Estudio comparativo de métodos heurísticos para programación de trabajos a gran escala sobre máquinas heterogéneas en paralelo

Juan Carlos Sotelo Villena, Walter Alberto Becerra Otoya, Luis Felipe Medina Aquino

jsotelo@uni.edu.pe, walter.becerra.o@uni.pe, lmedina@uni.edu.pe

Universidad Nacional de Ingeniería, Perú
Av. Túpac Amaru 210, Rímac
Lima – Perú

Resumen: *En este trabajo, se compara el desempeño de dos algoritmos heurísticos para la programación de trabajos a gran escala sobre máquinas no idénticas en paralelo (Parallel Machine Scheduling - PMS). Se presenta de forma genérica la evolución, tanto del problema de scheduling como de los métodos aplicados para su solución. Se revisan una serie de trabajos recientes sobre PMS principalmente de algoritmos heurísticos. Actualmente, la globalización de la economía y la tendencia de los clientes a individualizarse, han generado un escenario de lotes de producción más pequeños con múltiples variantes del producto, lo que se traduce en la necesidad cada vez mayor de casos de scheduling de gran escala, con cientos o miles de trabajos por programar. Por tanto, el problema de obtener soluciones aceptables en tiempos razonables, se torna crítico para que las empresas puedan ajustar y modificar, cada vez con mayor frecuencia, sus programas de producción. En este contexto, en este trabajo, se revisan e implementan a nivel experimental, el algoritmo Greedy Iterativo de Ying-Cheng y el algoritmo Genético de Savas. Se analiza como varía la calidad de la solución a medida que la carga de trabajos se incrementa desde 100 hasta 5000 trabajos. La variable aplicada para representar la calidad de la solución es el Atraso Máximo. El estudio realizado revela que el algoritmo Greedy Iterativo de Ying-Cheng se desempeña con ventaja cuando el número de trabajos tiende a ser mucho mayor. La investigación realizada evidencia que existe un amplio campo de estudio sobre temas de scheduling.*

Palabras clave: Programación paralela máquina, Programación, secuenciación, Algoritmo genético, Algoritmo voraz.

Abstract: *In this work it is studied the performance of two heuristic algorithms for large-scale programming on non-identical parallel machines (Parallel Machine Scheduling – PMS). Generically it presents the evolution of both, the scheduling problem as the methods used to solve them. A series of recent work on PMS mainly of heuristic algorithms are reviewed. Currently, the globalization of the economy and the trend of customers to individualized, have generated a scenario of smaller production batches with multiple product variants, resulting in the growing need for scheduling cases of large scale, hundreds or thousands of jobs per schedule. Therefore the problem of obtaining acceptable solutions at reasonable times, it becomes critical for companies to adjust and change, with increasing frequency, their production programs. In this context in this paper are reviewed and implemented on an experimental basis, the algorithm Greedy Iterative developed by Ying- Cheng and Genetic Algorithm developed by Savas. It analyzes the quality variation of the solution as the work load increases from 100 to 5000 jobs. The variable applied to represent the quality of the solution is the Maximum Delay. The study reveals that algorithm Greedy Iterative by Ying- Cheng is better when the number of jobs tends to be much higher. The research shows that there is a broad field of study on scheduling issues.*

Keywords: Parallel machine scheduling, scheduling, sequencing, genetic algorithm, greedy algorithm.

1 Introducción

La programación de trabajos sobre un conjunto de máquinas es un tipo de problema aceptado como un problema de combinatoria compleja del tipo NP-Hard [1] [2], que se define como el proceso de asignar un conjunto de trabajos a un conjunto de recursos (máquinas) dentro de un cierto periodo. Este tipo de problema es común en las industrias y representa un aspecto importante en la eficiencia y eficacia del proceso de producción, dado que una mejor utilización de los recursos incide en una reducción de los costes y en un mejor cumplimiento de las fechas de entrega al cliente.

Los trabajos de investigación sobre PMS se inician en la primera mitad del siglo XX cuando Gantt explora la secuenciación de trabajos y desarrolla el diagrama que lleva su nombre, luego Johnson crea su famosa regla para secuenciar trabajos secuenciar trabajos sobre una y dos máquinas [5].

A partir de la segunda mitad del siglo XX y con el auge de los ordenadores, se desarrollan múltiples métodos o algoritmos, primero basados en métodos exactos, como la programación lineal entera, pero que solo consiguen soluciones óptimas para problemas de tamaño reducido a pocas máquinas, pocos trabajos y pocas restricciones [7]. Luego, se desarrollan algoritmos basados en heurísticas para casos de mayor tamaño, donde el objetivo es encontrar una solución aceptable, no necesariamente óptima, en un tiempo computacional razonable. Actualmente, la investigación en PMS sigue vigente y los investigadores vienen explorando variantes del problema de PMS aplicando diversos algoritmos heurísticos, como búsqueda tabú [8] [11], recosido simulado [12] o algoritmos genéticos [5] [10], entre otros.

La dinámica de la economía globalizada ha generado un escenario de lotes de producción más pequeños y de múltiples variantes de producto, lo que se traduce en la necesidad cada vez mayor de tener casos de scheduling con miles de trabajos por secuenciar sobre decenas o centenas de máquinas, por tanto, el problema de obtener

soluciones en tiempo razonables se torna necesario para que las empresas puedan ajustar y modificar sus programas de producción cada vez con mayor frecuencia, debido a cambios en la prioridad de la demanda, eventualidades en la capacidad de la producción (averías) y eventualidades en el cumplimiento de los abastecimientos por parte de los proveedores [11].

Dada la alta complejidad que puede tomar el problema debido a la gran cantidad de datos y a las reglas muy particulares de cada sector de la industria, no es posible establecer necesariamente un mejor método o algoritmo y el estado del arte refiere a un conjunto importante de algoritmos heurísticos que compiten en la solución del problema bajo ciertas condiciones. Entre los algoritmos heurísticos más referidos tenemos a los algoritmos genéticos, la búsqueda tabú, el recocido simulado y los algoritmos greedy y grasp, entre otros. En esta investigación se estudian un Algoritmo Greedy Iterativo y un Algoritmo Genético frente a casos de alta cantidad de pedidos.

El resto de este paper está organizado de la siguiente manera. En la sección 2, se presenta el problema de programación de máquinas en paralelo. En la sección 3, se presenta una breve revisión de trabajos previos sobre PMS. La sección 4, describe el método de investigación aplicado. En las secciones 5 y 6, se describen de manera breve los algoritmos greedy iterativo (Ying-Cheng) y genético (Savas). En la sección 7, se describen los experimentos realizados y los resultados son presentados en la sección 8. Las conclusiones y recomendaciones se presentan en la sección 9.

2 El problema de programación de máquinas en paralelo (PMS)

Existe una amplia variedad de problemas de scheduling y pueden ser clasificados según algunos criterios [2]. Uno de los criterios más utilizados en la literatura del tema, suele clasificar los problemas de scheduling está en función de la organización de las máquinas y de la estructura de operaciones de los trabajos a realizar [7].

2.1. Definición del problema PMS

Parallel Machine Scheduling (PMS) es una clase de problema de scheduling definido y ampliamente estudiado en la literatura de investigación [7].

Baker [1] define el problema de PMS como un conjunto de trabajos de una sola operación que deben ser asignados y secuenciados en un conjunto de máquinas en paralelo. La figura 1 muestra un ejemplo de scheduling sobre 2 máquinas en paralelo.

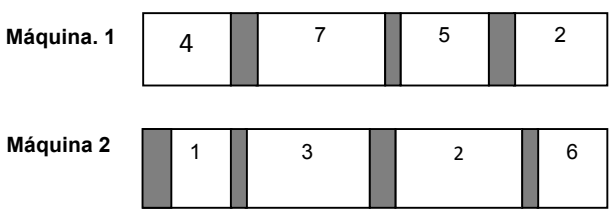

Figura 1: Ejemplo de asignación y secuenciación en dos máquinas.

En PMS número de soluciones posibles es del orden de (n!) m donde n es el número de trabajos y m es el número de máquinas, siendo un problema simultáneo de asignación y secuenciación.

Dentro de la clase de problema PMS es posible distinguir además dos subclases de problema:

PMS con máquinas idénticas, que constituye un caso particular pero muy ampliamente estudiado [8], debido a que es muy frecuente que se replique máquinas para ampliar la capacidad de producción. Para este caso, solo se presenta el problema de secuenciar en función de las restricciones del caso.

PMS con máquinas no idénticas, que constituye el caso general cuando los trabajos pueden ser resueltos solo por un subconjunto de las máquinas, pero con tiempo diferentes. Para este caso, se tiene en simultáneo el problema de asignación y secuenciación, es necesario determinar qué máquina debe realizar el trabajo y en qué secuencia respecto de los otros trabajos.

2.2. Medidas de efectividad

Para evaluar la calidad de las soluciones en PMS existen varios indicadores desarrollados por los investigadores, entre ellos Tiempo Máximo (Makespan), Atraso Total, Idleness o tiempo total de ocio, Tiempo total para completar todos los trabajos y , Atraso Maximo [6]. En esta investigación se toman el atraso máximo y el tiempo total para completar todos los trabajos como los indicadores de la calidad de la solución.

3 Revisión de trabajos en PMS

Para tomar conocimiento de los algoritmos heurísticos más referidos en la literatura de investigación sobre PMS, se revisaron 8 artículos de journals indexados publicados desde el año 2009 hasta el año 2012. En cada caso se indica el problema tratado, el aporte del autor y una apreciación crítica sobre el trabajo.

Gur y Assaf en el 2009 [6] estudian el problema de PMS sobre máquinas no idénticas y plantean la minimización del tiempo total para programar los trabajos incorporando una restricción para el mantenimiento de las máquinas. Estudian dos posibilidades del problema: una que asume que todas las máquinas deben ser mantenidas simultáneamente y antes de una fecha dada y otra suponiendo qué mantenimiento se puede realizar en puntos del tiempo diferentes por cada máquina pero antes de una fecha dada.

El algoritmo de Gur y Assaf introduce una heurística eficiente y una cota inferior simple de calcular, basada en, primero, acomodar los pedidos lo más eficientemente posible sin considerar el mantenimiento y luego reacomodan la programación para considerar el mantenimiento tratando de desmejorar la solución inicial lo menos posible manteniendo la secuencia inicial y solo desplazando los trabajos para considerar los tiempos de mantenimiento.

Los casos que estudian son solo de medio tamaño (decenas o muy pocas centenas de trabajos y solo hasta tres máquinas en paralelo). Solo consideran una actividad

de mantenimiento en el horizonte de programación. No consideran que los trabajos pueden estar restringidos por fecha de inicio. Es rescatable el manejo de la restricción en referencia a la necesidad de realizar actividades de mantenimiento. También es interesante su propuesta de crear los datos aleatoriamente utilizando la técnica de Montecarlo.

Kai Li, Ye Shi, Shan-lin Yang y Ba-yi Cheng en el 2009 estudian el problema de PMS sobre máquinas idénticas [8] considerando tiempos de proceso dependientes del consumo de recursos y propones un algoritmo basado en Recosido Simulado y desarrollan una serie de experimentos con data simulada para probar la bondad del algoritmo. La heurística del recosido simulado es un proceso de búsqueda inspirado en el proceso que ocurre al condesarse la materia en los procesos metalúrgicos [8].

En uno de los pocos trabajos encontrados donde el problema PMS es explorado con pruebas de tamaño considerable (hasta 1000 trabajos), pero con un número reducido de máquinas (8 máquinas), además no consideran el caso de máquinas no idénticas. En este trabajo, también se destaca la preparación de data aplicando simulación Montecarlo.

Kuo-Ching Ying y Hui-Miao Cheng en el 2010 estudian el problema de PMS con máquinas idénticas [9], pero donde los tiempos de tiempos de set up son dependientes de la secuencia de asignación y los pedidos tienen fechas de llegada diferentes. Para la solución, desarrollan un algoritmo Greedy Iterativo para minimizar el máximo atraso.

El algoritmo crear una solución inicial completa con algún criterio y luego inicia un proceso repetitivo de retirar aleatoriamente algunas asignaciones y reasignarlas de modo de lograr que el atraso máximo sea menor que el que existía en la solución anterior. El proceso es repetido hasta lograr un atraso máximo aceptable o hasta consumir un tiempo asignado al proceso.

Inci y Cenk en el 2011 [7] plantean que los problemas de PMS se pueden considerar como un proceso de 2 fases, una para determinar en qué máquina se debe realizar un trabajo y otro para determinar la secuencia dentro de cada máquina. Estudian el problema de minimizar el tiempo total de atraso para el caso de máquinas idénticas, pero considerando que los trabajos pueden fraccionarse y que existen tiempos de set up.

Plantean dos algoritmos, uno basado en Búsqueda Tabú y otro basado en Recocido Simulado para reducir el impacto del tiempo de set up en el tiempo total.

En sus pruebas solo trabajan con casos de tamaño medio (decenas de trabajos). Solo consideran el caso de máquinas idénticas y no tiene restricción alguna sobre la partición de trabajos. Encuentran que el Recosido Simulado les da mejores resultados que la Búsqueda Tabú.

Shih-Wei, Zne-Jung, Kuo-Ching y Chung-Cheng en el 2011 investigan el problema de PMS para minimizar el máximo atraso considerando máquinas idénticas, pero con tiempos de set up dependientes de la secuencia. Desarrollan un algoritmo Greedy que incorpora un criterio de aceptación basado en recocido simulado.

Este trabajo solo considera maquinas idénticas y estudia casos que son de tamaño medio (centenas de trabajos).

Savas Balin en el 2011 estudia el problema de PMS considerando el caso de minimizar el makespan para máquinas no idénticas y desarrolla un Algoritmo Genético, señalando que es fácil de adaptar e implementar [11]. Los algoritmos genéticos están inspirados en el proceso de evolución Darwiniano. Holland (1975) desarrolló la idea inicial y se trata de iterativamente mejorar una solución inicial por intercambio y mutación. Para problemas de scheduling, Davies (1987) fue el primero en introducir esta heurística [3].

Savas prueba su Algoritmo Genético en una serie de experimentos computacionales con resultados favorables. Es uno de los pocos autores que trabajan casos de medio a gran tamaño (hasta 500 órdenes, pero con muy pocas máquinas), pero no considera tiempos de set up, ni fechas restringidas de inicio. Concluye que los Algoritmos Genéticos son muy versátiles y pueden ser adaptados a diferentes problemas de scheduling.

Mir Abbas Bozorgirad y Rasaratnam Logendran [10] estudian el problema PMS de optimizar simultáneamente el tiempo para completar todos los trabajos y el atraso total en la programación de grupos de trabajos sobre máquinas no idénticas, considerando tiempos de set up. Formulan de un modelo matemático y afirman que el problema es NP-Hard y desarrollan un algoritmo basado en búsqueda Tabú.

Señalan que cuando se aplica Búsqueda Tabú la calidad final depende de la calidad de la solución inicial y proponen 2 mecanismos para identificar una solución inicial adecuada. Para ello, aplican dos reglas, considerar primero a los trabajos con menor tiempo de proceso y la considerar primero a los trabajos con menor fecha de entrega. Generan dos secuencias y luego las combinan. Esta lógica es aplicada en dos niveles, primero secuencia de grupos y luego secuencia de trabajos dentro de cada grupo.

Es rescatable la consideración sobre la importancia del afinamiento de los parámetros de una búsqueda Tabú y la conclusión de que dependen de la estructura del problema. También es interesante considerar la generación aleatoria de datos para crear los experimentos de prueba.

Estudian solo casos de tamaño medio (decenas de trabajos) sobre muy pocas máquinas.

Ching-Jong, Chien-Wen y Liang-Chuan en el año 2012 [9]. Estudian el problema de PMS desde el enfoque de minimizar el tiempo total para completar todos los trabajos, considerando el caso de máquinas idénticas, con tiempos de set up por familias de trabajos. Desarrollan una heurística mixta combinando la Búsqueda Tabú con dos reglas de dominancia tomadas de Dunstall y Wirth [4], para reducir el impacto del tiempo de set up en el tiempo total.

Los autores solo consideran el caso de máquinas idénticas y solo estudian casos de tamaño mediano (decenas de trabajos), no incorporan fecha de entrega diferentes.

Como consecuencia de la revisión de la literatura, se encontró que los algoritmos heurísticos referidos son entre otros: Greedy Iterativo, Búsqueda Tabú, Recosido Simulado, Algoritmos Genéticos, Cota mínima y reglas de dominancia.

4 Método de investigación

El método de investigación aplicado en el presente trabajo considera 5 pasos:

Paso 1: Revisión de papers sobre métodos y algoritmos para resolver problemas sobre PMS y selección de 2 algoritmos para el estudio comparativo.

Paso 2: Revisión e implementación a nivel experimental de los algoritmos seleccionados, considerando la homogenización del problema para una comparación bajo condiciones equivalentes.

Paso 3: Prueba experimental escalonada con juegos de datos de 5000 trabajos.

Paso 3.1: Generación de data aleatoria.

Paso 3.2: Aplicación incremental de la data aleatoria sobre ambos algoritmos y determinación del máximo atraso y del tiempo total asignado en cada caso.

Paso 4: Análisis de los resultados obtenidos en las pruebas experimentales.

Paso 5: Generación de conclusiones y recomendaciones.

5 Revisión e implementación del algoritmo Greedy Iterativo (Ying-Cheng)

Yin y Cheng [9] definen el problema de PMS bajo los siguientes supuestos:

- Las máquinas son idénticas.
- Cada trabajo tiene una fecha de entrega (due date) y una fecha de mínimo de inicio (release date).
- Los trabajos están ordenados por fecha de mínimo inicio.
- Tiempos de set up diferentes para pasar de un trabajo a otro.

El algortimo greedy iterativo propuesto por Ying y Cheng, define las siguientes variables:

N = {1,2, …, n} Conjunto de n trabajos

M = {1,2, …, m} Conjunto de m máquinas idénticas

d_i fecha de entrega del trabajo i

r_i fecha de mínimo inicio del trabajo i (ri ≤ ri + 1)

s_{ij} Tiempo de set up para pasar del trabajo i al trabajo j

s_{oi} Tiempo de set up para el primer trabajo que se coloca en una máquina.

p_i Tiempo de proceso del trabajo i

C_i Fecha en que se termina el trabajo i

L_i Atraso del trabajo i (Li = Ci - di)

Define como Función objetivo: Minimizar máximo atraso (L_i).

6 Revisión e implementación del Algoritmo Genético (Savas)

Savas define el problema de PMS bajo los siguientes supuestos:

- Cada trabajo se realiza en una sola máquina.
- Las máquinas tienen diferente velocidad para realizar diferentes trabajos.
- No existen tiempos de set up para las máquinas.
- Todos los trabajos son igualmente urgentes.
- Todos los trabajos son de similar dificultad, el tiempo de procesamiento de estos depende sólo de la máquina en que se trabaja.

El algoritmo genético desarrollado por Savas considera las siguientes variables:

$V_{(i)}$: velocidad de la máquina i

$P_{(i,j)}$: Tiempo de procesamiento del trabajo j cuando se realiza en la máquina i

$X_{(i,j)}$: matriz booleana (Si o No) que determina si el trabajo j se ha de realizar en la máquina i

Un trabajo se realiza solo en una sola máquina, de manera que se puede corroborar lo siguiente en la matriz booleana:

$$\sum_{i=1}^{m} X_{(i,j)} = 1, \ j = 1,2,3 \dots n$$

Por condición del problema, se considera que los trabajos son de similar dificultad, por lo que, para un trabajo j se cumple:

$$P_{(i1,j)}V1 = P_{(i2,j)}V2$$

Además, para calcular los tiempos de los trabajos programados en las máquinas, necesitamos multiplicar la matriz P con la matriz Booleana X. Se cumple que: duración total de la programación:

$$C_{max} = \max_{i=1}^{m} \{\sum_{k=0}^{n} X_{(i,j)}P_{(i,j)}\}$$

Función objetivo:

$$min_{k=1}^{N}\{C_{max}(k)\}, \ k = 1, \ \dots, \ N$$

donde N es el tamaño de la población.

7 Experimentación

El objetivo fue analizar cómo varía la calidad de la solución en cada uno de los dos algoritmos a medida que la carga de trabajos se incrementa desde 100 hasta 5000

trabajos. Como se indicó al definir el problema de PMS, para evaluar la calidad de la solución se tomaron los indicadores de atraso máximo y tiempo total para completar todos los trabajos.

7.1. Homogenización de la definición del problema para la comparación de los algoritmos

Para el caso del algoritmo Greedy Iterativo, se realizaron las siguientes relajaciones:

- No se consideran tiempos de set up. Además, se pasará de un vector de tiempos de trabajo que depende únicamente del pedido, hacia una matriz de tiempos de trabajo que depende de la máquina y el pedido.
- Todos los trabajos tienen igual release date (fecha de liberación) e igual due date (fecha de término).
- Tras una serie de pruebas previas, se vio por conveniente fijar un alfa =5, que equivale a que en cada iteración para mejorar la solución base se retiren y reacomoden 5 trabajos.

Para el caso del Algortimo Genético de Savas, se hicieron los siguientes ajustes:

- La población de soluciones en cada iteración es igual al 10% del número de trabajos que se quiera programar.
- La fracción de cromosomas que se elegirán en el proceso de selección natural es igual al 10% del total.
- Los valores de los parámetros para el cálculo de los valores Fitness se asignan: alfa = 0.01 y beta = 0.01, de acuerdo a recomendación del autor.

7.2. Datos para las pruebas

Siguiendo la tendencia encontrada en los trabajos revisados de utilizar data aleatoria para los experimentos, se generaron 3 juegos de datos aleatorios de 5000 pedidos, considerando lo siguiente:

- Cantidad de máquinas consideradas: 50.
- Cantidad de trabajos considerados: Incremental de 100 a 5000.
- Velocidad de máquinas (aleatorio): de 1 a 3 unidades por minuto.
- Tamaño de pedidos (aleatorio): de 10 a 1000 unidades.
- El tiempo para cada corrida fue de 30 minutos.

8 Resultados

Los resultados de las pruebas experimentales realizadas con los 3 conjuntos de datos, incrementado progresivamente el número de trabajos a programar, se presentan en la tabla 1.

Los resultados muestran que en el caso del algoritmo de Ying-Cheng el atraso máximo crece linealmente a diferencia del algoritmo de Savas donde el crecimiento no es lineal y a partir de los 3000 trabajos se incrementa más rápidamente que el de Ying-Cheng.

Tabla 1: Evolución del Atraso Máximo y Tiempo Total

Corrida	Nro. de Trabajos	Savas		Ying Chen	
		Atraso Max (min)	Tiempo Total (min)	Atraso Max (min)	Tiempo Total
1	100	11.11	441.07	10.10	446.39
2	500	46.00	2269.4	47.61	2266.71
3	1000	91.22	4527.08	94.58	4530.65
4	1500	134.61	6707.09	137.98	6703.17
5	2000	179.74	8965.72	185.89	8948.1
6	2500	222.09	11082.93	225.07	11083.11
7	3000	264.47	13201.59	268.96	13197.49
8	3500	325.92	15495.83	314.19	15384.28
9	4000	433.08	18233.31	358.44	17643.88
10	4500	532.69	20841.62	403.82	19810.84
11	5000	640.13	23424.15	444.69	21997.28
12	100	10.65	435.86	9.68	446.26
13	500	44.94	2206.82	46.82	2211.36
14	1000	89.86	4466.76	93.66	4471.8
15	1500	134.89	6721.05	138.93	6741.13
16	2000	178.90	8924.91	183.53	8924.86
17	2500	223.46	11149.59	228.61	11157.27
18	3000	268.21	13390.95	275.93	13411.38
19	3500	350.44	15975.11	318.03	15584.68
20	4000	455.18	18644.71	360.44	17752.82
21	4500	555.37	21320.52	404.01	19987.14
22	5000	663.29	24062.07	454.42	22262.67
23	100	11.17	470.5	10.31	470.03
24	500	48.26	2371.53	50.96	2371.45
25	1000	95.58	4751.39	102.16	4745.42
26	1500	141.62	7061.53	145.75	7047.62
27	2000	190.06	9478.75	193.29	9469.63
28	2500	236.93	11822.05	241.67	11830.91
29	3000	285.66	14262.33	289.64	14254.94
30	3500	366.95	16920.11	340.06	16634.49
31	4000	487.72	20024.82	391.84	19104.35
32	4500	572.72	22648.03	439.74	21410.03
33	5000	686.91	25579.17	484.32	23775.57

La figura 2 muestra, de forma gráfica, cómo varía el Atraso Máximo en cada algoritmo a medida que se incrementa el número de trabajos.

La figura 3 muestra que en cuanto al indicador de tiempo total para completar todos los trabajos si bien hay una diferencia favorable al algoritmo Greedy Iterativo, ésta es muy pequeña.

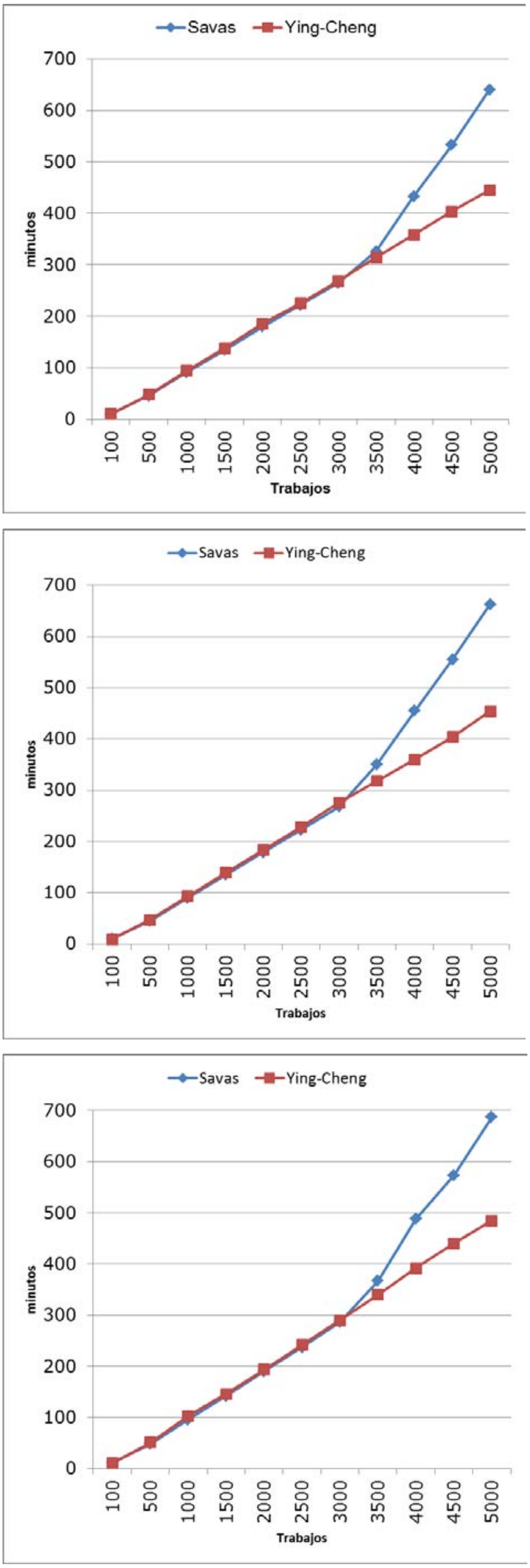

Figura 2: Evolución del Atraso Máximo al variar el número de trabajos

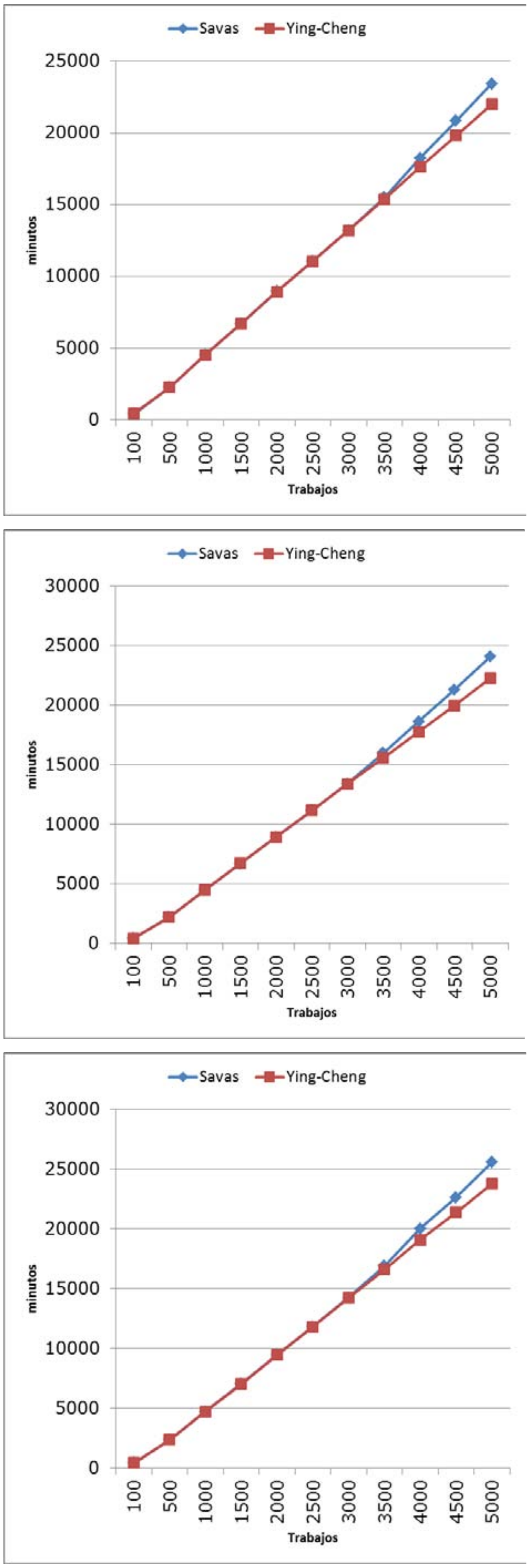

Figura 3: Evolución del Tiempo Total para completar todos los trabajos al variar el número de trabajos

9 Conclusiones y trabajos futuros

El trabajo realizado permite enunciar las siguientes conclusiones:

Los problemas sobre PMS son ampliamente estudiados en la literatura de investigación y cada investigador centra su trabajo en algún aspecto particular del problema, lo cual crea mayores posibilidades de investigación en el tema.

Los trabajos revisados evidencian que aún son escasos los estudios de PMS a gran escala.

Se estudiaron e implementaron a nivel experimental, dos algoritmos para resolver problemas sobre PMS: Algoritmo Greedy Iterativo (Ying-Cheng) y Algoritmo Genético (Savas).

Se analizó y se comparó como varia la calidad de la solución, expresada por el indicador de atraso máximo, de los dos algoritmos cuando se incrementa la carga de trabajo desde 100 hasta 5000 pedidos.

Para que la comparación sea bajo las mismas condiciones se relajó la definición del problema PMS de Ying-Cheng para hacerla equivalente al PMS definido por Savas.

Se encontró, en todas las pruebas, que frente a una alta carga de trabajos, el algoritmo Greedy Iterativo de Ying-Cheng es superior al algoritmo Genético de Savas.

En todas las pruebas hasta aproximadamente 3000 trabajos, no hay una diferencia notoria entre ambos algoritmos respecto del atraso máximo, pero a partir de este punto la diferencia comienza a crecer favoreciendo al algoritmo Greedy Iterativo.

Se recomienda explorar otros algoritmos heurísticos que definan el problema de PMS de manera similar a Ying-Cheng y realizar nuevos estudios de comparación.

Es recomendable también explorar la posibilidad de incluir una mejora en el algoritmo de Ying-Cheng para reemplazar la selección aleatoria de trabajos para el reacomodo por una selección de los trabajos con mayor atraso.

Referencias bibliográficas

1. Baker, Keneth, ”Principles of Sequencing and Scheduling”. John Wiley & Sons, Inc. USA, 2009.
2. Ching-Jong Liao, Chien-Wen Chao, Liang-Chuan Chen. “An improved heuristics or parallel machine wighted flowtime scheduling with family set-ups times". Journal Computer and Mathematics with Aplications, 2012
3. Davies, L. & Coombs, S. “Genetic algorithms and communications link speed desing: Theoretical considerations”. Grefenstette, 1987.
4. Dunstall, S. Wirth, A. Baker, K.R. “Lower bounds and algorithms for flowtime minimization on a single machine with set-up times”. Journal of Scheduling 3, 2000.
5. Gacias, B., Artigues, C., López, P., “Parallel machine scheduling with precedence constraints and setup times”. Computers and Operations Research 37 (12), pp. 2A11-2151, 2010
6. Gur Mosheiov, Assaf Sarig, "A Note: Simple heuristics for scheduling a maintenance activity on unrelated machines". (2009) Journal Computer and Operations Research, 2009.
7. Inci, Saricicek y Cenk Celik, "Two Meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness". Journal Applied Mathematical Modeling, 2011.
8. Kai Li, Ye Shi, Shan-lin Yang, Ba-yi Cheng. “Parallel machine scheduling problem to minimize the makespan with dependent processing times”. Journal Computers & Operations Research, 2009.
9. Kuo-ching Ying, Hui-Miao Cheng, “Dynamic parallel machine scheduling with sequence-dependent times using an iterated greedy heuristic". Journal Expert Systems with Applications, 2010.
10. Mir Abbas Bozzorgirad, Rassaratnam Logendran, “Sequence-dependent group scheduling problem on unrelated-parallel machines”. Journal Expert Systems with Applications. 2012.
11. Savas Balin, “Non-identical parallel machine scheduling using genetic algorithm”. Journal Expert Systems with Applications, 2011.
12. Shih-Wei Lin, Zne-Jung Lee, Kuo-Ching Ying, Chung-Cheng Lu, “Minimization del maximum lateness on parallel machines with sequence-dependent setup times and job release dates”. Journal Computer and Operations Research, 2011.