

Análisis Comparativo de Aproximaciones Metaheurísticas en los Problemas de JSS

Luis G. Rodríguez Marengo¹ Juan Carlos Cabarcas² Julio Mario Daza-Escorcía³

¹ Universidad Autónoma del Caribe, Barranquilla, Colombia, luig1987@hotmail.com

² Universidad Autónoma del Caribe, Barranquilla, Colombia, juan_cabarcas@yahoo.es

³ Universidad Autónoma del Caribe, Barranquilla, Colombia, juliomariodaza@hotmail.com

The problem of Job Shop Scheduling, is a very complex model in the world, and is because of his applications that is the object of a hundreds of studies around the world. In this investigation, it is trying to archive a very close approximation of a bunch of different metaheuristics (Ant Q, Tabu Search, GRASP and Genetic Algorithms) and analyse their sensibility around minimize $C_{\max}$. The Job Shop Scheduling (JSS) is a family paradigm for combinatorial optimization problems and satisfaction restrictions. These have been studied in recent decades due to its complexity (which is on the order of NP-Hard) and the large number of applications that can be applied in everyday life. Usually tasks programming in general operations include assigning a series of resources to a variety of jobs, satisfying some constraints and one or more optimization criteria, as can be to improve the flow time $C_{\max}$. What we wanted to make this research is to establish a parallel four different metaheuristics, which are, ANT-Q, Tabu Search, GRASP and Genetic Algorithms in order to seek performance measures, which you know best performance to meet the objective function as putting jss minimize $C_{\max}$ or flow time.

INTRODUCCIÓN

El problema de programación de operaciones en talleres de máquinas intermitentes, también conocido en la comunidad científica como Job Shop Scheduling (JSS) es un paradigma de la familia de problemas de optimización combinatoria y de satisfacción de restricciones. Estos han sido objeto de estudio durante las últimas décadas debido a su complejidad (que va en el orden del NP-Hard) y la gran cantidad de aplicaciones en las que se puede aplicar en la vida diaria. Usualmente las tareas de programación de operaciones en general consisten en asignar una serie de recursos a una serie de trabajos, satisfaciendo unas restricciones y uno o más criterios de optimización, como lo puede ser mejorar el tiempo de flujo $C_{\max}$.

Lo que se busca con la presente investigación es lograr establecer un paralelo entre cuatro diferentes metaheurísticas, las cuales son, Ant-Q, Tabu Search, GRASP y Genetic Algorithms, con el fin de buscar medidas de desempeño, saber cual tiene mejores rendimientos para resolver el JSS poniendo como función objetivo minimizar el tiempo de flujo $C_{\max}$.

CONSIDERACIONES DEL MODELO

El Job Shop Scheduling Problem que se va a tener en consideración es el siguiente:

- De precedencia, es decir, el orden de las operaciones sobre las máquinas es pre-especificado, las operaciones de un trabajo dado tienen que ser procesadas en un orden dado.
- Cada operación usa una de las $|M|$ máquinas por un tiempo determinado.
- Cada máquina puede procesar una operación a la vez, y en cuanto una operación inicia el procesamiento sobre una máquina dada, debe completar el procesamiento sobre esa máquina sin interrupción.
- Ni la fecha de liberación ni la fecha de entrega son especificados.
- Un trabajo no visita una máquina dos veces.
- No hay restricciones de precedencia entre las operaciones de diferentes trabajos.

PLANTEAMIENTO DEL MODELO

Según (Meisel & Prado, 2010), se puede plantear la función objetivo principal del modelo, es minimizar el $C_{\max}$ o tiempo de flujo y esto se redacta de la siguiente manera: Dado un conjunto de M máquinas (donde se define el tamaño de M por $|M|$) y un conjunto de trabajos J (donde el tamaño de J se denota por $|J|$), sea $\sigma_1^j < \sigma_2^j < \dots < \sigma_{|M|}^j$ el conjunto ordenado de $|M|$ operaciones del trabajo J donde $\sigma_k^j < \sigma_{k+1}^j$ indica que la operación σ_{k+1}^j puede ser procesada sólo después que haya sido completada la operación σ_k^j . Sea ϑ el conjunto de operaciones. Cada operación σ_k^j se define por dos parámetros: M_k^j es la máquina la cual σ_k^j es procesada y $p_k^j = p(\sigma_k^j)$ es el tiempo de procesamiento de la operación σ_k^j . Definiendo $t(\sigma_k^j)$ como el tiempo de comienzo de la k -ésima operación $\sigma_k^j \in \vartheta$, el problema de programación de máquinas intermitentes puede formularse como sigue, véase (1)

$$Z_{\min} = C_{\max} \quad (1)$$

RESTRICCION DE TIEMPO

La restricción de tiempo hace referencia a que el tiempo de comienzo de una operación, más el tiempo de procesamiento de las operaciones siguientes, tiene que ser menor o igual que el tiempo de flujo, para todo conjunto de operaciones, véase (2).

$$C_{m\acute{a}x} \geq t(\sigma_k^j) + p(\sigma_k^j) \forall \sigma_k^j \in \vartheta \quad (2)$$

RESTRICCION DE SECUENCIA

Otra restricción importante es la que define el tiempo de la operación $t(\sigma_k^j)$ como el inicio de la siguiente operación $t(\sigma_l^j)$ es decir, ninguna operación puede empezar sin antes haber terminado la operación anterior, véase (3).

$$t(\sigma_k^j) \geq t(\sigma_l^j) + p(\sigma_l^j) \forall \sigma_l^j < \sigma_k^j \quad (3)$$

RESTRICCIÓN DE USO

Esta restricción define el uso de las máquinas, restringiendo a solo un trabajo por maquina a la vez, al mismo tiempo que de los trabajos sólo pueden pasar una sola vez por cada máquina, véase (4)

$$t(\sigma_k^j) \geq t(\sigma_l^j) + p(\sigma_l^j) \vee \\ t(\sigma_l^j) \geq t(\sigma_k^j) + p(\sigma_k^j) \forall \sigma_l^j, \sigma_k^j \in \vartheta \text{ tal que } M_{\sigma_l^j} = M_{\sigma_k^j} \quad (4)$$

RESTRICCIÓN DE NO NEGATIVIDAD

Esta restricción define que no puede haber ningún tiempo de operación menor que 0, véase (5)

$$t(\sigma_k^j) \geq 0 \forall \sigma_k^j \in \vartheta \quad (5)$$

REFERENCIAS

Meisel, J. D., & Prado, L. K. (Julio de 2010). Un algoritmo genético híbrido y un enfriamiento simulado para solucionar el problema de programación de pedidos JobShop. *Revista EIA* .