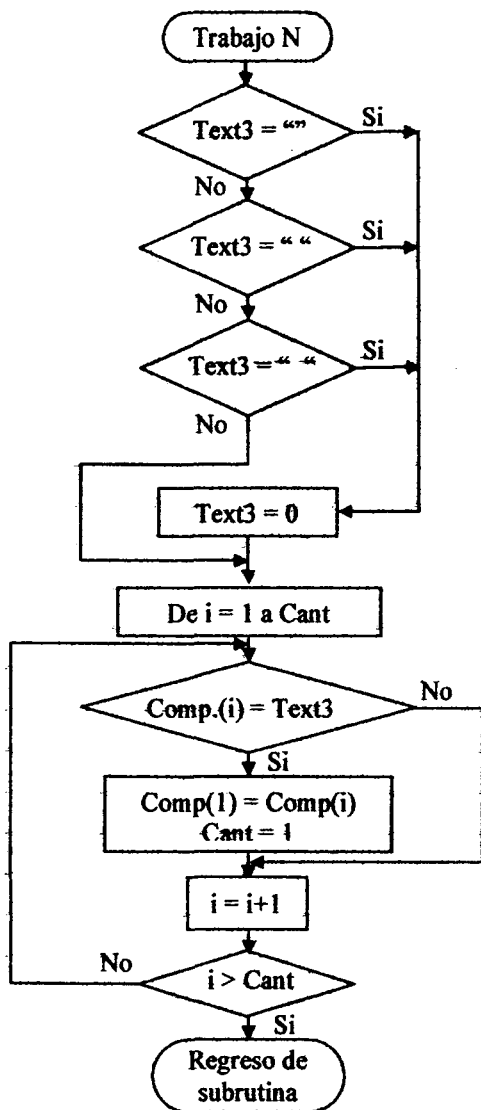


en la variable Cant, de tal forma que si solo es una operación no se realiza el ciclo de permutaciones.

Una vez que estén ordenadas las operaciones (Comp(1), Comp(2), etc.) se realiza un conteo de las operaciones aún empatadas y se almacena en la variable Cant.



 Subrutina de Trabajo N



Como primer paso se asegura que las casillas en blanco sean consideradas como cero.

Después se comparan todos los trabajos en competencia con los cuadros de texto. Si varios trabajos coinciden, entonces se le da preferencia al del primer cuadro que coincide.

Este diagrama se repite para los cinco cuadros de texto: Text3, Text4, Text5, Text6 y Text7.

A3. CÓDIGO DE PROGRAMACIÓN

```

*****Hacer una copia de los datos
For i = 1 To NumProd
  For j = 1 To OpTot(i)
    SecMs(i, j) = SecM(i, j)
    SecTs(i, j) = SecT(i, j)
  Next j
Next i
For i = 1 To NumMaq
  TMaqs(i) = TMaq(i)
Next i

*****Calcular operaciones totales
OpTotal = 0
For i = 1 To NumProd
  OpTotal = OpTotal + OpTot(i)
Next i

*****Inicio del tiempo disponible de cada trabajo
For i = 1 To NumProd
  TDisp(i) = 0
Next i

*****Iniciar con la primera operación de cada trabajo
For i = 1 To NumProd
  Op(i) = 1
Next i
'-----Comienza ciclo
For w = 1 To OpTotal

*****Seleccionar la primera columna de operaciones
For i = 1 To NumProd
  SelecM(i) = SecMs(i, Op(i))
  SellecT(i) = SecTs(i, Op(i))
Next i

*****Inicializar secuencia de máquinas
siguiente2:
For i = 1 To NumMaq
  Serie(i) = i
Next i

*****Ordenar máquinas en tiempos de menor a mayor
inicio:
Perm = 0
For i = 1 To (NumMaq - 1)
  If TMaqs(Serie(i)) > TMaqs(Serie(i + 1)) Then
    Acum1 = Serie(i)
    Serie(i) = Serie(i + 1)
    Serie(i + 1) = Acum1
    Perm = 1
  End If
Next i
If Perm = 1 Then
  GoTo inicio
End If

*****Seleccionar los trabajos con operaciones a comparar
k = 0
s = 1
siguiente:
For i = 1 To NumProd

```

```

If SelecM(i) = Serie(s) And TMaqs(Serie(s)) >= TDisp(i) And Op(i) < (OpTot(i) + 1) Then
    k = k + 1
    Comp(k) = i
End If
Next i

```

*****Incrementar el tiempo en las máquinas

```

If (k = 0) And (s = NumMaq) Then
    TMaqs(Serie(1)) = TMaqs(Serie(1)) + 1
    s = 1
    GoTo siguiente2
End If

```

*****Tomar la siguiente máquina disponible para la comparación

```

If k = 0 Then
    s = s + 1
    GoTo siguiente
End If
Cant = k
k = 0

```

*****Selección de reglas de despacho

```

MaqX = 0
MaqY = 0
If Check1.Value = 1 Then
    MaqX = Text1.Text
End If
If Check2.Value = 1 Then
    MaqY = Text2.Text
End If

```

```

If MaqX = Serie(s) Then
    Select Case Combo5.Text
    Case "SPT": GoSub SPT
    Case "EDD": GoSub EDD
    Case "LW": GoSub LW
    Case "MOPNR": GoSub MOPNR
    Case "MWKR": GoSub MWKR
    Case "MTOR": GoSub MTOR
    Case "Trabajo N": GoSub TrabajoN
    End Select

```

```

Select Case Combo6.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo7.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo8.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD

```

```

Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select
GoTo Fin
End If

```

```

If MaqY = Serie(s) Then
Select Case Combo9.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo10.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo11.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo12.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select
GoTo Fin
End If

```

```

Select Case Combo1.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select
Select Case Combo2.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW

```

```

Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo3.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select

```

```

Select Case Combo4.Text
Case "SPT": GoSub SPT
Case "EDD": GoSub EDD
Case "LW": GoSub LW
Case "MOPNR": GoSub MOPNR
Case "MWKR": GoSub MWKR
Case "MTOR": GoSub MTOR
Case "Trabajo N": GoSub TrabajoN
End Select
Fin:

```

*****Asignar resultados a variables

```

Ar = Comp(1)   Número de trabajo seleccionado
Ac = Op(Ar)    Número de operación seleccionada en el trabajo

```

```

M(Ar, Ac) = SecM(Ar, Ac)
Tl(Ar, Ac) = TMaqs(Serie(s))
TF(Ar, Ac) = TMaqs(Serie(s)) + SecT(Ar, Ac)
TMaqs(Serie(s)) = TF(Ar, Ac)
TDisp(Ar) = TF(Ar, Ac)
TiMaq(Serie(s)) = TMaqs(Serie(s))
TF=" & TF(Ar, Ac) & " "
Ac) & " "
Op(Ar) = Op(Ar) + 1
r(w) = Ar
α(w) = Ac
Next w
If NumMaq <> 0 Then
  Resumenes.Show
Else
  MsgBox "Introducir los datos correctamente." & Chr(10) & "Es probable que no ha asignado operaciones.", vbCritical +
vbDefaultButton1, "Error"
End If
Exit Sub

```

*****Shortest Processing Time

```

SPT:
ini:
Perm = 0
For i = 1 To (Cant - 1)
  If SellecT(Comp(i)) > SellecT(Comp(i + 1)) Then
    Acum1 = Comp(i)
    Comp(i) = Comp(i + 1)
    Comp(i + 1) = Acum1
    Perm = 1
  End If
Next i

If Perm = 1 Then

```

```

    GoTo ini
End If
Cont = 1
For i = 1 To (Cant - 1)
    If Sellect(Comp(i)) = Sellect(Comp(i + 1)) Then
        Cont = Cont + 1
    Else
        GoTo Salir
    End If
Next i
Salir:
Cant = Cont
Return

```

*****Earliest Due Date

```

EDD:
ini1:
Perm = 0
For i = 1 To (Cant - 1)
    If TiEnt(Comp(i)) > TiEnt(Comp(i + 1)) Then
        Acum1 = Comp(i)
        Comp(i) = Comp(i + 1)
        Comp(i + 1) = Acum1
        Perm = 1
    End If
Next i
If Perm = 1 Then
    GoTo ini1
End If
Cont = 1
For i = 1 To (Cant - 1)
    If TiEnt(Comp(i)) = TiEnt(Comp(i + 1)) Then
        Cont = Cont + 1
    Else
        GoTo Salir1
    End If
Next i
Salir1:
Cant = Cont
Return

```

*****Largest Weight

```

LW:
ini2:
Perm = 0
For i = 1 To (Cant - 1)
    If Costo(Comp(i)) < Costo(Comp(i + 1)) Then
        Acum1 = Comp(i)
        Comp(i) = Comp(i + 1)
        Comp(i + 1) = Acum1
        Perm = 1
    End If
Next i
If Perm = 1 Then
    GoTo ini2
End If
Cont = 1
For i = 1 To (Cant - 1)
    If Costo(Comp(i)) = Costo(Comp(i + 1)) Then
        Cont = Cont + 1
    Else
        GoTo Salir2
    End If
Next i
Salir2:

```

```
Cant = Cont
Return
```

```
*****Most Operations Remaining
```

```
MOPNR:
```

```
For i = 1 To NumProd
  MOPNRS(i) = OpTot(i) - Op(i)
Next i
```

```
ini3:
```

```
Perm = 0
For i = 1 To (Cant - 1)
  If MOPNRS(Comp(i)) < MOPNRS(Comp(i + 1)) Then
    Acum1 = Comp(i)
    Comp(i) = Comp(i + 1)
    Comp(i + 1) = Acum1
    Perm = 1
  End If
Next i
```

```
If Perm = 1 Then
  GoTo ini3
```

```
End If
```

```
Cont = 1
```

```
For i = 1 To (Cant - 1)
  If MOPNRS(Comp(i)) = MOPNRS(Comp(i + 1)) Then
    Cont = Cont + 1
  Else
    GoTo Salir3
  End If
Next i
```

```
Salir3:
```

```
Cant = Cont
```

```
Return
```

```
*****Longest Operations Remaining Processing Time
```

```
LORPT:
```

```
For i = 1 To NumProd
  MWKRS(i) = 0
Next i
For i = 1 To NumProd
  For j = Op(i) To OpTot(i)
    MWKRS(i) = MWKRS(i) + SecT(i, j)
  Next j
Next i
```

```
ini4:
```

```
Perm = 0
For i = 1 To (Cant - 1)
  If MWKRS(Comp(i)) < MWKRS(Comp(i + 1)) Then
    Acum1 = Comp(i)
    Comp(i) = Comp(i + 1)
    Comp(i + 1) = Acum1
    Perm = 1
  End If
Next i
```

```
If Perm = 1 Then
  GoTo ini4
```

```
End If
```

```
Cont = 1
```

```
For i = 1 To (Cant - 1)
  If MWKRS(Comp(i)) = MWKRS(Comp(i + 1)) Then
    Cont = Cont + 1
  Else
    GoTo Salir4
  End If
Next i
```

```

Next i
Salir4:
Cant = Cont
Return

*****Most Time and Operations Remaining
MTOR:
For i = 1 To NumProd
  MWKRS(i) = 0
Next i
For i = 1 To NumProd
  For j = Op(i) To OpTot(i)
    MWKRS(i) = MWKRS(i) + SecT(i, j)
  Next j
Next i
TT = 0
For i = 1 To NumProd
  For j = 1 To OpTot(i)
    TT = TT + SecT(i, j)
  Next j
Next i
For i = 1 To NumProd
  MTORS(i) = ((OpTot(i) - Op(i)) / OpTotal) + (MWKRS(i) / TT)
Next i

ini5:
Perm = 0
For i = 1 To (Cant - 1)
  If MTORS(Comp(i)) < MTORS(Comp(i + 1)) Then
    Acum1 = Comp(i)
    Comp(i) = Comp(i + 1)
    Comp(i + 1) = Acum1
    Perm = 1
  End If
Next i
If Perm = 1 Then
  GoTo ini5
End If
Cont = 1
For i = 1 To (Cant - 1)
  If MTORS(Comp(i)) = MTORS(Comp(i + 1)) Then
    Cont = Cont + 1
  Else
    GoTo Salir
  End If
Next i
Salir5:
Cant = Cont
Return

*****Trabajo N
TrabajoN:

If Text3.Text = "" Or Text3.Text = " " Or Text3.Text = " " Then
  Text3.Text = "0"
End If
If Text4.Text = "" Or Text4.Text = " " Or Text4.Text = " " Then
  Text4.Text = "0"
End If
If Text5.Text = "" Or Text5.Text = " " Or Text5.Text = " " Then
  Text5.Text = "0"
End If
If Text6.Text = "" Or Text6.Text = " " Or Text6.Text = " " Then
  Text6.Text = "0"
End If

```



```
If Text7.Text = "" Or Text7.Text = " " Or Text7.Text = " " Then
    Text7.Text = "0"
End If

For i = 1 To Cant
    If Comp(i) = Text3.Text Then
        Comp(1) = Comp(i)
        Cant = 1
        GoTo Salir6
    End If
Next i
For i = 1 To Cant
    If Comp(i) = Text4.Text Then
        Comp(1) = Comp(i)
        Cant = 1
        GoTo Salir6
    End if
Next i
For i = 1 To Cant
    If Comp(i) = Text5.Text Then
        Comp(1) = Comp(i)
        Cant = 1
        GoTo Salir6
    End If
Next i
For i = 1 To Cant
    If Comp(i) = Text6.Text Then
        Comp(1) = Comp(i)
        Cant = 1
        GoTo Salir6
    End If
Next i
For i = 1 To Cant
    If Comp(i) = Text7.Text Then
        Comp(1) = Comp(i)
        Cant = 1
        GoTo Salir6
    End If
Next i
If Text3.Text = 0 Then
    Text3.Text = ""
End If
If Text4.Text = 0 Then
    Text4.Text = ""
End If
If Text5.Text = 0 Then
    Text5.Text = ""
End If
If Text6.Text = 0 Then
    Text6.Text = ""
End If
If Text7.Text = 0 Then
    Text7.Text = ""
End If

Salir6:
Return
```

ANEXO B

Tabla comparativa de frecuencia de menor valor de Cmax entre Reglas y Secuencias de Reglas de Despacho

	1M 2T	1M 3T	1M 4T	1M 5T	2M 2T	2M 3T	2M 4T	2M 5T	3M 2T	3M 3T	3M 4T	3M 5T	4M 2T	4M 3T	4M 4T	4M 5T	5M 2T	5M 3T	5M 4T	5M 5T
SPT	100	100	100	100	100	80	80	40	80	70	70	80	80	80	40	80	80	80	80	30
EDD	100	100	100	100	90	40	70	40	70	70	40	10	70	50	40	40	80	70	40	0
LW	100	100	100	100	90	60	50	30	80	40	30	30	90	60	60	50	80	50	20	20
MOPNR	100	100	100	100	100	70	80	90	80	70	60	50	80	100	80	40	70	90	60	50
LORPT	100	100	100	100	80	90	60	50	90	80	50	50	80	100	70	50	90	100	60	40
MTOR	100	100	100	100	80	90	80	60	90	80	80	60	80	100	60	60	90	100	90	70

D. Std. Rel.	Media Rel.
20.988	64.375
24.732	51.250
23.523	52.500
17.783	73.125
19.958	71.250
13.889	79.375

SPT	100	100	100	100	100	20	80	40	80	60	60	50	80	60	60	40	90	50	50	40
EDD	100	100	100	100	90	20	70	40	70	60	40	10	60	50	40	40	80	70	40	30
LW	100	100	100	100	90	30	40	30	80	40	20	30	80	50	60	50	80	50	20	30
MOPNR	100	100	100	100	100	50	70	90	80	60	50	50	80	80	80	40	70	90	60	40
LORPT	100	100	100	100	80	60	60	50	90	70	50	40	80	80	70	50	90	100	60	60
MTOR	100	100	100	100	80	60	70	60	90	70	80	40	80	80	60	50	90	100	90	80
G1A	100	100	100	100	100	30	80	40	80	60	70	50	80	70	60	50	90	60	50	60
G2A	100	100	100	100	100	40	80	40	80	60	70	50	80	70	60	50	90	60	50	60
G3A	100	100	100	100	100	50	80	90	80	60	70	80	100	70	70	50	90	70	60	70
G4A	100	100	100	100	80	60	60	50	80	70	60	40	90	90	70	60	90	100	60	60
G5A	100	100	100	100	80	60	60	80	80	70	70	60	90	90	70	60	90	100	90	80
G6A	100	100	100	100	80	60	50	50	80	70	60	40	90	90	70	60	90	100	60	60
CB1	100	100	100	100	100	50	80	70	90	60	60	70	80	70	60	40	100	70	70	30
CB2	100	100	100	100	90	50	80	70	90	60	60	70	80	70	60	40	100	70	70	30
CB3	100	100	100	100	90	50	80	80	90	70	60	70	100	70	70	40	100	80	70	50
CB4	100	100	100	100	80	60	60	70	80	70	80	40	100	90	70	70	90	100	70	70
CB5	100	100	100	100	80	60	60	70	80	70	80	40	100	90	70	60	90	100	90	80
CB6	100	100	100	100	80	60	50	70	80	70	80	40	100	90	70	70	90	100	70	70
G1B	100	100	100	100	80	100	70	50	90	80	80	40	70	90	60	50	90	90	100	70
G2B	100	100	100	100	80	100	70	50	90	80	80	40	70	90	60	50	90	90	100	70
G3B	100	100	100	100	80	100	70	50	90	80	80	40	70	90	60	50	90	90	100	70
G4B	100	100	100	100	80	100	70	50	90	80	80	40	70	90	60	50	90	90	100	70
G5B	100	100	100	100	80	100	80	60	90	80	70	60	70	90	70	50	90	90	90	80
G6B	100	100	100	100	80	100	60	30	90	80	70	40	70	90	70	50	90	90	70	60

21.29163	73.84615
22.05108	62.30769
23.05790	60.00000
18.69715	83.84615
17.21191	83.84615
16.27882	90.76923
18.60779	79.23077
17.51190	80.00000
15.47848	91.53846
16.73320	86.15385
13.02242	94.61538
17.30848	85.38462
19.27866	84.61538
18.33712	83.84615
17.40450	90.00000
15.49193	92.30769
16.27882	93.84615
16.31717	91.53846
18.24600	93.07692
18.24600	93.07692
18.24600	93.07692
18.24600	93.07692
13.76893	96.15385
19.62142	87.69231

ANEXO C

C.1 PROBLEMAS DE CALENDARIZACIÓN

2M 2T 1

Maquinas	Tiempo	E	C
1,2	3,3	2	3
2,1	1,3	0	1
M1=0; M2=0			

2M 2T 2

Maquinas	Tiempo	E	C
2,1	4,1	1	2
1,2	2,2	1	2
M1=0; M2=0			

2M 2T 3

Maquinas	Tiempo	E	C
1,2	1,3	4	3
1,2	5,2	3	1
M1=0; M2=0			

2M 2T 4

Maquinas	Tiempo	E	C
2,1	6,5	2	0
1,2	3,3	2	1
M1=0; M2=0			

2M 2T 5

Maquinas	Tiempo	E	C
2,1	3,1	1	3
2,1	3,1	3	4
M1=0; M2=0			

2M 2T 6

Maquinas	Tiempo	E	C
1,2	3,7	7	0
2,1	3,1	3	1
M1=0; M2=0			

2M 2T 7

Maquinas	Tiempo	E	C
1,2	1,1	2	3
2,1	1,4	4	3
M1=0; M2=0			

2M 2T 8

Maquinas	Tiempo	E	C
2,1	2,4	2	1
1,2	4,6	1	0
M1=0; M2=0			

2M 2T 9

Maquinas	Tiempo	E	C
1,2	5,7	0	3
1,2	6,9	0	4
M1=0; M2=0			

2M 2T 10

Maquinas	Tiempo	E	C
1,2	6,8	7	8
2,1	2,3	1	9
M1=0; M2=0			

2M 3T 1

Maquinas	Tiempo	E	C
1,2	3,1	2	8
1,2	2,3	2	9
2,1	1,2	1	0
M1=2; M2=0			

2M 3T 2

Maquinas	Tiempo	E	C
2,1	2,2	3	4
2,1	2,7	1	7
1,2	3,3	0	3
M1=0; M2=0			

2M 3T 3

Maquinas	Tiempo	E	C
1,2	4,4	4	5
1	1	3	5
1,2	1,9	6	1
M1=0; M2=0			

2M 3T 4

Maquinas	Tiempo	E	C
1,2	3,6	9	0
2,1	2,2	2	0
1,2	2,1	0	2
M1=1; M2=2			

2M 3T 5

Maquinas	Tiempo	E	C
2,1	5,3	6	1
1,2	7,3	3	1
1	1	1	6
M1=0; M2=3			

2M 3T 6

Maquinas	Tiempo	E	C
1,2	3,6	0	4
1,2	2,7	0	7
1,2	3,1	1	1
M1=0; M2=3			

2M 3T 7

Maquinas	Tiempo	E	C
1,2	1,4	3	6
2,1	2,2	2	6
2,1	7,6	4	2
M1=1; M2=0			

2M 3T 8

Maquinas	Tiempo	E	C
2,1	4,2	1	4
2,1	3,2	1	7
2,1	6,1	2	9
M1=0; M2=0			

2M 3T 9

Maquinas	Tiempo	E	C
2,1	5,1	0	8
1,2	6,9	0	9
1,2	9,6	3	1
M1=0; M2=0			

2M 3T 10

Maquinas	Tiempo	E	C
1,2	1,3	3	0
2,1	1,2	4	2
1,2	3,1	1	0
M1=3; M2=2			

2M 4T 1

Maquinas	Tiempo	E	C
1,2	3,4	4	7
1,2	4,6	3	0
1,2	2,7	4	0
1,2	1,1	1	8
M1=0; M2=0			

2M 4T 2

Maquinas	Tiempo	E	C
1,2	4,3	0	2
2,1	6,3	0	3
1,2	7,3	2	7
2,1	2,2	1	1
M1=0; M2=0			

2M 4T 3

Maquinas	Tiempo	E	C
2,1	9,1	1	6
2,1	1,7	2	2
1,2	8,8	3	4
2,1	2,1	4	0
M1=1; M2=0			

2M 4T 4

Maquinas	Tiempo	E	C
1,2	3,4	0	0
2,1	3,1	0	3
2	2	3	3
1,2	2,3	1	0
M1=3; M2=2			

2M 4T 5

Maquinas	Tiempo	E	C
1,2	2,2	3	8
1,2	1,2	2	1
1	7	1	0
2	6	4	9
M1=0; M2=2			

2M 4T 6

Maquinas	Tiempo	E	C
1,2	6,4	0	3
1,2	7,2	0	4
1,2	1,4	3	7
2,1	4,2	3	1
M1=0; M2=0			

2M 4T 7

Maquinas	Tiempo	E	C
2,1	3,1	0	3
2,1	2,1	9	0
2,1	1,3	1	0
1,2	6,1	6	1
M1=1; M2=3			

2M 4T 8

Maquinas	Tiempo	E	C
1,2	7,6	7	1
2	8	6	3
1,2	1,8	5	0
2,1	2,1	0	0
M1=2; M2=0			

2M 4T 9

Maquinas	Tiempo	E	C
1	2	5	0
2,1	4,3	7	3
2,1	6,3	0	1
1,2	8,6	1	0
M1=0; M2=0			

2M 4T 10

Maquinas	Tiempo	E	C
2,1	3,1	4	1
1,2	6,2	7	7
1,2	7,3	9	3
2,1	2,7	1	9
M1=3; M2=4			

2M 5T 1

Maquinas	Tiempo	E	C
1,2	3,2	0	7
1,2	4,3	0	8
2,1	6,1	1	3
1,2	1,7	3	2
1,2	3,3	4	1
M1=0; M2=0			

2M 5T 2

Maquinas	Tiempo	E	C
1,2	6,4	5	6
2,1	1,2	0	1
1,2	4,6	0	4
2,1	3,6	3	3
1,2	1,3	0	1
M1=1; M2=0			

2M 5T 3

Maquinas	Tiempo	E	C
2,1	9,5	9	3
2,1	2,5	2	0
1,2	1,6	7	0
1,2	7,1	3	3
2,1	5,4	4	1
M1=0; M2=0			

2M 5T 4

Maquinas	Tiempo	E	C
1,2	3,4	0	0
2,1	2,3	3	0
1,2	3,3	4	0
2,1	1,2	1	3
1,2	4,1	2	1
M1=2; M2=1			

2M 5T 5

Maquinas	Tiempo	E	C
2,1	3,6	0	7
1,2	1,1	0	2
1,2	7,4	3	4
1	1	1	5
2,1	2,3	4	1
M1=0; M2=0			

2M 5T 6

Maquinas	Tiempo	E	C
1,2	3,4	6	9
1,2	3,2	6	0
1,2	2,2	1	9
1,2	2,3	7	3
2,1	4,6	8	4
M1=0; M2=0			

2M 5T 7

Maquinas	Tiempo	E	C
2,1	6,4	0	0
2	4	1	3
2,1	6,4	2	0
1	1	0	3
1,2	2,1	2	0
M1=1; M2=0			

2M 5T 8

Maquinas	Tiempo	E	C
1,2	2,3	3	0
1,2	2,3	4	0
1,2	2,1	0	1
1,2	3,6	1	0
1,2	1,3	0	3
M1=0; M2=0			

2M 5T 9

Maquinas	Tiempo	E	C
1,2	6,1	0	3
1	7	0	4
2,1	2,2	2	7
2,1	1,2	3	1
1,2	5,1	0	2
M1=3; M2=2			

2M 5T 10

Maquinas	Tiempo	E	C
2	4	1	0
2,1	3,4	2	3
1,2	6,6	4	0
1,2	1,6	7	9
1,2	9,1	0	2
M1=0; M2=0			

3M 2T 1

Maquinas	Tiempo	E	C
1,2,3	4,4,6	3	0
2,3,1	2,2,1	4	9
M1=0; M2=0; M3=0			

3M 2T 2

Maquinas	Tiempo	E	C
1,2,3	3,2,7	1	2
1,2,3	1,2,8	0	3
M1=0; M2=1; M3=0			

3M 2T 3

Maquinas	Tiempo	E	C
2,1,3	1,1,4	9	4
3,2	1,9	6	1
M1=3; M2=0; M3=2			

3M 2T 4

Maquinas	Tiempo	E	C
3,1	6,6	7	6
1,3,2	5,3,6	2	6
M1=0; M2=0; M3=0			

3M 2T 5

Maquinas	Tiempo	E	C
3,2,1	7,4,5	4	2
3,1,2	2,1,9	1	3
M1=1; M2=1; M3=2			

3M 2T 6

Maquinas	Tiempo	E	C
1,3,2	4,2,3	0	4
3	3	0	5
M1=3; M2=2; M3=3			

3M 2T 7

Maquinas	Tiempo	E	C
2,1,3	2,3,6	3	0
2,3,1	1,6,9	4	0
M1=0; M2=0; M3=0			

3M 2T 8

Maquinas	Tiempo	E	C
3,2,1	3,7,5	9	2
1,3	3,8	1	1
M1=1; M2=0; M3=3			

3M 2T 9

Maquinas	Tiempo	E	C
1,2,3	4,6,2	6	4
1,3,2	2,5,1	3	1
M1=0; M2=0; M3=0			

3M 2T 10

Maquinas	Tiempo	E	C
3,1,2	6,3,4	3	6
1,3,2	4,6,3	7	7
M1=1; M2=2; M3=3			

3M 3T 1

Maquinas	Tiempo	E	C
1,3,2	3,4,1	0	0
2,1,3	6,5,3	3	1
3	9	1	1
M1=0; M2=0; M3=1			

3M 3T 2

Maquinas	Tiempo	E	C
1,2,3	4,4,7	4	2
1,2,3	2,6,1	0	4
2,3,1	1,7,4	0	7
M1=0; M2=0; M3=0			

3M 3T 3

Maquinas	Tiempo	E	C
3,2,1	3,1,6	1	0
2,1,3	3,4,3	2	3
3,1	3,7	3	6
M1=0; M2=0; M3=0			

3M 3T 4

Maquinas	Tiempo	E	C
1,2,3	2,3,7	6	9
1,3,2	1,4,1	7	1
1,3,2	6,2,9	4	2
M1=1; M2=4; M3=1			

3M 3T 5

Maquinas	Tiempo	E	C
2,3,1	6,3,4	4	0
2,1,3	2,6,1	1	3
1,3,2	1,5,9	2	3
M1=1; M2=0; M3=1			

3M 3T 6

Maquinas	Tiempo	E	C
3,2,1	4,5,8	0	3
2,1,3	7,5,2	0	1
1,3,2	2,4,6	3	6
M1=0; M2=0; M3=3			

3M 3T 7

Maquinas	Tiempo	E	C
2,1,3	2,1,4	1	7
1	2	2	1
2,1,3	2,4,7	2	4
M1=4; M2=3; M3=0			

3M 3T 8

Maquinas	Tiempo	E	C
1,2,3	3,1,9	3	1
2,3,1	2,1,7	6	9
1,3,2	3,1,2	9	3
M1=0; M2=0; M3=0			

3M 3T 9

Maquinas	Tiempo	E	C
3,2,1	1,4,4	2	8
3,1,2	1,6,6	4	5
2,3,1	2,2,1	3	4
M1=0; M2=0; M3=0			

3M 3T 10

Maquinas	Tiempo	E	C
1,2,3	4,1,8	0	6
3,2,1	4,2,6	0	6
2,1,3	4,3,4	2	4
M1=1; M2=3; M3=2			

3M 4T 1

Maquinas	Tiempo	E	C
1,3,2	3,6,8	0	2
2,3,1	3,4,7	4	2
3,2,1	2,1,2	3	1
1,2,3	2,7,4	1	5
M1=0; M2=0; M3=0			

3M 4T 2

Maquinas	Tiempo	E	C
2,1,3	2,3,6	9	0
2	1	2	3
1,3	4,2	0	7
2,3,1	3,9,1	0	0
M1=1; M2=2; M3=1			

3M 4T 3

Maquinas	Tiempo	E	C
1,2,3	3,1,4	4	0
1,3,2	3,1,3	3	3
1,2,3	2,2,2	2	3
1,2,3	3,4,6	6	0
M1=0; M2=3; M3=0			

3M 4T 4

Maquinas	Tiempo	E	C
3,2,1	1,3,7	1	0
3,2	9,4	2	9
2,1,3	1,1,8	4	2
3,2,1	4,2,2	9	6
M1=0; M2=0; M3=0			

3M 4T 5

Maquinas	Tiempo	E	C
1,2,3	6,6,9	5	0
2,3,1	3,6,1	0	3
3,1,2	2,5,6	3	0
3,1,2	6,5,3	0	2
M1=3; M2=2; M3=2			

3M 4T 6

Maquinas	Tiempo	E	C
1,2,3	3,4,3	0	0
2,1,3	3,6,4	3	3
1,2,3	3,2,1	7	0
1,3	3,5	4	5
M1=0; M2=0; M3=1			

3M 4T 7

Maquinas	Tiempo	E	C
1,2,3	2,5,4	3	5
3,1,2	1,5,5	9	4
3,1,2	3,1,1	1	7
2,3,1	2,4,4	8	0
M1=0; M2=0; M3=0			

3M 4T 8

Maquinas	Tiempo	E	C
2,1	6,4	2	0
2,3,1	4,6,3	2	2
3,2,1	3,2,4	3	0
1,2,3	1,7,5	1	0
M1=0; M2=0; M3=0			

3M 4T 9

Maquinas	Tiempo	E	C
2,1,3	3,7,1	0	9
2,1,3	6,8,3	0	8
2,1,3	7,2,2	0	8
2,1,3	2,1,6	1	3
M1=2; M2=2; M3=0			

3M 4T 10

Maquinas	Tiempo	E	C
3,2,1	9,3,9	2	2
3,1,2	1,6,1	4	0
2,3,1	3,9,2	6	0
2,1,3	3,6,5	8	7
M1=1; M2=0; M3=3			

3M 5T 1

Maquinas	Tiempo	E	C
1,3,2	3,4,3	4	2
3,1,2	6,6,5	7	2
2,1,3	4,7,4	2	0
1,2,3	1,2,2	3	3
1,3,2	9,2,5	1	5
M1=0; M2=0; M3=0			

3M 5T 2

Maquinas	Tiempo	E	C
2,1,3	4,6,2	6	4
2,3,1	4,4,1	0	2
1,3,2	2,7,6	0	3
3,1,2	1,3,7	3	4
2,1,3	3,2,3	1	5
M1=0; M2=0; M3=0			

3M 5T 3

Maquinas	Tiempo	E	C
1,2,3	4,7,3	6	0
1,3,2	3,2,9	4	0
1,3,2	6,4,2	3	4
1,2,3	7,2,6	2	1
1,2,3	1,6,5	6	3
M1=0; M2=0; M3=0			

3M 5T 4

Maquinas	Tiempo	E	C
3,2,1	5,3,2	1	5
2,1,3	9,1,3	3	0
1,3,2	2,6,1	4	5
3,2,1	3,2,6	3	3
2,1,3	4,4,4	3	0
M1=2; M2=0; M3=1			

3M 5T 5

Maquinas	Tiempo	E	C
1,3	5,6	1	9
2,1,3	5,7,2	9	8
1,3,2	3,4,3	0	0
2,3,1	1,1,4	3	7
1,2	4,9	5	6
M1=1; M2=1; M3=2			

3M 5T 6

Maquinas	Tiempo	E	C
1,3,2	7,6,7	5	3
3,2,1	6,3,2	5	7
1,2,3	7,4,5	6	4
2,1,3	3,5,1	3	1
2,1	1,2	2	6
M1=3; M2=2; M3=3			

3M 5T 7

Maquinas	Tiempo	E	C
2,3,1	3,6,4	4	0
2,3,1	2,4,6	7	6
3,1,2	3,2,1	4	3
2,1,3	1,7,7	3	4
2,1,3	4,1,3	2	6
M1=0; M2=0; M3=0			

3M 5T 8

Maquinas	Tiempo	E	C
1,3,2	6,6,3	1	0
1,3,2	2,4,9	6	3
1,2,3	1,7,1	5	5
3,2,1	4,2,2	6	8
3,2,1	5,2,6	1	0
M1=0; M2=0; M3=0			

3M 5T 9

Maquinas	Tiempo	E	C
3,1,2	9,5,5	4	9
2,3,1	3,9,8	4	0
3,2,1	5,1,3	2	5
1,3,2	6,4,4	1	0
3,1,2	1,2,1	5	0
M1=1; M2=0; M3=0			

3M 5T 10

Maquinas	Tiempo	E	C
2,3,1	8,3,1	3	7
2,1,3	7,6,6	2	5
1,2,3	4,8,5	1	2
1,3,2	8,7,8	0	4
1,2,3	1,4,1	2	3
M1=0; M2=3; M3=3			

4M 2T 1

Maquinas	Tiempo	E	C
1,2,3,4	7,4,2,5	4	2
1,3,4,2	1,2,6,4	6	0
M1=0; M2=0; M3=0; M4=0			

4M 2T 2

Maquinas	Tiempo	E	C
2,1,4,3	6,6,4,2	1	0
1,3,2,4	4,2,2,6	8	1
M1=; M2=; M3=; M4=			

4M 2T 3

Maquinas	Tiempo	E	C
3,1,4,2	3,2,1,7	3	5
4,2,1,3	3,4,5,1	2	0
M1=1; M2=2; M3=1; M4=2			

4M 2T 4

Maquinas	Tiempo	E	C
1,4,3,2	1,5,4,6	2	3
4,1,3,1	9,2,3,8	4	2
M1=0; M2=0; M3=0; M4=0			

4M 2T 5

Maquinas	Tiempo	E	C
3,2,1,4	4,4,2,9	5	2
1,3,2,4	2,6,5,3	1	1
M1=0; M2=2; M3=2; M4=0			

4M 2T 6

Maquinas	Tiempo	E	C
1,3,4,2	3,6,2,4	5	8
1,2,3	2,4,1	2	1
M1=0; M2=0; M3=0; M4=0			

4M 2T 7

Maquinas	Tiempo	E	C
2,4,1,3	4,2,4,3	4	0
4,1,2,3	1,5,3,1	3	2
M1=1; M2=3; M3=0; M4=2			

4M 2T 8

Maquinas	Tiempo	E	C
1,2,3	2,8,5	3	0
3,4,2,1	6,1,1,5	1	3
M1=0; M2=0; M3=0; M4=0			

4M 2T 9

Maquinas	Tiempo	E	C
3,1,2,4	6,2,3,6	0	3
4,2,3,1	4,9,2,2	4	4
M1=0; M2=2; M3=3; M4=0			

4M 2T 10

Maquinas	Tiempo	E	C
2,3,4,1	5,5,4,5	6	1
2,3,1,4	1,5,5,2	2	2
M1=0; M2=1; M3=1; M4=2			

4M 3T 1

Maquinas	Tiempo	E	C
4,2,1,3	3,3,2,5	8	3
1,2,3,4	4,2,4,3	9	3
3,4,2,1	2,6,3,2	8	2
M1=2; M2=0; M3=0; M4=1			

4M 3T 2

Máquinas	Tiempo	E	C
1,3,4,2	3,6,3,6	4	2
1,2,3,4	3,4,5,2	6	1
1,2,4,3	3,2,3,5	9	0
M1=0; M2=0; M3=0; M4=0			

4M 3T 3

Máquinas	Tiempo	E	C
4,3,1,2	5,2,4,1	0	2
1,2,3,4	6,4,3,4	3	4
2,1,4,3	2,5,2,2	3	7
M1=0; M2=0; M3=0; M4=0			

4M 3T 4

Máquinas	Tiempo	E	C
2,1,3,4	4,5,3,6	6	6
4,3,1,2	4,6,3,5	5	0
2,1,4,3	2,4,2,6	9	2
M1=0; M2=0; M3=0; M4=0			

4M 3T 5

Máquinas	Tiempo	E	C
1,3,4,2	4,2,5,7	9	2
3,2,4,1	2,6,2,6	8	4
1,4,3,2	4,2,5,3	7	3
M1=3; M2=0; M3=1; M4=0			

4M 3T 6

Máquinas	Tiempo	E	C
4,2,1	7,6,4	4	0
1,3,4,2	4,3,2,2	7	5
3,4,1,2	2,5,4,5	9	1
M1=0; M2=0; M3=0; M4=0			

4M 3T 7

Máquinas	Tiempo	E	C
4,3,1,2	3,5,2,6	9	0
4,3,1,2	3,2,6,3	1	0
4,3,2,1	3,2,4,5	0	2
M1=0; M2=0; M3=0; M4=0			

4M 3T 8

Máquinas	Tiempo	E	C
4,3,2	2,5,2	6	2
3,4,2,1	5,6,5,3	4	2
2,3,1,4	1,5,4,4	8	1
M1=3; M2=1; M3=0; M4=0			

4M 3T 9

Máquinas	Tiempo	E	C
3,1,4,2	5,4,6,1	1	9
1,4,3,2	4,6,4,2	7	3
2,1,3,4	1,6,5,4	3	3
M1=0; M2=0; M3=0; M4=0			

4M 3T 10

Máquinas	Tiempo	E	C
2,1,3	5,2,2	5	8
3,2,1,4	5,1,3,5	9	4
4,3,1,2	1,6,5,6	2	9
M1=0; M2=0; M3=0; M4=0			

4M 4T 1

Máquinas	Tiempo	E	C
1,3,2,4	3,6,4,3	7	2
2,4,1,3	4,5,5,1	6	3
3,1,4,2	2,3,1,9	8	0
4,3,2,1	2,5,4,2	9	1
M1=0; M2=0; M3=0; M4=0			

4M 4T 2

Máquinas	Tiempo	E	C
2,4,1,3	5,6,5,5	4	5
4,3,2	4,7,6	0	2
2,1,4,3	3,3,1,4	5	7
4,3,2,1	2,5,9,6	8	3
M1=0; M2=2; M3=1; M4=2			

4M 4T 3

Máquinas	Tiempo	E	C
3,4,1,2	6,3,5,5	7	0
3,2,4,1	5,3,2,1	7	2
1,4,3,2	6,1,6,2	6	2
4,3,2,1	5,3,3,7	5	0
M1=2; M2=0; M3=0; M4=0			

4M 4T 4

Máquinas	Tiempo	E	C
4,3,2,1	6,3,5,5	9	3
1,3,2,4	3,2,3,2	2	9
3,2,4	5,6,7	6	1
2,4,1,3	7,4,2,6	4	5
M1=0; M2=3; M3=2; M4=0			

4M 4T 5

Máquinas	Tiempo	E	C
3,2,4	5,6,6	2	6
2,4,1,3	6,1,6,5	0	2
3,2,1,4	2,4,4,1	2	0
4,1,2	4,3,2	3	0
M1=0; M2=0; M3=0; M4=0			

4M 4T 6

Máquinas	Tiempo	E	C
4,3,2,1	5,6,2,5	8	3
2,1,3,4	6,7,6,5	9	5
4,2,3,1	3,7,1,6	8	6
3,2,1,4	4,6,5,6	9	1
M1=0; M2=0; M3=0; M4=0			

4M 4T 7

Máquinas	Tiempo	E	C
2,3,4,1	6,1,2,2	7	0
4,2,3,1	3,3,6,3	2	3
3,4,2,1	4,4,6,5	4	0
3,2,4,1	7,6,3,2	9	1
M1=0; M2=0; M3=0; M4=0			

4M 4T 8

Máquinas	Tiempo	E	C
1,3,2,4	5,1,2,2	1	4
4,3,1,2	3,6,2,6	6	5
4,2,1,3	5,2,3,5	5	0
1,3,4	2,3,2	3	2
M1=3; M2=2; M3=2; M4=0			

4M 4T 9

Máquinas	Tiempo	E	C
3,4,1,2	5,5,3,3	0	5
2,4,1	4,4,7	2	3
4,2,3,1	7,7,2,5	0	5
1,3,2	2,3,5	9	0
M1=0; M2=0; M3=0; M4=0			

4M 4T 10

Máquinas	Tiempo	E	C
2,4,1,3	4,2,6,5	3	0
2,3,1,4	2,5,3,3	9	0
2,1,3,4	4,6,4,1	8	1
2,3,1,4	3,4,1,5	7	2
M1=; M2=; M3=; M4=			

4M 5T

1

Máquinas	Tiempo	E	C
2,3,4,1	3,5,2,6	7	1
4,2,3,1	4,3,5,3	4	1
1,4,2,3	5,6,3,1	0	0
3,2,1,4	1,2,7,2	6	2
2,1,3,4	4,2,2,4	8	3
M1=0; M2=0; M3=0; M4=0			

4M 5T

2

Máquinas	Tiempo	E	C
3,1,2,4	7,2,6,2	9	1
2,3,4	2,4,3	9	0
1,4,2,3	6,6,2,1	2	0
4,2,3,1	3,3,6,4	8	3
2,3,1,4	1,4,2,6	7	4
M1=0; M2=0; M3=0; M4=0			

4M 5T

3

Máquinas	Tiempo	E	C
3,2,1,4	2,2,3,6	6	7
3,4,2,1	2,6,4,3	3	2
3,1,4,2	3,3,6,7	0	6
3,2,4,1	2,4,2,6	2	0
3,4,2,1	4,5,4,2	1	0
M1=1; M2=3; M3=0; M4=0			

4M 5T

4

Máquinas	Tiempo	E	C
2,3,4,1	5,3,6,3	4	0
4,3,2,1	2,3,3,4	8	0
2,1,3,4	3,1,2,3	7	2
3,2,1,4	2,2,4,4	8	1
1,4,3	1,4,7	3	0
M1=0; M2=0; M3=0; M4=0			

4M 5T

5

Máquinas	Tiempo	E	C
2,3,1,4	3,2,3,6	5	4
4,3,2,1	2,4,2,3	6	0
2,1,4,3	4,1,7,5	7	3
4,2,3,1	6,6,5,4	8	3
3,4,2,1	3,2,5,5	9	1
M1=0; M2=0; M3=2; M4=3			

4M 5T

6

Máquinas	Tiempo	E	C
3,2,4,1	6,6,5,2	7	7
3,4,2,1	3,2,3,3	8	6
2,1,3,4	4,4,3,4	5	5
3,4,2,1	1,5,2,1	6	7
4,2,3,1	2,2,1,2	7	6
M1=; M2=; M3=; M4=			

4M 5T

7

Máquinas	Tiempo	E	C
1,3,2,4	2,1,6,3	8	4
2,3,4	9,2,2	9	9
1,2,3,4	3,3,3,1	9	9
4,2,3,1	1,2,4,2	2	8
3,4,2,1	4,1,3,4	4	2
M1=0; M2=0; M3=0; M4=0			

4M 5T

8

Máquinas	Tiempo	E	C
4,1,3,2	3,5,3,3	5	6
4,2,3,1	2,2,3,4	0	5
3,4,1,2	4,5,4,5	5	0
3,2,1,4	4,4,5,5	3	0
1,2,3,4	3,3,2,4	0	7
M1=0; M2=0; M3=0; M4=0			

4M 5T

9

Máquinas	Tiempo	E	C
2,3,1,4	5,4,6,3	2	3
2,1,4,3	3,3,4,4	1	1
1,4,3,2	3,4,3,5	9	0
3,2,1	5,5,1	8	2
4,1,3	4,2,3	7	0
M1=3; M2=0; M3=1; M4=2			

4M 5T

10

Máquinas	Tiempo	E	C
4,3,1,2	6,3,3,2	6	0
3,4,2,1	3,5,5,6	4	0
2,3,4,1	2,2,4,3	8	3
3,1,4,2	4,4,5,4	7	0
4,2,1,3	5,5,5,3	5	2
M1=0; M2=0; M3=0; M4=0			

5M 2T

1

Máquinas	Tiempo	E	C
1,2,3,4,5	3,3,3,2,3	4	1
5,4,3,2,1	4,5,2,1,2	3	0
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 2T

2

Máquinas	Tiempo	E	C
2,1,5,3,4	4,3,2,6,1	7	4
2,4,3,1,5	3,6,2,3,4	9	2
M1=3; M2=2; M3=0; M4=0; M5=3			

5M 3T 9

Máquinas	Tiempo	E	C
4,2,3,1,5	3,4,6,3,3	8	3
3,1,5,2,4	1,2,5,2,1	8	0
2,3,1,5,4	6,1,4,4,1	9	4
M1=4; M2=2; M3=0; M4=0; M5=5			

5M 3T 10

Máquinas	Tiempo	E	C
1,5,4,3,2	1,4,1,5,6	7	0
2,3,4,1,5	6,4,6,3,3	2	3
1,2,3,5	2,2,3,4	4	0
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 4T 1

Máquinas	Tiempo	E	C
1,5,4,3,2	3,4,6,2,3	7	1
3,1,2,4,5	4,6,2,7,4	9	2
2,3,1,4,5	5,2,4,3,2	2	0
4,2,5,3,1	2,4,5,1,5	8	3
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 4T 2

Máquinas	Tiempo	E	C
3,2,1,5,4	4,2,5,1,6	8	3
3,2,4,1,5	4,2,2,2,4	8	7
3,1,2,4,5	2,5,6,4,6	9	2
3,4,5,2,1	4,1,1,3,2	1	1
M1=0; M2=2; M3=0; M4=1; M5=2			

5M 4T 3

Máquinas	Tiempo	E	C
1,2,3,5	6,4,5,6	0	0
3,2,1,5,4	2,2,2,3,2	3	2
2,3,4,5,1	1,6,5,4,4	8	0
4,3,2,5	4,1,4,5	9	3
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 4T 4

Máquinas	Tiempo	E	C
5,4,1,2,3	6,3,5,3,1	2	4
3,4,2,5,1	3,4,4,2,1	5	4
5,1,4,2,3	2,1,5,4,3	7	0
2,3,1,5,4	4,4,4,5,2	8	3
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 4T 5

Máquinas	Tiempo	E	C
1,3,5,2,4	3,5,1,5,5	9	1
1,4,3,5,2	3,4,2,3,3	4	3
2,1,4,3,5	2,1,4,2,1	0	7
2,5,4,3,1	2,3,6,4,4	2	2
M1=1; M2=0; M3=2; M4=0; M5=1			

5M 4T 6

Máquinas	Tiempo	E	C
5,4,3,1,2	3,6,3,1,3	0	1
4,3,5,2,1	2,3,4,5,7	3	0
2,3,1,5,4	5,5,5,2,2	7	0
1,2,4,3,5	1,2,1,6,4	2	2
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 4T 7

Máquinas	Tiempo	E	C
5,4,2,3,1	6,2,2,6,3	9	1
5,1,4,3,2	5,2,1,3,2	7	3
4,3,1,2,5	3,5,4,2,3	6	2
2,3,1,5,4	1,6,2,5,4	8	4
M1=3; M2=0; M3=2; M4=2; M5=0			

5M 4T 8

Máquinas	Tiempo	E	C
1,4,3,5,2	3,5,6,4,3	5	0
3,2,4,1,5	3,2,2,3,2	7	1
4,5,2,3,1	3,4,1,5,1	3	0
5,3,2,4,1	3,1,3,6,4	4	2
M1=0; M2=0; M3=3; M4=3; M5=0			

5M 4T 9

Máquinas	Tiempo	E	C
5,4,3,1,2	3,3,6,1,3	6	1
3,1,2,4,5	5,2,6,3,3	6	4
4,5,1,2,3	2,6,2,2,2	2	3
5,2,4,3,1	1,1,5,5,2	9	1
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 4T 10

Máquinas	Tiempo	E	C
1,4,3,2,5	5,3,2,1,4	1	3
3,2,5,1,4	6,1,7,4,6	0	0
4,1,5,2,3	3,4,3,4,1	7	0
2,4,1,3	4,2,4,2	7	2
M1=3; M2=0; M3=0; M4=0; M5=0			

5M 5T 1

Máquinas	Tiempo	E	C
3,2,4,1,5	4,2,6,1,4	9	0
2,4,1,3,5	3,1,2,7,4	8	1
4,1,5,2,3	1,5,3,2,5	7	3
1,5,4,3,2	5,4,4,5,7	9	4
3,2,1,5,4	2,1,3,3,2	6	1
M1=0; M2=4; M3=0; M4=3; M5=0			

5M 5T 2

Máquinas	Tiempo	E	C
2,4,3,1,5	3,6,2,2,3	9	6
4,3,1,2,5	2,2,4,2,2	0	2
3,1,2,4,5	5,3,5,3,2	8	7
2,5,4,3,1	1,4,6,1,1	6	1
2,3,4,5,1	4,6,3,3,3	2	3
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 5T 3

Máquinas	Tiempo	E	C
2,4,3,5,1	2,6,3,5,2	4	1
4,1,5,2	3,2,2,1	0	2
3,2,1,5,4	4,6,3,6,2	3	6
5,3,4,2,1	2,2,4,2,3	7	5
1,3,4,2	5,1,2,5	7	3
M1=3; M2=1; M3=0; M4=3; M5=2			

5M 5T 4

Máquinas	Tiempo	E	C
5,4,2,1,3	3,2,5,3,6	6	3
3,4,1,2,5	3,3,5,3,2	8	3
4,1,3,5,2	2,1,2,6,5	0	4
2,3,5,4,1	3,4,6,6,7	1	3
1,2,3,5,4	2,5,3,1,1	9	7
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 5T 5

Máquinas	Tiempo	E	C
1,4,3,5,2	4,6,4,5,4	3	6
2,1,3,5,4	2,3,3,3,6	4	9
2,3,1,4,5	5,6,5,5,2	7	1
4,5,1,2,3	3,1,2,2,1	2	2
3,2,4,1,5	1,4,6,5,5	1	0
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 5T 6

Máquinas	Tiempo	E	C
4,3,1,2,5	1,3,4,4,6	5	4
3,1,2,5	3,6,1,4	4	9
4,2,5,3,1	6,1,7,3,2	3	0
1,5,4,3	4,2,1,2	7	0
2,1,4,5,3	2,4,2,5,6	2	1
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 5T 7

Máquinas	Tiempo	E	C
5,3,1,4,2	4,6,1,2,5	9	3
3,1,4,5,2	1,6,2,2,7	8	3
2,5,3,1,4	3,2,6,5,2	9	2
3,2,4,5,1	5,7,3,3,1	7	1
1,4,5,3,2	2,1,2,2,3	6	6
M1=0; M2=1; M3=0; M4=3; M5=0			

5M 5T 8

Máquinas	Tiempo	E	C
1,4,3,5,2	2,4,1,4,2	6	0
1,3,5,4,2	2,5,5,3,3	0	4
3,2,4,1,5	4,1,4,1,2	3	1
1,5,4,3,2	3,6,2,2,4	8	1
3,4,3,5	1,2,7,4	7	2
M1=0; M2=2; M3=1; M4=2; M5=0			

5M 5T 9

Máquinas	Tiempo	E	C
4,3,1,2	5,1,6,2	5	5
2,5,3,1,4	2,6,2,5,6	0	8
1,2,4,5,3	4,7,1,3,2	6	0
3,1,5,2,4	3,2,4,1,3	2	2
4,5,3,2	6,3,5,2	1	1
M1=0; M2=0; M3=0; M4=0; M5=0			

5M 5T 10

Máquinas	Tiempo	E	C
1,3,4,2,5	6,2,3,5,4	9	6
1,2,5,4,3	6,4,5,6,4	8	1
1,2,4,3,5	2,3,1,3,2	9	0
2,1,3,5,4	3,3,4,1,5	7	3
3,4,1,5,2	4,6,4,2,1	6	0
M1=0; M2=0; M3=0; M4=0; M5=0			

PROBLEMAS MODELO

Problema 6 de S. Lawrence 5M 15T

Máquinas	Tiempo
2,3,5,1,4	21,34,95,53,55
4,5,2,3,1	52,16,71,26,21
3,1,2,4,5	31,12,42,39,98
4,2,5,1,3	77,77 79,55,66
5,4,3,2,1	37,34,64,19,83
3,2,1,4,5	43,54,92,62,79
1,4,2,5,3	93,69,87,77,87
1,2,3,5,4	60,41,38,83,24
3,4,5,1,2	98,17,25,44,49
1,5,4,2,3	96,77,79,75,43
5,3,1,4,2	28,35,95,76,7
1,5,3,2,4	61,10 95,9,35
5,4,2,3,1	59,16,91,59,46
5,2,1,3,4	43,52,28,27,50
1,2,3,5,4	87,45,39,9,41

Problema 30 de S. Lawrence 10M 20T

Máquinas	Tiempo
7,4,2,9,8,5,10,1,6,3	32,16,33,12,70,10,75,82,88,20
9,5,4,6,10,2,7,1,3,8	39,81,91,56,69,45,59,86,36,68
4,3,8,6,5,1,9,10,7,2	84,67,41,73,81,88,38,17,83,5
5,6,3,9,2,1,7,8,4,10	20,6,15,19,30,94,45,17,18,88
10,7,6,5,4,8,9,2,3,1	24,49,16,11,60,5,63,25,15,45
2,9,3,7,10,1,4,8,6,5	86,50,77,54,48,93,32,92,45,71
6,7,4,10,3,1,8,9,5,2	86,90,78,88,57,32,57,86,71,39
3,4,10,5,8,6,9,1,7,2	59,18,31,41,20,83,65,54,94,69
4,5,7,1,2,3,10,6,8,9	47,79,76,59,72,8,30,73,57,84
1,3,5,8,4,6,7,9,10,2	59,89,10,45,8,54,88,20,7,62
6,7,5,4,3,9,10,2,8,1	63,9,77,37,5,13,79,24,10,82
1,2,3,8,5,10,9,4,7,6	74,32,61,53,92,20,10,5,45,23
3,10,1,6,5,7,2,9,4,8	85,51,61,99,37,94,98,65,33,75
1,4,6,7,8,9,3,5,10,2	51,24,8,30,12,23,7,17,35,81
2,6,9,3,7,4,5,10,8,1	71,42,68,31,29,63,65,70,27,93
2,6,5,8,3,9,10,4,7,1	28,38,51,70,33,78,45,90,54,72
1,3,5,7,9,6,8,2,10,4	18,90,25,92,85,35,29,81,80,59
6,3,2,5,1,4,8,7,10,9	67,96,38,86,97,94,86,35,82,45
3,9,5,7,6,10,2,4,8,1	92,51,59,52,8,70,75,54,60,33
4,8,6,1,3,10,2,5,9,7	98,80,78,82,7,89,69,51,79,62

Problema estándar de Fisher/Thompson 6M 6T

Máquinas	Tiempo
3,1,2,4,6,5	1,3,6,7,3,6
2,3,5,6,1,4	8,5,10,10,10,4
3,4,6,1,2,5	5,4,8,9,1,7
2,1,3,4,5,6	5,5,5,3,8,9
3,2,5,6,1,4	9,3,5,4,3,1
2,4,6,1,5,3	3,3,9,10,4,1

Problema estándar de Fisher/Thompson 10M 6T

Máquinas	Tiempo
1,2,3,4,5,6,7,8,9,10	29,78,9,36,49,11,62,56,44,21
1,3,5,10,4,2,7,6,8,9	43,90,75,11,69,28,46,46,72,30
2,1,4,3,9,6,8,7,10,5	91,85,39,74,90,10,12,89,45,33
2,3,1,5,7,9,8,4,10,6	81,95,71,99,9,52,85,98,22,43
3,1,2,6,4,5,9,8,10,7	14,6,22,61,26,69,21,49,72,53
3,2,6,4,9,10,1,7,5,8	84,2,52,95,58,72,47,65,6,25
2,1,4,3,7,6,10,9,8,5	46,37,61,13,32,21,32,89,30,55
3,1,2,6,5,7,9,10,8,4	31,86,46,74,32,88,19,48,36,79
1,2,4,6,3,10,7,8,5,9	76,69,76,51,85,11,40,89,26,74
2,1,3,7,9,10,6,4,5,8	85,13,61,7,64,76,47,52,90,45

Problema estándar de Fisher/Thompson 5M 620T

Máquinas	Tiempo
1,2,3,4,5	29,9,49,62,44
1,2,4,3,5	43,75,69,46,72
2,1,3,5,4	91,39,90,12,45
2,1,5,3,4	81,71,9,85,22
3,2,5,1,4	84,52,48,47,6
2,1,3,4,5	46,61,32,32,30
3,2,1,4,5	31,46,32,19,36
1,4,3,2,5	76,76,85,40,26
2,3,1,4,5	85,61,64,47,90
2,4,1,5,3	78,36,11,56,21
3,1,2,4,5	90,11,28,46,30
1,3,2,4,5	85,74,10,89,33
3,1,2,4,5	95,99,52,98,43
1,2,5,3,4	6,61,69,49,53
2,1,4,5,3	2,95,72,65,25
1,3,2,4,5	37,13,21,89,55
1,2,5,3,4	86,74,88,48,79
2,3,1,4,5	69,51,11,89,74
1,2,3,4,5	13,7,76,52,45

C.2 RESULTADOS DE PROBLEMAS DE CALENDARIZACIÓN

2M 2T

	1	2	3	4	5	6	7	8	9	10
SPT	6	6	8	11	7	10	5	10	21	14
EDD	6	6	10	11	7	10	5	10	21	14
LW	6	6	8	11	7	10	5	10	22	14
MOPNR	6	6	8	11	7	10	5	10	21	14
LORPT	6	6	10	11	7	10	5	10	22	14
MTOR	6	6	10	11	7	10	5	10	22	14
G1A	6	6	8	11	7	10	5	10	21	14
G2A	6	6	8	11	7	10	5	10	21	14
G3A	6	6	8	11	7	10	5	10	21	14
G4A	6	6	10	11	7	10	5	10	22	14
G5A	6	6	10	11	7	10	5	10	22	14
G6A	6	6	10	11	7	10	5	10	22	14
CB1	6	6	8	11	7	10	5	10	21	14
CB2	6	6	8	11	7	10	5	10	21	14
CB3	6	6	8	11	7	10	5	10	21	14
CB4	6	6	10	11	7	10	5	10	22	14
CB5	6	6	10	11	7	10	5	10	22	14
CB6	6	6	10	11	7	10	5	10	22	14
G1B	6	6	10	11	7	10	5	10	22	14
G2B	6	6	10	11	7	10	5	10	22	14
G3B	6	6	10	11	7	10	5	10	22	14
G4B	6	6	10	11	7	10	5	10	22	14
G5B	6	6	10	11	7	10	5	10	22	14
G6B	6	6	10	11	7	10	5	10	22	14

2M 3T

	1	2	3	4	5	6	7	8	9	10
SPT	10	12	15	12	11	17	19	14	22	10
EDD	12	12	18	12	11	17	19	14	22	10
LW	9	12	17	12	11	17	19	15	22	10
MOPNR	10	12	17	11	11	17	15	14	21	9
LORPT	9	12	14	11	11	17	15	15	21	9
MTOR	9	12	14	11	11	17	15	15	21	9
G1A	10	12	14	12	11	17	19	14	22	10
G2A	10	12	14	12	11	17	19	14	22	10
G3A	9	12	14	12	11	17	15	14	21	9
G4A	9	12	14	11	11	17	15	15	21	9
G5A	9	12	14	11	11	17	15	15	21	9
G6A	9	12	14	11	11	17	15	15	21	9
CB1	9	12	14	12	11	17	15	15	22	9
CB2	9	12	14	12	11	17	15	15	22	9
CB3	9	12	14	12	11	17	15	15	21	9
CB4	9	12	14	11	11	17	15	15	21	9
CB5	9	12	14	11	11	17	15	15	21	9
CB6	9	12	14	11	11	17	15	15	21	9
G1B	9	12	14	11	11	17	15	15	21	9
G2B	9	12	14	11	11	17	15	15	21	9
G3B	9	12	14	11	11	17	15	15	21	9
G4B	9	12	14	11	11	17	15	15	21	9
G5B	9	12	14	11	11	17	15	15	21	9
G6B	9	12	14	11	11	17	15	15	21	9

2M 4T

	1	2	3	4	5	6	7	8	9	10
SPT	20	16	25	14	12	18	12	24	16	24
EDD	22	16	21	14	12	19	12	24	16	24
LW	21	17	21	14	12	18	13	24	16	26
MOPNR	21	16	20	14	12	19	12	24	16	24
LORPT	22	16	21	14	13	19	12	24	16	24
MTOR	22	16	21	14	12	19	12	24	16	24
G1A	20	16	25	14	12	18	12	24	16	24
G2A	20	16	25	14	12	18	12	24	16	24
G3A	20	16	20	14	12	16	12	24	16	24
G4A	22	16	21	14	13	19	12	24	16	24
G5A	22	16	20	14	12	19	12	24	16	24
G6A	22	16	21	14	13	19	12	24	16	24
R1	20	16	25	14	12	18	12	24	16	24
CB1	20	16	21	14	12	19	12	24	16	24
CB2	20	16	21	14	12	19	12	24	16	24
CB3	20	16	21	14	12	19	12	24	16	24
CB4	22	16	21	14	13	19	12	24	16	24
CB5	22	16	21	14	12	19	12	24	16	24
CB6	22	16	21	14	13	19	12	24	16	24
G1B	22	16	21	14	12	19	12	24	16	24
G2B	22	16	21	14	12	19	12	24	16	24
G3B	22	16	21	14	12	19	12	24	16	24
G4B	22	16	21	14	12	19	12	24	16	24
G5B	22	16	20	14	12	19	12	24	16	24
G6B	22	16	21	14	13	19	12	24	16	24

2M 5T

	1	2	3	4	5	6	7	8	9	10
SPT	21	20	28	18	24	16	26	17	25	21
EDD	21	24	23	20	21	16	25	18	25	31
LW	21	23	24	20	24	18	25	17	25	26
MOPNR	21	20	23	17	20	16	21	18	25	20
LORPT	21	21	23	17	26	19	21	19	25	22
MTOR	21	21	23	17	23	16	21	19	25	22
G1A	21	20	28	18	24	16	26	17	25	21
G2A	21	20	28	18	24	16	26	17	25	21
G3A	21	20	23	17	20	16	21	17	25	22
G4A	21	21	23	17	26	18	21	19	25	22
G5A	21	20	23	17	20	16	21	19	25	22
G6A	21	21	23	17	26	18	21	19	25	22
R1	21	20	28	18	24	16	26	17	25	21
CB1	21	21	23	17	23	16	21	17	25	22
CB2	21	21	23	17	23	16	21	17	25	22
CB3	21	20	23	17	23	16	21	17	25	22
CB4	21	20	23	17	23	16	21	19	25	22
CB5	21	20	23	17	23	16	21	19	25	22
CB6	21	20	23	17	23	16	21	19	25	22
G1B	21	20	25	17	23	16	21	19	25	22
G2B	21	20	25	17	23	16	21	19	25	22
G3B	21	20	25	17	23	16	21	19	25	22
G4B	21	20	25	17	23	16	21	19	25	22
G5B	21	20	25	17	20	16	21	19	25	22
G6B	21	21	25	17	26	18	21	19	25	22

3M 2T

	1	2	3	4	5	6	7	8	9	10
SPT	14	18	12	17	23	15	16	18	14	19
EDD	14	18	12	17	23	15	18	18	14	19
LW	14	18	12	17	23	15	18	18	13	19
MOPNR	14	20	12	17	22	15	18	18	13	19
LORPT	14	20	12	17	22	15	16	18	13	19
MTOR	14	20	12	17	22	15	16	18	13	19
G1A	14	18	12	17	23	15	16	18	14	19
G2A	14	18	12	17	23	15	16	18	14	19
G3A	14	18	12	17	23	15	16	18	14	19
G4A	14	20	12	17	22	15	16	18	14	19
G5A	14	20	12	17	22	15	16	18	14	19
G6A	14	20	12	17	22	15	16	18	14	19
CB1	<u>14</u>	<u>18</u>	12	17	22	<u>15</u>	<u>16</u>	18	<u>14</u>	19
CB2	<u>14</u>	<u>18</u>	12	17	22	<u>15</u>	<u>16</u>	18	<u>14</u>	19
CB3	<u>14</u>	<u>18</u>	12	17	22	<u>15</u>	<u>16</u>	18	<u>14</u>	19
CB4	<u>14</u>	<u>20</u>	12	17	22	<u>15</u>	<u>16</u>	18	<u>14</u>	19
CB5	<u>14</u>	<u>20</u>	12	17	22	<u>15</u>	<u>16</u>	18	<u>14</u>	19
CB6	<u>14</u>	<u>20</u>	12	17	22	<u>15</u>	<u>16</u>	18	<u>14</u>	19
G1B	14	20	12	17	22	15	16	18	13	19
G2B	14	20	12	17	22	15	16	18	13	19
G3B	14	20	12	17	22	15	16	18	13	19
G4B	14	20	12	17	22	15	16	18	13	19
G5B	14	20	12	17	22	15	16	18	13	19
G6B	14	20	12	17	22	15	16	18	13	19

3M 3T

	1	2	3	4	5	6	7	8	9	10
SPT	17	19	21	25	17	20	18	16	14	20
EDD	17	19	21	22	18	20	18	16	14	20
LW	17	19	28	26	18	26	18	16	14	20
MOPNR	17	16	21	26	18	20	18	16	14	20
LORPT	17	16	21	22	18	20	21	16	13	20
MTOR	17	16	21	22	18	20	21	16	13	20
G1A	17	19	21	25	17	20	21	16	13	20
G2A	17	19	21	25	17	20	21	16	13	20
G3A	17	19	21	25	17	20	21	16	13	20
G4A	17	16	21	22	18	20	21	16	13	20
G5A	17	16	21	22	18	20	21	16	13	20
G6A	17	16	21	22	18	20	21	16	13	20
CB1	<u>17</u>	<u>19</u>	<u>21</u>	<u>22</u>	<u>18</u>	20	<u>21</u>	16	<u>13</u>	20
CB2	<u>17</u>	<u>19</u>	<u>21</u>	<u>22</u>	<u>18</u>	20	<u>21</u>	16	<u>13</u>	20
CB3	<u>17</u>	<u>16</u>	<u>21</u>	<u>22</u>	<u>18</u>	20	<u>21</u>	16	<u>13</u>	20
CB4	<u>17</u>	<u>16</u>	<u>21</u>	<u>22</u>	18	20	<u>21</u>	16	<u>13</u>	20
CB5	<u>17</u>	<u>16</u>	<u>21</u>	<u>22</u>	18	20	<u>21</u>	16	<u>13</u>	20
CB6	<u>17</u>	<u>16</u>	<u>21</u>	<u>22</u>	18	20	<u>21</u>	16	<u>13</u>	20
G1B	17	16	20	22	18	20	21	16	13	20
G2B	17	16	20	22	18	20	21	16	13	20
G3B	17	16	20	22	18	20	21	16	13	20
G4B	17	16	20	22	18	20	21	16	13	20
G5B	17	16	20	22	18	20	21	16	13	20
G6B	17	16	20	22	18	20	21	16	13	20

3M 4T

	1	2	3	4	5	6	7	8	9	10
SPT	21	37	21	22	25	20	13	24	27	30
EDD	20	27	24	22	27	20	22	29	31	30
LW	21	27	22	32	27	20	14	24	31	30
MOPNR	20	30	23	22	25	21	13	24	31	30
LORPT	20	26	21	32	25	20	13	24	32	32
MTOR	20	26	20	22	25	20	13	24	32	30
G1A	21	37	20	22	25	20	13	24	27	30
G2A	21	37	20	22	25	20	13	24	27	30
G3A	21	30	20	22	25	20	13	24	27	30
G4A	20	26	20	32	25	20	13	24	32	32
G5A	20	26	20	22	25	21	13	24	32	30
G6A	20	26	20	32	25	20	13	24	32	32
CB1	<u>21</u>	<u>31</u>	20	<u>22</u>	<u>25</u>	20	<u>13</u>	24	32	30
CB2	<u>21</u>	<u>31</u>	20	<u>22</u>	<u>25</u>	20	<u>13</u>	24	32	30
CB3	<u>21</u>	<u>30</u>	20	<u>22</u>	<u>25</u>	20	<u>13</u>	24	32	30
CB4	20	<u>26</u>	20	<u>22</u>	<u>25</u>	20	<u>13</u>	24	32	30
CB5	20	<u>26</u>	20	<u>22</u>	<u>25</u>	20	<u>13</u>	24	32	30
CB6	20	<u>26</u>	20	<u>22</u>	<u>25</u>	20	<u>13</u>	24	32	30
G1B	20	26	20	22	25	20	13	25	32	30
G2B	20	26	20	22	25	20	13	25	32	30
G3B	20	26	20	22	25	20	13	25	32	30
G4B	20	26	20	22	25	20	13	25	32	30
G5B	20	26	20	22	25	21	13	25	32	30
G6B	20	26	20	32	25	20	13	25	32	32

3M 5T

	1	2	3	4	5	6	7	8	9	10
SPT	33	24	32	22	26	34	23	27	36	38
EDD	34	26	48	28	26	34	32	28	38	38
LW	30	24	35	27	29	34	27	30	34	38
MOPNR	30	24	33	25	29	32	24	30	32	38
LORPT	33	24	38	25	25	32	24	30	32	38
MTOR	31	24	38	25	24	32	24	30	33	38
G1A	33	24	32	22	26	34	23	27	36	38
G2A	33	24	32	22	26	34	23	27	36	38
G3A	30	24	32	23	24	32	23	25	36	38
G4A	33	24	38	25	25	32	24	30	32	38
G5A	30	24	38	25	24	32	24	30	32	38
G6A	33	24	38	25	25	32	24	30	32	38
CB1	31	<u>24</u>	32	<u>22</u>	24	32	<u>23</u>	27	<u>33</u>	<u>38</u>
CB2	31	<u>24</u>	32	<u>22</u>	24	32	<u>23</u>	27	<u>33</u>	<u>38</u>
CB3	31	<u>24</u>	32	23	<u>24</u>	32	<u>23</u>	<u>25</u>	<u>33</u>	<u>38</u>
CB4	31	<u>24</u>	<u>38</u>	25	<u>24</u>	32	<u>24</u>	<u>30</u>	<u>33</u>	<u>38</u>
CB5	31	<u>24</u>	<u>38</u>	25	<u>24</u>	32	<u>24</u>	<u>30</u>	<u>33</u>	<u>38</u>
CB6	31	<u>24</u>	<u>38</u>	25	<u>24</u>	32	<u>24</u>	<u>30</u>	<u>33</u>	<u>38</u>
G1B	31	24	38	25	24	32	24	30	33	38
G2B	31	24	38	25	24	32	24	30	33	38
G3B	31	24	38	25	24	32	24	30	33	38
G4B	31	24	38	25	24	32	24	30	33	38
G5B	30	24	38	25	24	32	24	30	32	38
G6B	33	24	38	25	25	32	24	30	32	38

4M 2T

	1	2	3	4	5	6	7	8	9	10
SPT	19	23	16	24	24	20	16	24	22	23
EDD	21	23	16	24	28	20	16	24	22	23
LW	21	23	16	24	24	15	16	24	22	23
MOPNR	21	23	16	24	24	15	16	24	22	27
LORPT	21	23	16	24	24	15	16	24	22	27
MTOR	21	23	16	24	24	15	16	24	22	27
G1A	19	23	16	24	24	20	16	24	22	23
G2A	19	23	16	24	24	20	16	24	22	23
G3A	19	23	16	24	24	15	16	24	22	21
G4A	19	23	16	24	24	15	16	24	22	27
G5A	19	23	16	24	24	15	16	24	22	27
G6A	19	23	16	24	24	15	16	24	22	27
CB1	19	23	16	24	24	20	16	24	22	23
CB2	19	23	16	24	24	20	16	24	22	23
CB3	19	23	16	24	24	15	16	24	22	21
CB4	19	23	16	24	24	15	16	24	22	27
CB5	19	23	16	24	24	15	16	24	22	27
CB6	19	23	16	24	24	15	16	24	22	27
G1B	21	23	16	24	24	16	16	24	22	27
G2B	21	23	16	24	24	16	16	24	22	27
G3B	21	23	16	24	24	16	16	24	22	27
G4B	21	23	16	24	24	16	16	24	22	27
G5B	21	23	16	24	24	16	16	24	22	27
G6B	21	23	16	24	24	16	16	24	22	27

4M 3T

	1	2	3	4	5	6	7	8	9	10
SPT	18	20	17	25	28	21	28	27	24	22
EDD	21	20	17	27	29	21	27	32	24	22
LW	21	20	17	27	29	21	25	27	24	22
MOPNR	18	20	17	21	24	21	25	23	24	22
LORPT	18	20	17	21	24	21	25	23	24	22
MTOR	18	20	17	21	24	21	25	23	24	22
G1A	18	20	17	25	28	21	23	23	24	22
G2A	18	20	17	25	28	21	23	23	24	22
G3A	18	20	17	25	28	21	23	23	24	22
G4A	18	20	17	21	24	21	23	23	24	22
G5A	18	20	17	21	24	21	23	23	24	22
G6A	18	20	17	21	24	21	23	23	24	22
CB1	18	20	17	25	28	21	23	23	24	22
CB2	18	20	17	25	28	21	23	23	24	22
CB3	18	20	17	25	28	21	23	23	24	22
CB4	18	20	17	21	24	21	23	23	24	22
CB5	18	20	17	21	24	21	23	23	24	22
CB6	18	20	17	21	24	21	23	23	24	22
G1B	18	20	17	21	24	21	25	20	24	22
G2B	18	20	17	21	24	21	25	20	24	22
G3B	18	20	17	21	24	21	25	20	24	22
G4B	18	20	17	21	24	21	25	20	24	22
G5B	18	20	17	21	24	21	25	20	24	22
G6B	18	20	17	21	24	21	25	20	24	22

4M 4T

	1	2	3	4	5	6	7	8	9	10
SPT	27	35	29	31	23	32	27	24	28	30
EDD	27	39	25	36	23	32	28	23	28	29
LW	27	35	28	31	23	37	28	20	28	27
MOPNR	27	33	25	31	23	32	27	20	28	30
LORPT	27	33	24	34	23	32	28	20	28	30
MTOR	27	33	25	34	23	32	28	20	28	30
G1A	27	35	29	31	23	32	27	24	28	30
G2A	27	35	29	31	23	32	27	24	28	30
G3A	27	35	27	31	23	32	27	20	28	30
G4A	27	33	24	34	23	32	28	20	28	30
G5A	27	33	25	34	23	32	27	20	28	30
G6A	27	33	24	34	23	32	28	20	28	30
CB1	27	33	28	34	23	32	27	24	28	30
CB2	27	33	28	34	23	32	27	24	28	30
CB3	27	33	27	34	23	32	27	20	28	30
CB4	27	33	24	34	23	32	28	20	28	30
CB5	27	33	25	34	23	32	27	20	28	30
CB6	27	33	24	34	23	32	28	20	28	30
G1B	27	33	25	34	23	32	28	20	28	30
G2B	27	33	25	34	23	32	28	20	28	30
G3B	27	33	25	34	23	32	28	20	28	30
G4B	27	33	25	34	23	32	28	20	28	30
G5B	27	33	25	34	23	32	28	20	28	30
G6B	27	33	24	34	23	32	27	20	28	30

4M 5T

	1	2	3	4	5	6	7	8	9	10
SPT	23	25	36	25	31	25	27	25	24	32
EDD	23	24	36	26	33	25	27	29	27	31
LW	28	24	38	25	32	25	27	24	20	31
MOPNR	28	25	35	22	31	25	27	25	20	30
LORPT	28	24	34	21	32	25	27	25	20	31
MTOR	28	24	32	22	32	25	27	25	20	29
G1A	23	25	31	25	31	25	27	25	27	32
G2A	23	25	31	25	31	25	27	25	27	32
G3A	25	25	31	27	31	25	27	25	22	29
G4A	25	24	34	21	32	25	27	24	20	31
G5A	25	24	32	22	32	25	27	24	20	29
G6A	25	24	34	21	32	25	27	24	20	31
CB1	23	24	31	25	32	25	27	25	23	30
CB2	23	24	31	25	32	25	27	25	23	30
CB3	25	24	31	22	32	25	27	25	22	29
CB4	25	24	34	21	32	25	27	24	20	29
CB5	25	24	32	22	32	25	27	24	20	29
CB6	25	24	34	21	32	25	27	24	20	29
G1B	25	24	32	22	32	25	27	25	20	29
G2B	25	24	32	22	32	25	27	25	20	29
G3B	25	24	32	22	32	25	27	25	20	29
G4B	25	24	32	22	32	25	27	25	20	29
G5B	25	24	32	22	32	25	27	25	20	29
G6B	25	24	34	21	32	25	27	25	20	31

5M 2T

	1	2	3	4	5	6	7	8	9	10
SPT	14	21	22	23	19	25	29	32	29	25
EDD	14	26	22	23	19	25	29	29	29	30
LW	14	26	22	23	19	25	29	32	29	25
MOPNR	14	26	22	23	19	26	29	32	29	25
LORPT	14	21	22	23	19	26	29	29	29	25
MTOR	14	21	22	23	19	26	29	29	29	25
G1A	14	21	22	23	19	25	29	32	29	25
G2A	14	21	22	23	19	25	29	32	29	25
G3A	14	21	22	23	19	25	29	32	29	25
G4A	14	21	22	23	19	26	29	29	29	25
G5A	14	21	22	23	19	26	29	29	29	25
G6A	14	21	22	23	19	26	29	29	29	25
CB1	14	21	<u>22</u>	23	19	25	29	29	<u>29</u>	<u>25</u>
CB2	14	21	<u>22</u>	23	19	25	29	29	<u>29</u>	<u>25</u>
CB3	14	21	<u>22</u>	23	19	25	29	29	<u>29</u>	<u>25</u>
CB4	14	21	<u>22</u>	23	19	26	29	<u>29</u>	<u>29</u>	<u>25</u>
CB5	14	21	<u>22</u>	23	19	26	29	<u>29</u>	<u>29</u>	<u>25</u>
CB6	14	21	<u>22</u>	23	19	26	29	<u>29</u>	<u>29</u>	<u>25</u>
G1B	14	21	22	23	19	26	29	29	29	25
G2B	14	21	22	23	19	26	29	29	29	25
G3B	14	21	22	23	19	26	29	29	29	25
G4B	14	21	22	23	19	26	29	29	29	25
G5B	14	21	22	23	19	26	29	29	29	25
G6B	14	21	22	23	19	26	29	29	29	25

5M 3T

	1	2	3	4	5	6	7	8	9	10
SPT	22	21	26	24	35	32	20	27	24	27
EDD	22	21	23	23	35	32	22	23	24	31
LW	22	21	28	24	36	32	22	23	24	25
MOPNR	22	21	23	23	35	29	20	27	24	25
LORPT	22	21	23	23	35	29	20	23	24	25
MTOR	22	21	23	23	35	29	20	23	24	25
G1A	22	21	26	24	35	32	20	27	24	27
G2A	22	21	26	24	35	32	20	27	24	27
G3A	22	21	23	24	35	29	20	27	24	27
G4A	22	21	23	23	35	29	20	23	24	25
G5A	22	21	23	23	35	29	20	23	24	25
G6A	22	21	23	23	35	29	20	23	24	25
CB1	22	21	26	24	35	29	20	23	<u>24</u>	27
CB2	22	21	26	24	35	29	20	23	<u>24</u>	27
CB3	22	21	23	24	35	29	20	23	<u>24</u>	27
CB4	22	21	23	23	35	29	20	23	<u>24</u>	25
CB5	22	21	23	23	35	29	20	23	<u>24</u>	25
CB6	22	21	23	23	35	29	20	23	<u>24</u>	25
G1B	22	21	26	23	35	29	20	23	24	25
G2B	22	21	26	23	35	29	20	23	24	25
G3B	22	21	26	23	35	29	20	23	24	25
G4B	22	21	26	23	35	29	20	23	24	25
G5B	22	21	26	23	35	29	20	23	24	25
G6B	22	21	26	23	35	29	20	23	24	25

5M 4T

	1	2	3	4	5	6	7	8	9	10
SPT	26	29	28	24	27	23	25	24	23	27
EDD	26	39	28	23	26	23	25	26	28	29
LW	26	38	28	32	26	26	25	26	28	29
MOPNR	26	35	28	23	29	23	25	24	19	29
LORPT	26	30	28	23	24	26	24	26	19	29
MTOR	26	39	28	23	24	23	24	24	19	29
G1A	26	29	28	24	27	23	25	24	23	27
G2A	26	29	28	24	27	23	25	24	23	27
G3A	26	29	28	24	24	23	25	24	23	29
G4A	26	30	28	23	24	26	24	26	19	29
G5A	26	29	28	23	24	23	24	24	19	29
G6A	26	30	28	23	24	26	24	26	19	29
CB1	26	29	<u>28</u>	23	24	23	25	24	23	<u>29</u>
CB2	26	29	<u>28</u>	23	24	23	25	24	23	<u>29</u>
CB3	26	29	<u>28</u>	23	<u>24</u>	23	25	24	23	<u>29</u>
CB4	26	30	<u>28</u>	23	<u>24</u>	26	24	24	<u>19</u>	<u>29</u>
CB5	26	29	<u>28</u>	23	<u>24</u>	23	24	24	<u>19</u>	<u>29</u>
CB6	26	30	<u>28</u>	23	<u>24</u>	26	24	24	<u>19</u>	<u>29</u>
G1B	26	29	28	23	24	23	24	24	19	26
G2B	26	29	28	23	24	23	24	24	19	26
G3B	26	29	28	23	24	23	24	24	19	26
G4B	26	29	28	23	24	23	24	24	19	26
G5B	26	29	28	23	24	23	24	25	19	26
G6B	26	30	28	23	24	26	24	26	19	26

5M 5T

	1	2	3	4	5	6	7	8	9	10
SPT	30	27	26	30	33	30	28	34	25	40
EDD	30	28	26	30	33	29	31	29	25	40
LW	34	29	26	30	34	36	28	28	24	37
MOPNR	34	27	27	26	32	30	28	29	23	32
LORPT	30	27	26	26	31	29	31	28	24	33
MTOR	30	27	26	26	30	29	31	28	24	30
G1A	30	27	26	30	33	30	28	28	23	36
G2A	30	27	26	30	33	30	28	28	23	36
G3A	30	27	26	26	32	30	28	29	23	30
G4A	30	27	26	26	31	29	31	28	24	33
G5A	30	27	26	26	32	29	31	28	24	30
G6A	30	27	26	26	31	29	31	28	24	33
CB1	36	27	27	30	33	<u>30</u>	<u>28</u>	<u>29</u>	<u>24</u>	30
CB2	36	27	27	30	33	<u>30</u>	<u>28</u>	<u>29</u>	<u>24</u>	30
CB3	30	27	27	26	32	<u>30</u>	<u>28</u>	<u>29</u>	<u>24</u>	30
CB4	30	27	26	26	31	29	<u>31</u>	<u>28</u>	<u>24</u>	30
CB5	30	27	26	26	30	29	<u>31</u>	<u>28</u>	<u>24</u>	30
CB6	30	27	26	26	31	29	<u>31</u>	<u>28</u>	<u>24</u>	30
G1B	30	27	26	26	30	29	31	28	<u>24</u>	32
G2B	30	27	26	26	30	29	31	28	24	32
G3B	30	27	26	26	30	29	31	28	24	<u>32</u>
G4B	30	27	26	26	30	29	31	28	24	32
G5B	30	27	26	26	30	29	31	28	24	<u>30</u>
G6B	30	27	26	26	31	29	31	28	24	32

SOLUCIÓN DE PROBLEMAS MODELO

En sección se presentan algunos resultados obtenidos en la solución de problemas modelo propuestos por algunos autores, en los cuales se conoce el valor óptimo de $C_{\text{máx}}$ o el mejor resultado obtenido hasta el momento. Se presentan los siguientes problemas:

- Problema 6 de S. Lawrence de 5 máquinas y 15 trabajos.
- Problema 30 de S. Lawrence de 10 máquinas 20 trabajos.
- Problema estándar de Fisher/Thompson de 6 máquinas y 6 trabajos.
- Problema estándar de Fisher/Thompson de 10 máquinas y 10 trabajos.
- Problema estándar de Fisher/Thompson de 5 máquinas y 20 trabajos.

Problema 6 de S. Lawrence de 5 máquinas y 15 trabajos.

El valor óptimo de $C_{\text{máx}}$ para este problema es de 926 unidades de tiempo. Al analizar este modelo con el método propuesto en esta tesis, se obtuvo el mismo resultado (926) con todos los grupos GB, además de G3A, G4A, G5A, G6A, CB3, CB4, CB5, y CB6. Por lo tanto se obtuvo el resultado óptimo con el método propuesto.

Problema 30 de S. Lawrence de 10 máquinas 20 trabajos.

El valor óptimo de $C_{\text{máx}}$ para este problema es de 1355 unidades de tiempo. En este caso no se obtuvo el valor óptimo. Los grupos de reglas que alcanzaron un mejor resultado fueron G4A, G6A, CB4, CB6 y GB6 con 1555 unidades de tiempo, es decir, tiene una desviación del 14.7% de la solución óptima.

Problema estándar de Fisher/Thompson de 6 máquinas y 6 trabajos.

El valor óptimo de $C_{\text{máx}}$ en la solución del problema es 55 unidades de tiempo. Todos los grupos GB del método propuesto, además de G4A, G5A, G6A, LORPT y MTOR obtuvieron el mejor resultado con 59 unidades. La desviación de la solución óptima es de 7.27%.

Problema estándar de Fisher/Thompson de 10 máquinas y 10 trabajos.

El valor óptimo de $C_{\text{máx}}$ es de 930 unidades de tiempo. La reglas SPT y los grupos G1A, G2A, CB1 y CB2 fueron los que obtuvieron el mejor resultado con 1103 unidades. La desviación de la solución óptima es de 18.6%. Se observa que la regla SPT es la que ayuda a obtener este resultado ya que los grupos también la incluyen.

Problema estándar de Fisher/Thompson de 5 máquinas y 20 trabajos.

El valor óptimo de $C_{\text{máx}}$ es de 1165 unidades de tiempo. El grupo CB1 con la máquina 4 como cuello de botella, es la combinación que proporciona el mejor resultado con 1312 unidades, es decir, la desviación es de 12.6% de la solución óptima.

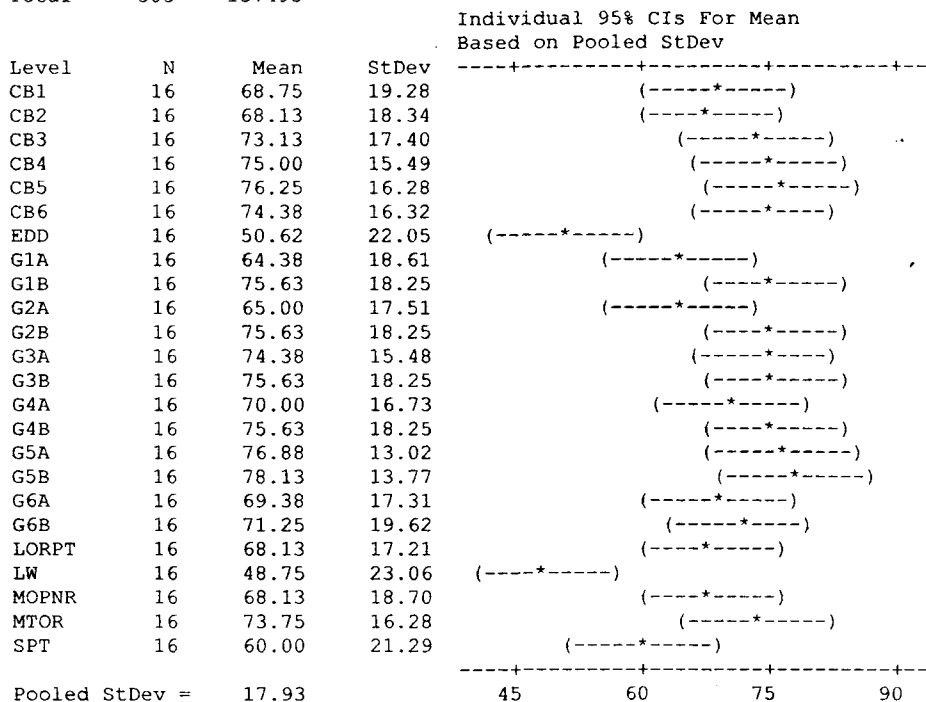
Como se puede observar, con los resultados se confirma que los algoritmos heurísticos difícilmente obtienen el valor óptimo en la solución, pero logran un buen resultado. De los 5 problemas analizados, la mayor desviación con respecto a la solución óptima es de 18.6%. Por el lado contrario en otro problema se obtuvo exactamente la solución óptima.

ANEXO D

RESULTADOS DE MINITAB

Analysis of Variance for PORCEN.

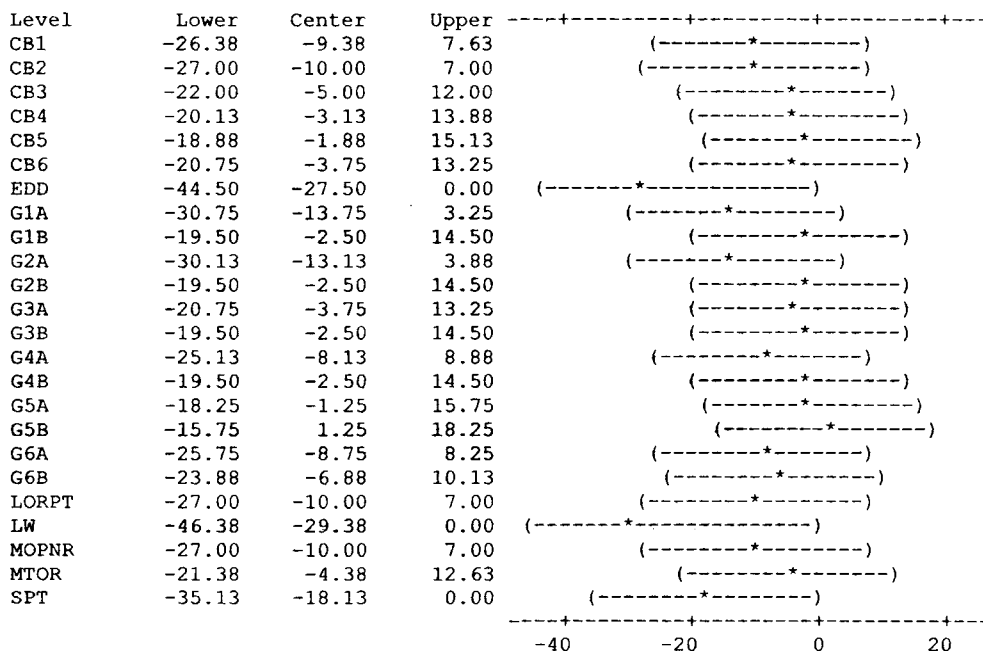
Source	DF	SS	MS	F	P
REGLAS	23	21812	948	2.95	0.000
Error	360	115681	321		
Total	383	137493			



Hsu's MCB (Multiple Comparisons with the Best)

Family error rate = 0.0500
Critical value = 2.68

Intervals for level mean minus largest of other level means



ANEXO E

