

APLICACIÓN DE LA HEURÍSTICA DE DANNENBRING EN LA SECUENCIACIÓN DE N TAREAS EN M MÁQUINAS: UN CASO DE ESTUDIO

APPLICATION DANNENBRING'S HEURISTIC FOR SEQUENCING N JOBS THROUGH M MACHINES: A STUDY CASE.

RESUMEN

Este documento presenta como es aplicada la heurística de Dannenbring para resolver un problema de programación de n tareas a través de m máquinas. El documento hace de forma breve una introducción al problema del Flow Shop, el modelo matemático, los pasos para resolver el problema, la solución con base en una medida de desempeño y las conclusiones.

JORGE HERNÁN RESTREPO C

Ingeniero Industrial, M. Sc.
Profesor Asociado
Universidad Tecnológica de Pereira
jhrestrepoco@utp.edu.co

PALABRAS CLAVES: Heurística, programación de tareas.

ABSTRACT

This paper shows how is applied the Dannenbring's heuristic to solve a programming problem of n jobs through m machines. The document explains a short introduction of Flow shop problem, the mathematical model, the steps to solve the problem, problem solution and conclusions.

KEYWORDS: Heuristic, jobs programming

1. INTRODUCCIÓN¹

Los modelos de secuenciales tienen aplicaciones principalmente en un taller de tareas, donde un conjunto de máquinas, de propósito general, ejecutan una serie de operaciones sobre órdenes de trabajos o tareas de producción. Las tareas son a menudo únicas y ordenadas por un determinado cliente. El proceso en el taller de tareas es un modelo fundamental para un considerable número de sistemas operacionales, tales como las actividades de mantenimiento, asignación de aulas de clase a un grupo de materias, la programación de llegada y salida de buses, la programación de n tareas en m máquinas en un taller de fabricación intermitente, etc.

La programación de un taller de tareas consiste en determinar el orden o la secuencia de las tareas en las máquinas para optimizar alguna medida de ejecución.

Existen cuatro factores que describen y clasifican un problema específico de programación de un taller de tareas de acuerdo a:

1. El patrón de llegada de los trabajos: si n tareas llegan simultáneamente al taller y quedan disponibles para iniciar su proceso tendremos un

problema de programación estática. Si las tareas llegan intermitentemente, posiblemente de acuerdo a un proceso estocástico, el problema de programación es dinámico.

2. El número de máquinas que integran el taller.: Existe un problema de secuenciación cuando n trabajos son programados en m máquinas.
3. El flujo de producción: el flujo de proceso de las tareas a través de las máquinas debe ser especificado, si todas las tareas siguen la misma ruta el flujo de producción es continuo o en serie. En el extremo opuesto, donde no existe una ruta preconcebida de procesos se tiene un taller cuyo flujo de producción es aleatorio. Los trabajos pueden ser independientes unos de otros, o bien interdependientes. Cuando se mezclan los diferentes tipos de flujos de producción, los de serie con los aleatorios, existen rutas generales de proceso.
4. El objetivo que se desea optimizar: la medida de desempeño que frecuentemente se utiliza es la optimización del tiempo total de proceso de todas las tareas o trabajos en todas las máquinas, pero se puede pensar también en la tardanza máxima, tardanza promedio o mínimo número de trabajos tardíos entre otras.

¹ Ospina Bolaños Dagoberto, Sistemas Administrativos de Producción y Operaciones, Programación Secuencial, Editorial UTP 1996, página 231.

2. REGLAS Y NOTACIÓN²

En todos los problemas de programación considerados en número de tareas y máquinas son finitos. En número de trabajos es denotado por n y el número de máquinas por m . normalmente j se refiere a la tarea e i para la máquina. Si una tarea requiere un número de pasos de proceso u operaciones, entonces el par (i,j) significa el paso del proceso del trabajo j en la máquina i . las siguientes piezas de datos son asociadas con el trabajo j .

1. Tiempo de proceso $p_{i,j}$: representa el tiempo de proceso del trabajo j en la máquina i .
2. Fecha de llegada r_j : esto significa cuando el trabajo llega del trabajo j al sistema.
3. Fecha de entrega d_j : Fecha prometida para la entrega del trabajo j .
4. Importancia o peso w_j : Es básicamente el factor de prioridad, denotando la importancia del trabajo j con relación a los otros trabajos

Un problema de programación es descrito por la tripleta $\alpha/\beta/\gamma$. Donde:

α : Este campo contiene el ambiente de la máquina

β : En este campo contiene las características del proceso y las restricciones.

γ : En este campo contiene el objetivo a ser minimizado.

3. EL PROBLEMA DEL FLOW SHOP (Fm)³

Hay m máquinas en serie. Cada trabajo tiene que ser procesado en cada una de las m máquinas. Cada trabajo tiene que seguir la misma ruta. Por ejemplo, primero en la máquina 1, luego en la máquina 2 y así sucesivamente. Después de la terminación en una máquina, un trabajo es unido a la cola de la siguiente máquina. Usualmente todas las colas son asumidas con la disciplina de primero en llegar primero en ser atendido FIFO. Si la disciplina FIFO esta en el efecto del Flow Shop, es referida como una permutación Flow Shop y el campo B incluye la entrada *pmu*.

4. HEURÍSTICA DE DANNENBRING.⁴

² M.L Pinedo, Scheduling: theory, algorithms and systems. Editorial Springer, tercera edición, Página 13.

³ M.L Pinedo, Scheduling: theory, algorithms and systems. Editorial Springer, tercera edición, Página 15

En las pasadas tres décadas, extensivas investigaciones han sido hechas sobre el problema del Flow Shop. Pero no hay algoritmos que provean una fácil solución óptima. Las técnicas de programación entera y el branch and bound pueden ser usadas para encontrar una óptima solución. Sin embargo, ellos no son efectivos en problemas grandes o igual en problemas medianos. El problema del Flow Shop ha sido presentado verdaderamente como un problema NP completo. Por esta razón, muchas heurísticas han sido desarrolladas para entregar una solución muy buena y de forma rápida.

Dannenbring diseño un método el cual trata de combinar las ventajas de las heurísticas de Palmer y CDS. Su idea es construir dos(2) máquinas en las que pueda aplicar el algoritmo de Johnson, pero que la consecución de los tiempos de proceso reflejen el mismo comportamiento como el índice máximo de Palmer.

Los tiempos de proceso resultantes para la primera y segunda máquina son:⁵

Para la máquina 1:

$$a_j = \sum_{i=1}^m [m - i + 1] P_{i,j}$$

Para la máquina 2:

$$b_j = \sum_{i=1}^m i P_{i,j}$$

Luego se resuelve el problema con Johnson

5. PROBLEMA

Para mostrar el funcionamiento de la heurística de Dannenbring, planteamos un problema de 10 tareas en 5 máquinas. El objetivo es minimizar el tiempo de terminación o $C_{máx}$.

	MÁQUINA				
TAREA	1	2	3	4	5

⁴ Mitsuo Gen, Runwei Cheng, Genetic algorithms and engineering design, Página 178

⁵ Simon French, Sequencing and scheduling: an introduction to the mathematics of the job-shop, editorial Ellis Horwood Limited, página 170.

1	3	7	3	3	2
2	10	4	9	9	8
3	7	6	3	1	10
4	2	3	1	7	1
5	3	2	4	2	4
6	10	8	7	10	8
7	9	1	10	4	4
8	10	5	8	1	5
9	8	2	9	4	1
10	6	1	7	4	4

Resolviendo con Johnson:

M1		M2	
a1=	60	b1=	48
a2=	119	b2=	121
a3=	80	b3=	82
a4=	40	b4=	44
a5=	43	b5=	47
a6=	131	b6=	127
a7=	91	b7=	77
a8=	101	b8=	73
a9=	84	b9=	60
a10=	67	b10=	65

Aplicando la heurística tenemos:

$$a_j = \sum_{i=1}^m [m - i + 1] P_{i,j}$$

a1=	60
a2=	119
a3=	80
a4=	40
a5=	43
a6=	131
a7=	91
a8=	101
a9=	84
a10=	67

$$b_j = \sum_{i=1}^m i P_{i,j}$$

b1=	48
b2=	121
b3=	82
b4=	44
b5=	47
b6=	127
b7=	77
b8=	73
b9=	60
b10=	65

El algoritmo de Johnson⁶ se puede expresar así:

Sea k y l dos contadores. k=1 inicialmente y creciendo a 2, 3,4,... como las 1era, 2da, 3era, 4ta,.. Posiciones en la secuencia de proceso que están completadas. Similarmente, l=n tareas inicialmente y decreciendo (n-1), (n-2),..... como las (n-1), (n-2),.....posiciones en la secuencia de proceso que están completadas.

Paso 1: presente a k=1, l=n

Paso 2: presente la lista de las tareas o trabajos no programados.

Paso 3: encuentre el menor de los tiempos a_j y b_j para las tareas no programadas.

Paso 4: si el menor tiempo es para el trabajo J_i ($i=1,2,..n$ tareas) en la primera máquina. Por ejemplo a_j es el menor, entonces:

1. Programe J_i en la k e cima posición de la secuencia de procesamiento.
2. Borre J_i de la lista de las tareas no programadas.
3. Incremente k a k+1.
4. Vaya al paso 6.

Paso 5: si el menor tiempo es para el trabajo J_i en la segunda máquina. Por ejemplo b_j es el menor, entonces:

1. Programe J_i en la l e cima posición de la secuencia de procesamiento.
2. Borre J_i de la lista de las tareas no programadas.
3. reduzca l a (l-1).
4. Vaya al paso 6.

Paso 6: sí hay algún trabajo sin programar vaya al paso 3. De lo contrario pare.

⁶ Simon French, Sequencing and scheduling: an introduction to the mathematics of the job-shop, editorial Ellis Horwood Limited, página 69.

Nota: si sí el menor tiempo ocurre para mas de un trabajo en el paso 3, tome J_i arbitrariamente.

La secuencia encontrada aplicando el algoritmo de Johnson es:

Secuencia
4
5
3
2
6
7
8
10
9
1

Por lo tanto, computando la programación se tiene el siguiente valor de C_{max} :

TAREA	MÁQUINA				
	1	2	3	4	5
4	2	5	6	13	14
5	5	7	11	15	19
3	12	18	21	22	32
2	22	26	36	46	54
6	32	40	47	57	65
7	41	42	57	61	69
8	51	56	65	66	74
10	57	58	72	76	80
9	65	67	81	85	86
1	68	75	84	88	70

$C_{max} = 70$

6. CONCLUSIONES Y RECOMENDACIONES

- La heurística es una herramienta amigable, que permite encontrar rápidamente soluciones sin realizar exhaustivos cálculos.
- En la práctica se requieren de herramientas que sean fáciles de entender y manipular. Día a día se requieren hacer programaciones y

reprogramaciones de trabajos, donde el cliente requiere oportunamente una promesa de

7. BIBLIOGRAFÍA.

- [1]. Ospina Bolaños Dagoberto, Sistemas Administrativos de Producción y Operaciones, Programación Secuencial, Editorial UTP 1996, página 231.
- [2]. M.L Pinedo, Scheduling: theory, algorithms and systems. Editorial Springer, tercera edición, página 13.
- [3]. M.L Pinedo, Scheduling: theory, algorithms and systems. Editorial Springer, tercera edición, página 15.
- [4]. Mitsuo Gen, Runwei Cheng, Genetic algorithms and engineering design, página 178.
- [5]. Simon French, Sequencing and scheduling: an introduction to the mathematics of the job-shop, editorial Ellis Horwood Limited, página 170.
- [6]. Simon French, Sequencing and scheduling: an introduction to the mathematics of the job-shop, editorial Ellis Horwood Limited, página 69.