

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 24/09/2012 | Aceptado: 17/12/2012

Comparación de un algoritmo genético y la técnica nube de partículas en la solución del flow shop scheduling

Comparison of a genetic algorithm and the particles swarm technique in the solution of the flow shop scheduling

Yunior César Fonseca Reyna^{1*}, José Eduardo Márquez Delgado¹

¹ Departamento de Informática, Universidad de Granma, Km 18 ½ Carretera Manzanillo, Bayamo, Granma, Cuba.
jmarquezd@udg.co.cu

*Autor para la correspondencia: fonseca@uci.cu

Resumen

El problema de secuenciamiento de tareas o Flow Shop Scheduling Problems (*FSSP*) es un problema clásico de la programación de trabajos. Este problema es típico de la optimización combinatoria y se presenta en talleres con tecnología de maquinado donde existen máquinas-herramientas convencionales y se fabrican diferentes tipos de piezas que tienen en común una misma ruta. La solución de este problema consiste en encontrar una secuencia de tareas que emplee un tiempo mínimo de procesamiento (*makespan*). El mismo está incluido dentro de la gran variedad de problemas de planificación de recursos, el cual como muchos otros en este campo, es de difícil solución y está clasificado técnicamente como de solución en un tiempo no polinomial. En la presente investigación se implementa un algoritmo genético simple y la técnica nube de partículas para dar solución a dicho problema y se compara la calidad de dichas soluciones lográndose resultados satisfactorios y de gran interés. Para validar los resultados arrojados por ambas técnicas computacionales se usan como casos de prueba, instancias de problemas de planificación de la literatura especializada de diferentes tamaños y complejidad.

Palabras clave: Algoritmo genético, Flow Shop, makespan, nube de partículas, optimización, secuenciamiento de tareas.

Abstract

Flow Shop Scheduling Problems (FSSP) is a classic problem of the tasks programming. This problem is typical of the combinatorial optimization and it is presented in factories with technology where conventional machine-tools exist and different types of pieces are manufactured having in common the same route. The solution of this problem consists on finding a sequence of tasks that uses a minimum processing time (makespan). The same one is included inside the great variety of planning resources problems and it is classified technically as NP-hard problem. In the present research is implemented a simple genetic algorithm and the technic of particle swarm optimization to give solution to this problem and the quality of both solutions are compared obtaining interesting results. To validate the results released by both computational technics, instances of planning problems of different sizes and complexity from specialized literature are used as cases of test.

Keywords: Genetic Algorithm, Flow Shop, makespan, optimization, particle swarm, scheduling.

Introducción

La planificación de operaciones es un problema de optimización que se presenta con frecuencia en sistemas de producción convencionales automatizados. Este es un problema común donde está involucrada la toma de decisiones con respecto a la mejor asignación de recursos a procesos de información en los cuales se tienen restricciones de temporalidad. Dichos problemas deben ser resueltos en una amplia gama de aplicaciones tales como: programación de despacho de vuelos en los aeropuertos, programación de líneas de producción de una fábrica, programación de cirugías en un hospital, reparación de equipos o maquinarias en un taller, en las telecomunicaciones, construcción de piezas en procesos tecnológicos de maquinado, entre otros (Šeda, 2007) , (Álvarez, *et al.*, 2008).

La cadena de producción estructurada para la planificación de procesos requiere que cada una de las tareas a ejecutar pase por todas las máquinas en el mismo orden. Este tipo de problema, en el cual el orden de procesamiento es continuo, es decir, una vez que el procesamiento de un trabajo comienza no debe haber ningún tiempo de espera entre el procesamiento de las tareas consecutivas de este, es denominado **Secuenciamiento de Tareas** o **Flow Shop Scheduling**. El mismo está incluido dentro de la gran variedad de problemas de planificación de recursos, el cual, como muchos otros en este campo, es de difícil solución y está clasificado técnicamente como de solución en un tiempo no polinomial (Pinedo, 2008). El problema fundamental consiste en encontrar una programación adecuada que

pueda reducir significativamente los costos de producción y reducir los tiempos de proceso permitiendo cumplir con los compromisos de entrega en un período razonable.

En los últimos años se han propuesto un gran número de enfoques para modelar y solucionar el problema de la planificación de tareas con diferentes grados de éxito. Entre estos enfoques se pueden mencionar la *Programación Matemática* (Johnson, 1954), y dentro de este ámbito la *Programación Lineal Entera Mixta* (Šeda, 2007), (Manne, 1960). Otro método utilizado para resolver el problema de la programación de la producción han sido las técnicas de *Ramificación y Acotación* (Companys, 2005), (Lagewegm, *et al.*, 1977). También se han usado las heurísticas basadas en *Cuello de Botellas* (Yamada, 2003), (Adams, 1988) (Lee, *et al.*, 2004).

Con el desarrollo de la Inteligencia Artificial (*IA*) han emergidos otras metodologías como es el caso de las *Redes Neuronales* (Zhang, 1995) y últimamente los métodos de Búsqueda Locales y Metaheurísticas, entre los cuales se encuentra el *Recocido Simulado* (Álvarez, *et al.*, 2008), (Kirkpatrick, *et al.*, 1983), (Cerny, 1985); la *Búsqueda Tabú* (Glover, 1990), los *Algoritmos Genéticos (AGs)* (Matía, 2010), (Toro, *et al.*, 2006b), (Delgado, 2005), (Yamada, 2003), la *Optimización basada en Nubes de Partículas (PSO)* (Toro, *et al.*, 2006), siendo estas últimas las metodologías utilizadas en el desarrollo de esta investigación, entre otras.

Los AGs y PSO son particularmente aplicados en problemas complejos de optimización: problemas con diversos parámetros o características que precisan ser combinados en función de encontrar mejores soluciones; problemas con muchas restricciones o condiciones que no pueden ser representadas matemáticamente, y problemas con grandes espacios de búsqueda (Cavalcanti, 2009).

En esta investigación se pretende realizar un estudio de los resultados obtenidos en la aplicación de dichas técnicas al problema de secuenciamiento de tareas determinando cuál de ellas arroja mejores resultados usando como casos de prueba instancias de problemas de planificación de diferentes tamaños y complejidad.

Definición del problema

De forma general, el problema consiste en programar de forma óptima, un conjunto de N tareas que deben de ser procesadas en un conjunto de M máquinas, considerando que todas las tareas tienen la misma secuencia de producción. El objetivo es minimizar el tiempo total requerido para terminar todas las tareas (*makespan*).

Los parámetros más relevantes del FSSP se detallan a continuación:

El índice $i \{i = 1 \dots N\}$ corresponde al número de tarea, $j \{J_1, \dots, J_M\}$ corresponde al número de la máquina y P_{ij} es el tiempo de procesamiento requerido por la tarea i en la máquina j . Por tal razón, un trabajo consiste de M operaciones y la j -ésima operación de cada trabajo debe ser procesada en la máquina j .

El objetivo es encontrar la secuencia de los trabajos a programar la cual debe minimizar el *makespan*. En este contexto se considera el trabajo i como un conjunto de operaciones, pasando por cada máquina j una sola vez:

- $i = \{o_{1i}, o_{2i}, o_{3i}, \dots, o_{ji}\}$ donde o_{ji} representa la j -ésima operación del trabajo i y debe ser procesada en la máquina j ;
- Por cada operación o_{ji} hay asociado un tiempo de procesamiento P_{ji} .

Sea $\pi_1, \pi_2, \pi_3, \pi_4, \dots, \pi_N$ la secuencia de trabajos a realizar. El procedimiento para el cálculo de $C(\pi_i, j)$, que representa el tiempo total de ejecución para el tiempo que tarda el trabajo i en salir de cada una de las máquinas se representa a continuación.

Para la primera máquina se realizaría mediante el conjunto de ecuaciones (1).

$$\begin{aligned}
 C(\pi_1, 1) &= p_{\pi_1, 1} \\
 &\cdot \\
 &\cdot \\
 C(\pi_i, 1) &= C(\pi_{i-1}, 1) + p_{\pi_i, 1}, \text{ con } i = 2, \dots, N
 \end{aligned} \tag{1}$$

Posteriormente se realiza idéntico procedimiento pero ahora con la máquina dos (conjunto de ecuaciones -2-).

$$\begin{aligned}
 C(\pi_1, 2) &= C(\pi_1, 2) + p_{\pi_1, 2} \\
 &\cdot \\
 &\cdot \\
 C(\pi_i, 2) &= C(\pi_{i-1}, 2) + p_{\pi_i, 2}, \text{ con } i = 2, \dots, N
 \end{aligned} \tag{2}$$

Generalizando, el $C(\pi_i, j)$ puede ser calculado como:

$$C(\pi_i, j) = C(\pi_i, j-1) + p_{\pi_i, j} \tag{3}$$

El tiempo acumulado de la secuencia de producción hasta que finaliza la ejecución de la tarea π_i en la máquina j se obtiene mediante la ecuación (4).

$$C(\pi_i, j) = \max\{ C(\pi_{i-1}, j), C(\pi_i, j-1) \} + p_{\pi_i, j} \tag{4}$$

Bajo estas condiciones, el *makespan* o (C_{max}) como también se le conoce, está dado por $C(\pi_N, M)$, que corresponde al tiempo del último trabajo en la última máquina (Yamada, 2003), (Ríos-Mercado, 2001), (Ríos-Mercado, 1999). En otras palabras, es el tiempo para completar todos los trabajos. Esta versión del problema es conocida en la literatura como $n/m/P/C_{max}$.

A continuación se ilustra un ejemplo de cómo se procede para evaluar el *makespan*. En este caso se solicita el ordenamiento de 4 trabajos que deben ejecutarse en 3 máquinas cumpliendo con los tiempos de operación presentados en la Tabla 1.

Tabla 1. Tiempos de Procesamiento

P_{ji}	$i = 1$	2	3	4
$j = 1$	4	4	3	2
2	7	3	2	1
3	5	5	3	1

Una secuencia de trabajo es representada como $X = [\pi_1, \pi_2, \pi_3, \pi_4, \dots, \pi_N]$. La Figura 1 muestra la obtención del *makespan* para la secuencia de trabajos $X = [2, 1, 3, 4]$ donde $C_{max} = 24$.

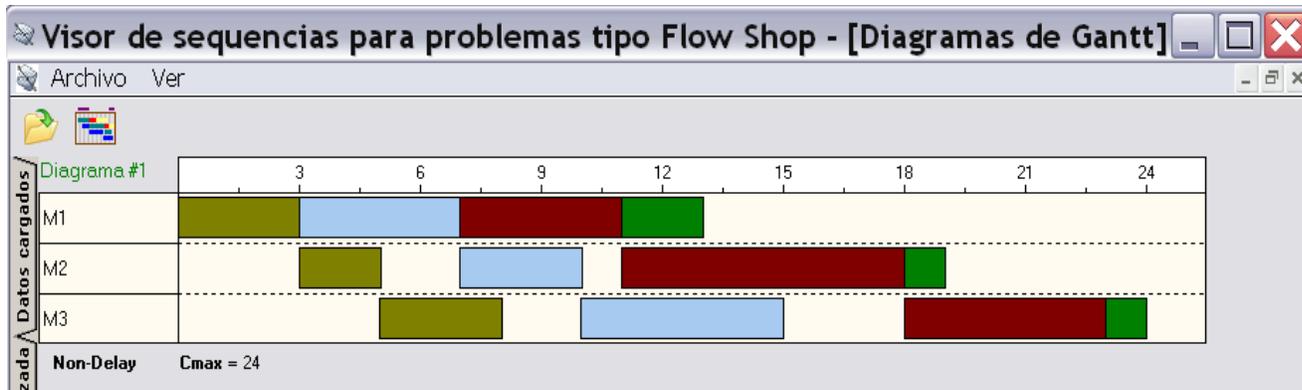


Figura 1. Obtención del Makespan a través del Diagrama de Gantt.

Metodologías de solución

En esta investigación el problema de FSSP se caracteriza por que todos los productos siguen la misma secuencia de producción; la idea es determinar la mejor secuencia de trabajos y para ello se podría numerar todas las secuencias posibles y elegir aquella que optimiza alguna medida de desempeño.

Por ejemplo, para una programación de 23 trabajos implicaría analizar $23! = 2.59 \cdot 10^{22}$ secuencias posibles. Si un equipo pudiera examinar un billón por segundo, tardaría 8.2 siglos para examinarlas todas, eso sin considerar restricciones asociadas a los recursos adicionales (mano de obra y materia prima) y las dependencias entre trabajos (como la preparación), que le dan un mayor grado de dificultad. De acuerdo a la explosión combinatorial del espacio de soluciones, las metaheurísticas han mostrado su gran potencial como herramientas de solución para este tipo de problemas.

Se proponen las técnicas AGs y PSO para la solución del FSSP considerando como función objetivo la minimización del tiempo total de ejecución de todas las tareas. Ambas técnicas computacionales están basadas en una población de soluciones las cuales se desplazan en este espacio de búsqueda logrando mejores soluciones.

Los AGs han ganado una gran importancia en el sector académico en los últimos años debido a la calidad en las respuestas y en el alto desempeño computacional obtenido al adaptarse a nuevos problemas. La principal característica del Algoritmo Genético implementado consiste en mantener constante el tamaño de la población de alternativas de solución, de manera que en cada iteración se reemplaza una alternativa de la población usando un eficiente mecanismo de modificación de la misma (cruzamiento, mutación). En cada iteración la población es reemplazada sistemáticamente por un único descendiente generado. Esta estrategia tiene la ventaja de permitir encontrar múltiples soluciones y además conservar la diversidad del conjunto de alternativas.

Por otra parte, la técnica de optimización combinatorial conocida como Nube de Partículas o PSO, está basada en una población de soluciones que se guía a través de la inteligencia colectiva de dicha población. Esta técnica de optimización está inspirada en el comportamiento social de los animales tales como una bandada de pájaros, un cardumen de peces o un enjambre de abejas. El algoritmo PSO implementado parte de una población para iniciar un proceso de búsqueda, donde cada individuo se refiere a una partícula que está agrupada dentro de una nube (un pájaro, dentro de una bandada). Cada partícula dentro de la nube representa un candidato a la solución del problema de optimización. En PSO cada partícula se desplaza a través de un espacio de búsqueda multidimensional, ajustando su posición de acuerdo a su propia experiencia y la de otras partículas en su vecindario, es decir, una partícula hace uso de la mejor posición encontrada por sí misma (*pbest*) y de la mejor posición de su vecindario (*gbest*), para dirigirse hacia la solución óptima.

Resultados y discusión

Dado que los AGs y PSO son mecanismos de carácter estocástico y no exacto, su validez como métodos de búsqueda de soluciones debe ser realizada de forma experimental. En general, se deben evaluar no solamente la eficiencia y eficacia como en cualquier otro método de búsqueda, sino también la estabilidad por tratarse de métodos de naturaleza estocástica. En el caso de los problemas de *scheduling* existen bancos de ejemplos de uso común entre los investigadores, lo cual facilita la comparación de distintos métodos de solución así como evaluar la influencia de cada uno de ellos variando internamente algún parámetro de control.

Para la evaluación de las técnicas computacionales implementadas aplicadas al problema de planificación, se han seleccionado problemas estándar de amplio estudio de la comunidad científica sobre el FSSP disponibles en la Internet en el repositorio *OR-Library*¹. Este repositorio contiene instancias de problemas de diferentes tamaños y complejidad.

Inicialmente, para encontrar los parámetros óptimos para cada una de las técnicas se escogieron tres problemas de diferentes tamaños, específicamente: tai20_05_02, tai20_20_02, tai100_10_01 y se crearon parámetros iniciales para cada problema. Luego se fueron variando los valores de los mismos hasta lograr una calibración que presentara las mejores soluciones. Estos problemas fueron evaluados 10 veces cada uno con varias combinaciones de parámetros y se tomó la combinación más estable en cuanto a la calidad de los resultados.

Para poder establecer una comparación entre ambas técnicas se tuvieron en cuenta dos de sus parámetros principales: en el número de iteraciones, el cual constituyó el criterio de parada de los algoritmos. El número de iteraciones para ambos fue de 200. También nos basamos en el número de individuos para el AG y PSO. Para esta investigación el número de individuos fue de 40. A continuación se muestran los parámetros que mejores resultados reportaron en las pruebas efectuadas a las instancias de problemas anteriormente mencionadas.

Tabla 2. Parámetros óptimos del Algoritmo Genético

Parámetro	Valor
Número de Generaciones	200
Cantidad de cromosomas	40
Factor de Cruzamiento	0.53
Factor de Mutación	0.021
Método de Selección	Ranking
Método de Cruzamiento	Dos Puntos de Cruce

¹ OR-Library: <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>

Método de Mutación	Mutación por Valor
Tipo de Modelo	Modelo Estacionario

Tabla 3. Parámetros óptimos para PSO

Parámetro	Valor
Número de iteraciones	200
Cantidad de partículas	40
Razón de aprendizaje cognitivo ($c1$)	1.5
Razón de aprendizaje social ($c2$)	1.5
Peso de la inercia (w)	$0.5 * \text{rand}() / 2$

Una vez calibrados cada uno de los parámetros de los algoritmos propuestos, se escogieron aleatoriamente 14 instancias de problemas estándar pertenecientes a la *OR-Library* (específicamente 2 por tamaño y complejidad) y para cada uno de ellos se evaluaron 20 veces los algoritmos propuestos y luego se tomó la *moda* de la mejor solución por cada instancia. A continuación se muestran los rangos de la solución propuestos por la literatura para cada uno de los problemas seleccionados y la *moda* de los resultados obtenidos por el AG y PSO respectivamente. Para mostrar en cada problema el mejor resultado obtenido entre los algoritmos, se diferenció el valor del mismo de color azul.

Tabla 4. Resultados obtenidos con AG y PSO

Instancia	Resultados de Taillard		AG	PSO
	Lower bound	Upper bound		
ta_20_5_02	1290	1359	1359	1359
ta_20_5_05	1198	1236	1241	1336
ta_20_10_01	1448	1582	1575	1575
ta_20_10_07	1388	1484	1512	1484
ta_20_20_02	1711	2100	2103	2099
ta_20_20_04	1810	2223	2235	2229
ta_50_05_05	2837	2863	2873	2863
ta_50_05_06	2793	2829	2829	2829
ta_50_10_08	2959	3039	3086	3044
ta_50_10_10	3046	3091	3112	3093
ta_100_5_01	5437	5493	5501	5493
ta_100_5_04	4963	5014	5035	5023
ta_100_10_01	5759	5770	5916	5795
ta_100_10_05	5431	5468	5679	5480

Conclusiones

Se implementó un Algoritmo Genético y la Técnica basada en Nube de Partículas. Las mismas se evaluaron tomando como casos de pruebas problemas estándar sobre el FSSP disponibles en la Internet y se estableció una comparación de los mismos llegando a las siguientes conclusiones:

- ◆ Estos métodos se proyectan como una alternativa interesante en la solución de problemas de alta complejidad matemática, y con el cual se podrían resolver problemas de mayor dificultad que los estudiados en esta investigación.
- ◆ La adaptación del Algoritmo Genético y la técnica de optimización combinatorial PSO al FSSP arrojaron resultados de excelente calidad proporcionando secuencias de trabajo dentro de los límites propuestos por la literatura especializada.
- ◆ La técnica de optimización combinatorial PSO arrojó mejores o iguales resultados para cada uno de los problemas seleccionados que el Algoritmo Genético, demostrando que la misma influye más en los resultados del FSSP.

Referencias

- ADAMS, J. The Shifting Bottleneck Procedure for job shop scheduling. 1988; 34 (3):391-401.
- ÁLVAREZ, M. D., *et al.* Simulated Annealing Heuristic For Flow Shop Scheduling Problems. *Scientia et Technica*. 2008; XIV (40):159-164.
- CAVALCANTI, M. Algoritmos Genéticos: Principios y Aplicaciones. Pontificia Universidad Católica de Rio de Janeiro, Departamento de Ingeniería Eléctrica. 2009.
- CERNY, V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization theory and Application*. 1985; 45(1):41-51.
- COMPANYYS, R. Un algoritmo branch-and-bound doble para resolver el problema flow-shop con bloqueos. IX Congreso de Ingeniería de Organización 2005 8 y 9 de septiembre de 2005 Gijón, Barcelona-España.
- DELGADO, A. E. Aplicación de Algoritmos Genéticos para la programación de tareas en una celda de manufactura. *Ingeniería e Investigación*. 2005; 25(002):24-31.
- GLOVER, F. Tabu Search: a tutorial Interfaces. 1990; 20(No. 4):74-94.
- JOHNSON, S. M. Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*. 1954; 1:402-452.
- KIRKPATRICK, S., *et al.* Optimization by Simulating Annealing. *Science*. 1983; 220(4598):661-680.

- LAGEWEGM, B. J., *et al.* Job-Shop Scheduling by Implicit Enumeration. *Management Science*. 1977; 24(4):441-450.
- LEE, G. C., *et al.* Bottleneck-focused scheduling for a hybrid flowshop. *International Journal of Production Research*. 2004; 42:165-181.
- MANNE, A. S. On the Job-Shop Scheduling Problem. [*Operations Research*]. 1960; 8 (2):219.
- MATÍA, J. Optimización de la secuenciación de tareas en taller mediante algoritmos genéticos 2010.18-28
- PINEDO, M. *Scheduling Theory, Algorithms, and Systems*. 3 edition ed. New Jersey: Prentice Hall Inc., Englewood Cliffs, U.S.A.; 2008.
- RÍOS-MERCADO, Z. An enhanced TSPbased heuristic for makespan minimization in a Flowshop with setup times. *Journal of Heuristics*. 1999; 5 (1):57-74.
- RÍOS-MERCADO, Z. Secuenciando óptimamente líneas de flujo en sistemas de manufactura. *Revista de Ingenierías*. Universidad de Nuevo León. México. 2001; IV (10):48-67.
- ŠEDA, M. *Mathematical Models of Flow Shop and Job Shop Scheduling Problems*. World Academy of Science, Engineering and Technology. 2007; 1 (31):122-127.
- TORO, M., *et al.* Adaptación de la técnica de Particle Swarm al problema de secuenciación de tareas. *Scientia et Technica UTP*. 2006; XII (32):307-313.
- TORO, M., *et al.* Algoritmo genético modificado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal - Flow Shop. *Scientia et Technica*. 2006b; XII (30):285-290.
- YAMADA, T. *Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems* [Doctor of informatics]. Kyoto, Japan: Kyoto University; 2003.57-107
- ZHANG, H. Applications of Neural Networks in Manufacturing: A state-of-the-art Survey. *International Journal of Production Research*. 1995; 33(3):705-728.