

Aplicación del fraccionamiento de operaciones en una heurística constructiva en programación secuencial para asignación de varios trabajos a varias máquinas en paralelo.

Application of splitting of operations in a constructive heuristic sequential programming for assigning multiple jobs to multiple machines in parallel

Pedro Pablo Ballesteros Silva, Cindy Juranny Duque Vanegas

Coordinador Especialización en Logística Empresarial, Asistente Administrativa Especialización en Logística Empresarial, Universidad Tecnológica de Pereira, Pereira, Colombia

ppbs@utp.edu.co
cjuque@utp.edu.co

Resumen— Este artículo trata de la aplicación del fraccionamiento razonable de operaciones hasta donde sea posible en una heurística para resolver un problema de programación secuencial de varios trabajos a varias máquinas en paralelo, en un ambiente de producción por pedidos, que se utiliza en organizaciones donde cada pedido del cliente es único y posee sus propios parámetros y condiciones. Se hace una breve introducción de los conceptos de programación secuencial, sus elementos constitutivos, relación de medidas de desempeño y se complementa con la presentación de un estudio de caso, con su enunciado, la organización de la información, la secuencia tecnológica, las rutas de cada tarea, el grafo que representa el sistema productivo, la lista de precedencias, el programa de ejecución factible, los criterios de decisión para la programación de operaciones, la aplicación de Matlab para resolver el problema y se desarrolla la propuesta de mejoramiento a partir del fraccionamiento razonable de operaciones.

Al final, se relacionan valiosas conclusiones que pueden tomarse como producto de esta importante heurística, que podrá servir para mejorar el control de producción en cualquier empresa y su aplicación en un caso específico de la producción por pedidos.

Palabras clave— Optimización, algoritmos, fraccionamiento de operaciones, programación secuencial, heurísticas constructivas, talleres, operaciones, máquinas, parámetros y variables, medidas de desempeño, restricciones, secuencia tecnológica, rutas, lista de precedencia, programa factible, criterios de decisión.

Abstract— this paper is about the application of heuristics to solve a programming problem sequential multiple jobs to multiple machines in parallel, in a production by order, which applies in organizations where each customer order is unique and has its own parameters and conditions. A brief introduction of sequential programming concepts, it's constituent elements, performance measures relationship and is complemented by a case study presentation, with it's statement, the organization of information, the technological sequence, routes each task, the graph that represents the production system, the precedence list,

the feasible implementation program, the decision criteria for programming operations and the application of Matlab to solve the problem.

In the end, relate valuable conclusions that can be taken as a result of this important heuristic, which may serve to improve the control of any enterprise production and it's application in a specific case of the production orders.

Key Word — Optimization, algorithms, splitting of operations, sequential programming, constructive heuristics, workshop, operations, machines, parameters and variables, performance measures, constraints, technological sequence, routes, list of precedence feasible program, decision criteria.

I. INTRODUCCIÓN

Desde hace mucho tiempo, uno de los mayores problemas que se presenta en la administración de operaciones de cualquier empresa, ya sea en el área de manufactura como en el área de servicios es precisamente la programación secuencial de actividades de los diferentes trabajos propuestos por los clientes.

Existen dificultades o restricciones en la asignación de recursos, tanto en máquinas, instalaciones, personal o insumos; en ocasiones la capacidad de producción no es suficiente para atender en forma adecuada la demanda de productos o servicios, se presentan cuellos de botella, factores que afectan el normal desarrollo de las actividades e inciden en el cumplimiento de las fechas de entrega. Estos problemas presentan en su solución un significativo nivel de dificultad y se han convertido en el tema de investigación de muchos expertos, quienes han propuesto soluciones que van desde heurísticas, metaheurísticas y la aplicación de métodos exactos, utilizando técnicas manuales o haciendo uso de software especializados.

El problema consiste en que dado un grupo de trabajos que se efectuaran en varias máquinas en forma paralela, se debe proponer un programa de producción que minimice el tiempo total de terminación del conjunto de tareas, C_{max} , y que muestre una buena solución manual y con uso de equipo de cómputo en tiempos razonables.

Para su solución se aplica una heurística constructiva que se valida con el código en Matlab de dicho problema.

La literatura especializada en programación secuencial utiliza diferentes técnicas para resolver esta clase de problemas incluyendo algoritmos como Branch and Bound, Branch Price, Generación de Columnas, Giffler y Thompson (para problemas activos y sin dilación de máquinas), Johnson (para problemas flow shop de 2 y 3 máquinas), Johnson (para problemas job shop de 2 máquinas y múltiples trabajos), método de Page, algoritmo CDS, algoritmo NEH, el método de Gupta, algoritmo IG, técnicas de descomposición de Benders y Danzig Wolf, relajación lineal y lagrangeana.

II. CONCEPTOS BÁSICOS

A. Qué es Programación secuencial.

Un punto de partida básico para este trabajo es definir claramente qué significa programación secuencial. Como se menciona en [1], consiste en un proceso de toma de decisiones que se utiliza sobre una base regular en industrias manufactureras y de servicios [4]. Está asociada a la asignación de los recursos a las diferentes tareas durante períodos de tiempo determinados, y la meta es optimizar uno o más objetivos o medidas de desempeño [3].

Los recursos y tareas en una organización pueden adoptar muchas formas diferentes.

Los recursos pueden ser máquinas en un taller, pistas de aterrizaje en un aeropuerto, equipos en un sitio de construcción, las unidades de procesamiento en un entorno de computación, y así sucesivamente.

Las tareas pueden ser operaciones en un proceso de producción, los despegues y aterrizajes en un aeropuerto, las etapas de un proyecto de construcción, las ejecuciones de programas de ordenador, entre otros. Cada tarea puede tener un nivel de prioridad determinado, una partida lo antes posible, tiempo y una fecha de vencimiento. Los objetivos también pueden adoptar muchas formas diferentes [5].

Un objetivo puede ser la minimización del tiempo de finalización de la última tarea y otro puede ser la minimización del número de tareas que se han completado respectivas fechas de vencimiento [11].

La programación secuencial, entendida como un proceso de toma de decisiones, desempeña un papel trascendental en la mayoría de los sistemas de fabricación y producción, así como en el medio ambiente de procesamiento de la información [9].

También es importante en la configuración de transporte y distribución y en otros tipos de industrias de servicios.

Los siguientes ejemplos ilustran el papel de programación en una serie de entornos del mundo real: fábricas de bolsas de papel, una planta de fabricación de semiconductores, asignación de entradas en un aeropuerto, programación de tareas en una unidad central de proceso.

También la programación secuencial se puede considerar como la herramienta administrativa que se ocupa de buscar un orden apropiado para la realización de un grupo de tareas o de trabajos [2] y definir los tiempos en que deben iniciarse sus operaciones diferentes con el propósito de lograr un valor óptimo o adecuado de una medida o índice de desempeño, dado un conjunto de recursos, por lo general limitados.

Resolver un problema de secuenciación es proponer un programa o programas donde se pueden observar tanto los tiempos de inicio como los de terminación de las diferentes operaciones relacionadas con las tareas asociadas y con los valores de la medida o medidas de desempeño establecidas como criterio de calidad [6].

B. Elementos constitutivos de la Programación Secuencial

1. Ambientes de producción

Incluye las definiciones como operación, trabajo, tarea, orden de operaciones, secuencia tecnológica y ruta; y los factores que están relacionados con el conjunto de recursos, la naturaleza de las tareas, secuencia tecnológica, patrón de llegada de las tareas(que puede ser aleatorio, constante, variable), disciplinas, y criterios de evaluación [6].

2. Modelo de un proceso de taller

Para este artículo se considera la siguiente clasificación:

- Taller de una sola máquina.
- Taller de máquinas en paralelo: para máquinas idénticas, máquinas uniformes y máquinas no relacionadas.
- Talleres con patrón de flujo uniforme: pueden considerar estructuras de flow shop básico, flow shop flexible, línea de flujo y línea de flujo flexible.
- Talleres con patrón de flujo no uniforme: contienen modelos de job shop, open shop, flexible job shop y flexible open shop [2].

3. Parámetros y variables que intervienen en la formulación del problema.

Los parámetros pueden estar asociados al taller (se refieren a información sobre máquinas y estaciones, sistemas de almacenamiento y capacidades, equipos de transferencia de piezas y productos); relacionados con el conjunto de tareas, datos sobre las operaciones (tiempos de procesamiento y secuencia tecnológica) y tiempos o costos de preparación [9].

Las variables que por lo general están asociadas a la solución del problema están referidas al tiempo de iniciación y terminación de operaciones (C_{ij}) y tiempos de partida (D_{ij})

4. Restricciones fenómenos que afectan al ejecución del problema.

Pueden citarse: Interrupción de operaciones (prmp), restricciones de precedencia (prec), recirculación (recrc), salida de servicio de la máquina (brkdwn), elegibilidad de máquina ($M(j)$), orden de entrada de las tareas igual en todas las máquinas o estaciones (prmu), bloqueo de máquina (block), falta de trabajo en la máquina (starv), programación sin tiempo de espera en proceso (nwt) [7].

5. Objetivos propuestos.

Existen varios objetivos en la programación secuencial entre los que se pueden nombrar: reducción del inventario en proceso, porcentaje de utilización del equipo, cumplimiento de los plazos de entrega entre otros.

C. Medidas de desempeño

Se pueden agrupar según la siguiente clasificación:

- Medidas de desempeño relacionadas con las tareas.
- Medidas de desempeño asociadas a las máquinas.
- Medidas de desempeño basadas en los costos de inventario y en los costos de utilización del equipo.
- Medidas de desempeño regular
- Medidas de desempeño equivalentes.

El documento está estructurado de la siguiente forma: enunciado del problema, organización de datos en una tabla, secuencia tecnológica, ruta de cada tarea, grafo que representa el proceso productivo, lista de precedencias, programa de ejecución factible, aplicación de Matlab para resolver el problema y la presentación de la propuesta de mejoramiento a partir del fraccionamiento razonable de operaciones.

III. CONTENIDO

El caso de estudio se describe en los siguientes apartes:

A. Enunciado del problema.

Dado el siguiente problema de secuenciación, determinar un programa de producción que sea factible, teniendo en cuenta las relaciones de precedencia y las restricciones impuestas.

B. Organización de los datos en una tabla.

Un grupo de 5 tareas J1, J2, J3, J4, J5 se va programar en un taller de 4 máquinas según la información y restricciones de la siguiente tabla: (ver tabla 1)

	j	J1	J2	J3	J4	J5	
m	$\mu \setminus r^2$	0	10	6	15	18	
1	0	1.1		3.1	4.1	5.1	$4.1 \leq 5.1$
		3 24		8 24	5 35	6 30	
2	3	1.2	2.1	3.2		5.2	$3.2 \leq 5.2$
		4 32	12 48	5 15		10 50	
3	14	1.3	2.2	3.3	4.2	5.3	
		5 40	5 20	4 12	8 56	5 25	
4	20		2.3	3.4	4.3	5.4	
			3 12	7 21	6 42	8 40	
	$d_{(j)}$	140	180	250	280	200	

Precedencias adicionales: $1.1 \leq 3.3$

Tamaño de los lotes: $Q_1=8; Q_2=4; Q_3=3; Q_4=7; Q_5=5;$

Tabla 1. Información de las tareas y restricciones.

C. Secuencia Tecnológica.

A continuación se muestra la secuencia tecnológica de cada tarea:(ver figura 1)

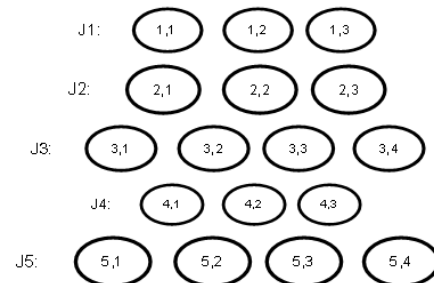


Figura 1. Secuencia tecnológica.

D. Ruta de cada tarea:

Se puede observar en la figura 2.

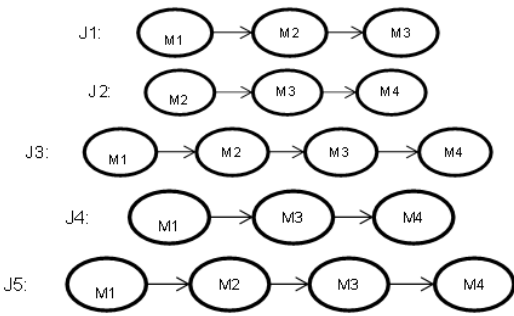


Figura 2. Ruta de cada tarea.

E. Grafo que representa el proceso productivo. (ver figura 3)

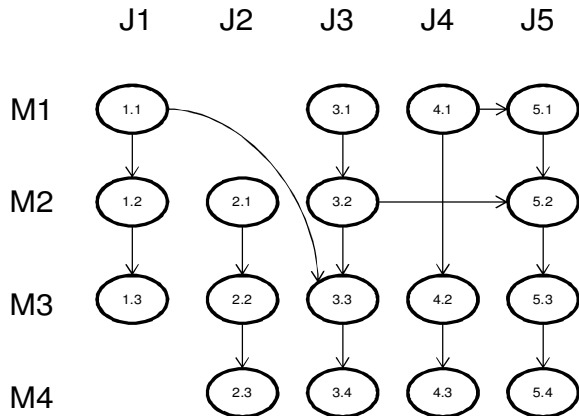


Figura 3. Grafo que representa el proceso.

F. Lista de Precedencias.

La lista se muestra a continuación:

1.1		
1.2	1.1	
1.3	1.2	
2.1		
2.2	2.1	
2.3	2.2	
3.1		
3.2	3.1	1.1
3.3	3.2	
3.4	3.3	
4.1		
4.2	4.1	
4.3	4.2	
5.1		4.1
5.2	5.1	3.2
5.3	5.2	
5.4	5.3	

G. Programa de Ejecución Factible.

Se forma a partir de la lista de precedencias, iniciando con las operaciones que no tienen restricciones, así: (ver tabla 2) [8].

Pasos	Operaciones programables					Selección
	1	2	3	4	5	
1	1.1 0/24	2.1 10/58	3.1 26/30	4.1 15/50		1.1
2	1.2 24/56	2.1 10/58	3.1 24/48	4.1 24/59		2.1
3	1.2 58/90	2.2 58/78	3.1 24/48	4.1 24/59		4.1
4	1.2 58/90	2.2 58/78	3.1 59/83	4.2 59/115	5.1 59/89	1.2
5	1.3 90/130	2.2 58/78	3.1 59/83	4.2 59/115	5.1 59/89	2.2
6	1.3 90/130	2.3 78/90	3.1 59/83	4.2 78/134	5.1 59/89	5.1
7	1.3 90/130	2.3 78/90	3.1 89/113	4.2 78/134		2.3
8	1.3 90/130	3.1 89/113		4.2 78/134		4.2
9	1.3 134/174	3.1 89/113		4.3 134/176		3.1
10	1.3 134/174	3.2 113/128		4.3 134/176		3.2
11	1.3 134/174	3.3 113/128	4.3 134/176	5.2 128/178		5.2
12	1.3 134/174	3.3 134/146	4.3 134/176	5.3 178/228		1.3
13	3.3 174/186	4.3 134/176	5.3 178/203			4.3
14	3.3 174/186		5.3 178/203			3.3
15	3.4 186/207		5.3 186/211			5.3
16	3.4 186/207		5.4 211/251			3.4
17			5.4 211/251			5.4

Tabla 2. Relación de pasos y operaciones programables.

Los criterios de decisión para programación de operaciones son:

Se inicia con la operación que presente menor tiempo de iniciación.

En caso de empate, se tienen en cuenta en primera instancia las restricciones de precedencia y luego las fechas de entrega. Se aplica el algoritmo hasta terminar de programar todas las operaciones.

Por lo anterior, la selección queda así:

1.1	Por menor tiempo de iniciación.	3.2	Por menor tiempo de iniciación.
2.1	Por menor tiempo de iniciación.	5.2	Por menor tiempo de iniciación.
4.1	Por restricción de precedencia.	1.3	Por menor tiempo de entrega.
1.2	Por menor tiempo de entrega.	4.3	Por menor tiempo de iniciación.
2.2	Por menor tiempo de iniciación.	3.3	Por menor tiempo de iniciación.
5.1	Por menor tiempo de entrega.	5.3	Por menor tiempo de entrega.
2.3	Por menor tiempo de entrega.	3.4	Por menor tiempo de iniciación.
4.2	Por menor tiempo de iniciación.	5.4	Por menor tiempo de iniciación.
3.1	Por menor tiempo de iniciación.		

Con esta información, el programa de producción es: (ver tabla 3).

	j	J1	J2	J3	J4	J5	
m	i	0	10	6	15	18	
1	0	1.1 0/24 3 24		3.1 89/113 8 24	4.1 24/59 5 35	5.1 59/89 6 30	4.1<=5.1
2	3	1.2 58/90 4 32	2.1 10/58 12 48	3.2 113/128 5 15		5.2 128/178 10 50	3.2<=5.2
3	14	1.3 134/174 5 40	2.2 58/78 5 20	3.3 174/186 4 12	4.2 78/134 8 56	5.3 186/211 5 25	
4	20		2.3 78/90 3 12	3.4 186/207 7 21	4.3 134/176 6 42	5.4 211/251 8 40	
40		140	180	250	280	200	

Tabla 3. Programa de ejecución factible.

H. Aplicación en MATLAB para resolver el problema.

Se diseñó el código en Matlab que permitió encontrar la solución del algoritmo [10].

Para ello, se establecieron varios archivos. Uno es el que se muestra a continuación relacionado con la tabla de salida de resultados (es el script y código que constituye la tabla Z, que permite organizar la salida de la información, obtenida desde el principal), como se muestra a continuación: (ver figura 4)

```

Z(i,1) es el trabajo p
Z(i,2) es la tarea q del trabajo p
Z(i,3) es el trabajo p' tal que la
la tarea q' restringe a p.q
Z(i,4) es la tarea q' tal p'.q'
restringe a p.q
Z(i,5) es el trabajo p'' tal que la
tarea q'' restringe a p.q
Z(i,6) es la tarea q'' tal que
p''.q'' restringe a p.q
Z(i,7) es la maquina m donde se
realiza la tarea p.q
Z(i,8) es el tiempo del proceso
de los insumos en la
maquina m, coorespondiente a p.q
Z(i,9) es el tiempo de inicio del
proceso en la maquina m
Z(i,10) es tiempo de finalizacion
del proceso en la maquina m
Z(i,11) es la bandera de control
1 si no se ha procesado,
0 se realizo el proceso.
    
```

Z(i,1)=	C1
Z(i,2)=	C2
Z(i,3)=	C3
Z(i,4)=	C4
Z(i,5)=	C5
Z(i,6)=	C6
Z(i,7)=	C7
Z(i,8)=	C8
Z(i,9)=	C9
Z(i,10)=	C10
Z(i,11)=	C11

Figura 4. Información para identificar las columnas de salida de MATLAB.

La última iteración muestra el siguiente resultado: Ver tabla 4.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
1	1	1	0	0	0	0	1	24	0	24	0
1	2	0	0	0	0	0	2	32	58	90	0
1	3	0	0	0	0	0	3	40	134	174	0
2	1	0	0	0	0	0	2	48	10	58	0
2	2	0	0	0	0	0	3	20	58	78	0
2	3	0	0	0	0	0	4	12	78	90	0
3	1	0	0	0	0	0	1	24	89	113	0
3	2	0	0	0	0	0	2	15	113	128	0
3	3	0	0	0	0	0	3	12	174	186	0
3	4	0	0	0	0	0	4	21	186	207	0
4	1	0	0	0	0	0	1	35	24	59	0
4	2	0	0	0	0	0	3	56	78	134	0
4	3	0	0	0	0	0	4	42	134	176	0
5	1	0	0	0	0	0	1	30	59	89	0
5	2	0	0	0	0	0	2	50	128	178	0
5	3	0	0	0	0	0	3	25	186	211	0
5	4	0	0	0	0	0	4	40	211	251	0

Tabla 4. Iteración final del Programa de ejecución factible

Las columnas C9 y C10 indican las fechas de inicio y terminación respectivamente de cada una de las actividades de los diferentes trabajos que se han procesado, que coinciden con la heurística aplicada manualmente [12].

El diagrama Gantt para este problema es:

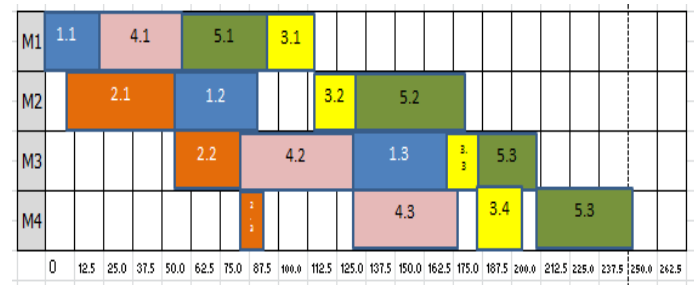


Figura 5. Diagrama Gantt del problema resuelto.

El programa termina en 251 unidades de tiempo.

I. Propuesta de mejoramiento a partir del fraccionamiento razonable de operaciones.

En el sistema estudiado (ver diagrama Gantt) se observa mucho tiempo ocioso, algunas veces forzado por secuencia tecnológica y otras por organización no adecuada de operaciones. A continuación se describe su cálculo:

- En la máquina M1: no se presenta tiempo ocioso.
- En la máquina M2: el tiempo ocioso por secuencia tecnológica se presenta entre las operaciones 3,2 y 1,2 y corresponde a 113-90 = 23 unidades de tiempo.
- En la máquina M3: entre las operaciones 2,2 y 2,1 hay un tiempo ocioso forzado por secuencia tecnológica: 58-10= 48 unidades de tiempo.

- En la M4:

Por secuencia tecnológica se presenta tiempo ocioso entre las operaciones 2,1 y 2,2 con la operación 2,3: $78-10=68$ unidades de tiempo.

También por secuencia tecnológica, entre las operaciones 2,3 y 4,3 el tiempo ocioso es: $134-90=44$ unidades de tiempo.

Entre las operaciones 4,3 y 3,4 el tiempo ocioso por secuencia tecnológica es: $186-176=10$ unidades de tiempo.

Por secuencia tecnológica se presenta tiempo ocioso entre las operaciones 3,4 y 5,3: $207-186=21$ unidades de tiempo.

El tiempo ocioso total es: $23+48+68+44+10+21$ unidades de tiempo = 214 unidades de tiempo.

La propuesta de mejoramiento consiste en que a partir de un "fraccionamiento adecuado" de operaciones se pueda disminuir este tiempo ocioso y por lo tanto el makespan también obtendrá un mejor desempeño.

Las operaciones que se pueden fraccionar son: 2.1, 2.2 y 2.3; 3.1, 3.2, 3.3, y 3.4; 4.2 y 4.3; 5.2, 5.3 y 5.4.

Una estrategia inicial es considerar el tamaño de los lotes para cada trabajo J ($j=1, 2, 3, 4, 5$)

Por ejemplo, el trabajo J2 tiene las operaciones 2.1, 2.2 y 2.3 y el tamaño del lote es de $Q_2=4$ unidades.

Cada operación se fracciona en 4 partes:

- 2.1a, 2.1b, 2.1c, 2.1d: para cada una el tiempo de procesamiento es de 12 unidades de tiempo.
- 2.2a, 2.2b, 2.2c, 2.2d: para cada operación el tiempo de proceso es de 5 unidades de tiempo.
- 2.3a, 2.3b, 2.3c, 2.3d: para cada operación el tiempo de proceso establecido es de 3 unidades de tiempo.

El método propuesto consiste en que la operación 2.2a (en M3) se inicia tan pronto termine la operación 2.1a (en M2); la operación 2.2b (en M3) comienza una vez termine la operación 2.1b (en M2); y la operación 2.2c (en M3) se inicia cuando termina la operación 2.1c (en M2) y la operación 2.2d (en M3) sólo comienza cuando termine la operación 2.1d (en M2).

En forma similar, la operación 2.3a (en M4) se inicia cuando termine la operación 2.2a (en M3) y así sucesivamente hasta que la operación 2.3d (en M4) comience una vez haya terminado la operación 2.2d (en M3).

El mismo método se aplica para las operaciones del trabajo J3 (3.1, 3.2, 3.3 y 3.4) que se fracciona en las operaciones 3.1a, 3.1b, 3.1c; 3.2a, 3.2b, 3.2c; 3.3a, 3.3b, 3.3c; 3.4a, 3.4b, 3.4c.

El tiempo de proceso para las operaciones fraccionadas del trabajo 3 es:

- Para 3.1a, 3.1b, 3.1c: 8 unidades de tiempo para cada una.
- Para 3.2a, 3.2b, 3.2c: 5 unidades de tiempo para cada una.
- Para 3.3a, 3.3b, 3.3c: 4 unidades de tiempo para cada una.
- Para 3.4a, 3.4b, 3.4c: 7 unidades de tiempo para cada una.

El trabajo J4, compuesto por las operaciones 4.1, 4.2 y 4.3 se fracciona en las siguientes operaciones: (4.2a, 4.2b, 4.2c, 4.2d, 4.2e, 4.2d; 4.2e, 4.2f, 4.2g); (4.3a, 4.3b, 4.3c, 4.3d, 4.3e, 4.3d; 4.3e, 4.3f, 4.3g). La operación 4.1 no necesita fraccionarse.

Por otra parte, el trabajo J5, compuesto por las operaciones 5.1, 5.2, 5.3 y 5.4 se fracciona en las siguientes operaciones: (5.2a, 5.2b, 5.2c, 5.2d, 5.2e); (5.3a, 5.3b, 5.3c, 5.3d, 5.3e); (5.4a, 5.4b, 5.4c, 5.4d, 5.4e). La operación 5.1 tampoco necesita fraccionarse.

A continuación se muestran los nuevos tiempos de iniciación y de terminación para las operaciones fraccionadas con este método.

Para el trabajo J2 la nueva programación de sus operaciones es:

2.1a	2.1b	2.1c	2.1d
10/22	22/34	34/46	46/58

El tiempo de proceso para cada operación es de 12 unidades de tiempo.

2.2a	2.2b	2.2c	2.2d
22/27	34/39	46/51	58/63

El tiempo de proceso para cada una de las anteriores operaciones es de 5 unidades de tiempo.

2.3a	2.3b	2.3c	2.3d
27/30	39/42	51/54	63/66

El tiempo de proceso para cada una de las anteriores operaciones es de 3 unidades de tiempo.

Con el nuevo método el trabajo J2 inicia en 10 unidades de tiempo y termina en 66 unidades de tiempo. Se nota un

adelanto de 24 unidades de tiempo (90-66) que corresponde a la disminución de tiempo ocioso con la nueva programación. Para el trabajo J3, la nueva programación es:

3.1a	3.1b	3.1c
89/97	97/105	105/113

El tiempo de proceso para cada operación es de 8 unidades de tiempo.

3.2a	3.2b	3.2c
97/102	105/110	113/118

El tiempo de proceso para cada una de las anteriores operaciones es de 5 unidades de tiempo.

La operación 3.3a comienza una vez termine la operación 1.3 de la maquina 3. Como la operación 4.2g termina en 119 unidades de tiempo, en seguida se hace la operación 1.3, cuyo tiempo de proceso es de 40 unidades de tiempo terminando en 159 unidades de tiempo.

La programación de 3.3 fraccionada con inicio en 3.3a con 159 unidades de tiempo es:

3.3a	3.3b	3.3c
159/163	163/167	167/171

El tiempo de proceso para cada operación es de 4 unidades de tiempo.

En seguida se programan las operaciones 3.4a, 3.4b y 3.4c que tienen una duración de 7 unidades de tiempo, iniciando en 163.

3.4a	3.4b	3.4c
163/170	170/177	177/184

Para el trabajo J4, la operación 4.2a comienza una vez terminada la actividad 2.2d en la máquina 3:

4.2a	4.2b	4.2c	4.2d	4.2e	4.2f	4.2g
63/71	71/79	79/87	87/95	95/103	103/111	111/119

El tiempo de proceso para 4.2a, 4.2b, 4.2c, 4.2d, 4.2e, 4.2f y 4.2g es de 8 unidades de tiempo

La programación de las operaciones 4.3a, 4.3b, 4.3c, 4.3d, 4.3e, 4.3f y 4.3g cuyo tiempo de proceso es de 6 unidades de tiempo es:

4.3a	4.3b	4.3c	4.3d	4.3e	4.3f	4.3g
71/77	79/85	87/93	95/101	103/109	111/117	119/125

Finalmente se programa las operaciones 5.2, 5.3 y 5.4 del trabajo 5 con sus correspondientes fraccionamientos.

Téngase en cuenta que la operación 5.2a comienza después de la operación 3.2c y así sucesivamente:

5.2a	5.2b	5.2c	5.2d	5.2e
118/128	128/138	138/148	148/158	158/168

El tiempo de proceso para cada operación es de 10 unidades de tiempo.

La operación 5.3a se inicia después de la operación 3.3c y así sucesivamente.

La programación de las operaciones 5.3a, 5.3b, 5.3c, 5.3d, 5.3e es:

5.3a	5.3b	5.3c	5.3d	5.3e
171/176	176/181	181/186	186/191	191/196

El tiempo de proceso para cada operación es de 5 unidades de tiempo.

Las operaciones 5.4a, 5.4b, 5.4c, 5.4d y 5.4e se programan después de la operación 3.4c y tiene un tiempo de proceso de 8 unidades de tiempo por cada una.

Su programación es:

5.4a	5.4b	5.4c	5.4d	5.4e
176/184	184/192	192/200	200/208	208/216

Como puede apreciarse el nuevo makespan con el modelo propuesto es 216 unidades de tiempo y el makespan inicial fue de 251 unidades de tiempo.

El programa se ha disminuido en 35 unidades de tiempo, que representa una notable mejoría en la parte económica, que se puede determinar si se conocen los valores de la hora de mano de obra y el valor de la hora – máquina, entre otros elementos del costo de producción.

A continuación se muestra el nuevo diagrama Gantt con las operaciones fraccionadas (ver figura 6)

Diagrama Gantt mejorado

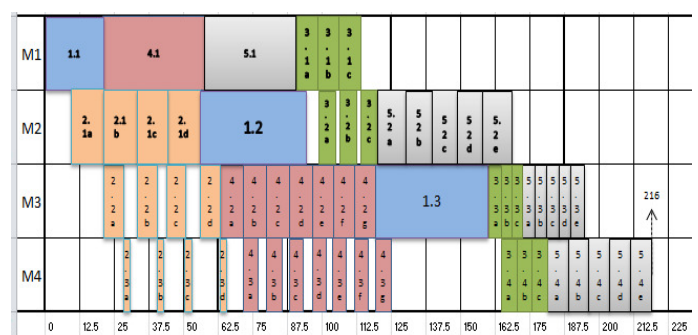


Figura 6. Diagrama Gantt del problema resuelto con fraccionamiento de operaciones

IV. CONCLUSIONES

- Existen múltiples formas para resolver problemas de secuenciación, que van desde el uso de heurísticas, Metaheurísticas y aplicación de modelos matemáticos, que permiten hacer la asignación de la forma más adecuada.
- Con la heurística constructiva aplicada en este caso, se obtuvo un buen resultado, que fue validado con la aplicación del código en Matlab.
- Es importante considerar que la implementación de algoritmos para resolver esta clase de problemas debe tener en cuenta la cantidad de operaciones por trabajo, el número de máquinas y las respectivas restricciones tecnológicas, que se pueden complicar por limitados recursos computacionales de los equipos.
- El tema expuesto en este artículo es un punto de partida para comprender la importancia de la adecuada administración de los sistemas de producción en sus diferentes contextos de operación y se entenderá por qué el control es significativo para que las empresas tengan fortalezas en sus capacidades competitivas.
- Una adecuada asignación en la programación secuencial se puede convertir en una ventaja competitiva de las organizaciones en sus intenciones de atender con respuesta rápida los requerimientos de los clientes.

V. RECOMENDACIONES.

Como valor agregado que pueda complementar y mejorar los resultados de este algoritmo, se sugiere aplicar el fraccionamiento de las operaciones hasta donde sea razonable, con el propósito de disminuir el tiempo ocioso que se presenta en el programa de producción obtenido en este caso.

Las conclusiones son obligatorias y deben ser claras. Deben expresar el balance final de la investigación o la aplicación del conocimiento.

REFERENCIAS

- [1] M.L. Pinedo. *Scheduling: Theory, Algorithms and Systems*. New York: Spring, 2008, p. 1.
- [2] M. Pinedo, X. Chao. *Operations Scheduling with Applications in manufacturing and Services*. Nueva York: McGraw Hill/Irwin, 1999.
- [3] M. Hoyos. *Curso de programación secuencial*. Notas de clase, Universidad tecnológica de Pereira, 2011.
- [4] H. Noori, R. Radford. *Administración de Operaciones y Producción: calidad total y respuesta sensible rápida*. Bogotá, Colombia, Mc Graw Hill, 2009, p. 523 y siguientes.

- [5] P.I. Roder, *Finite Scheduling Systems*, APICS, August 1993, pp.40-42
- [6] J. Martin, *Experts Systems Solve a Scheduling Problem for Carpenter Technology Systems*, APICS, November, 1992 pp.37-39
- [7] B. Render, M. Stair y M. Hanna. *Quantitative Analysis for Management*. 8a. ed. Upper Saddle River, New York: Prentice – Hall (2003).
- [8] J.R. Olson, M.J. Schniederjans. *A Heuristic Scheduling System for Ceramic Industrial Coatings*. Interfaces 30, núm.5 (septiembre –octubre de 2000)
- [9] S.F. Bolander y S.G. Taylor. *Scheduling Techniques: A Comparison of Logic*. Production and Inventory Management Journal (primer trimestre de 2000):1-5. K.R. Baker and J. C. Smith (2003). *A Multiple-Criterion Model for Machine Scheduling*. Journal of Scheduling, Vol. 6, pp. 7–16.
- [10] P. Brucker (2004) *Scheduling Algorithms* (Fourth Edition), Springer Verlag Berlin.
- [11] Z.L. Chen and G. Pundoor (2006). *Order Assignment and Scheduling in a Supply Chain*. Operations Research, Vol. 54, pp. 555-572.
- [12] Z.L. Chen and G. Vairaktarakis (2005). *Integrated Scheduling of Production and Distribution Operations*. Management Science, Vol. 51, pp. 614-628.