

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 20/07/2016 | Aceptado: 30/09/2016

Un algoritmo eficiente para problemas single machine con tiempos de procesamiento difusos

An efficient algorithm for single machine problems with fuzzy processing times

Edwin Lazo Eche¹, Flabio Gutiérrez Segura^{1*}, Edmundo Vergara Moreno³

^{1,2} Universidad Nacional de Piura. Urb. Miraflores s/n, Castilla, Piura, Perú. lazoedwin88@gmail.com, flabio@unp.edu.pe

³ Universidad Nacional de Trujillo. Calle Diego de Almagro 344, Trujillo, Perú. evergara@unitru.edu.pe

* Autor para correspondencia: flabio@unp.edu.pe

Resumen

Los modelos clásicos de *scheduling* en single machine han sido estudiados con diversas características de procesamiento y funciones objetivo. La mayoría de técnicas propuestas para resolver este tipo de problemas no consideraron la naturaleza imprecisa de algunas variables que intervienen en su formulación. En este trabajo se propone un algoritmo para el cálculo del máximo tiempo de culminación ponderada en problemas de single machine con tiempos de procesamiento difusos y asignación de prioridades (pesos) a los trabajos. La metodología planteada representó la imprecisión de los tiempos de procesamiento mediante números difusos triangulares y utilizó una medida de comparación robusta y flexible basada en el concepto del intervalo esperado de un número difuso. Se obtuvo un *schedule* óptimo y robusto que soporta variaciones en los tiempos de procesamiento. El *schedule* óptimo se representó en forma gráfica mediante el diagrama de Gantt difuso. El algoritmo propuesto resulta ser eficiente en pruebas para un determinado conjunto de trabajos en la búsqueda de un *schedule* óptimo para el cálculo del máximo tiempo de culminación ponderada.

Palabras clave: *scheduling* difuso, single machine, números difusos, intervalo esperado.

Abstract

The classic models of scheduling on single machine have been studied with different processing characteristics and objective functions. Most proposed techniques for solving such problems did not consider the imprecise nature of some variables involved in its formulation. This paper proposes a heuristic for calculating the maximum weighted completion time in single machine problems with fuzzy processing times and priority assignments (weights) to the jobs. The proposed methodology represented imprecise processing times by triangular fuzzy numbers

and used a measure robust and flexible comparison based on the concept of the expected interval of a fuzzy number. Optimal and robust schedule that support variations in processing times was obtained. The optimal schedule is represented graphically by fuzzy Gantt chart. The proposed algorithm proves to be efficient based on tests made for a given set of jobs with the aim of find feasible schedule to calculate the maximum weighted completion time.

Keywords: *fuzzy scheduling, single machine, fuzzy numbers, expected interval.*

Introducción

El *Scheduling* en máquinas y sistemas es la fuente de trabajos de investigación desde aproximadamente 1950. Las diferentes áreas donde se aplica tienen como común denominador la asignación de tareas, la planificación de horizontes de trabajos y la optimización de las operaciones entre recursos y tareas, entre otros. Un *scheduling* puede definirse como la tarea de determinar el inicio y finalización de cada operación (Alharkan, 2010) a ser procesada en un taller (fábrica, planta). Al resultado de esta planificación se le denomina *schedule* (secuencia de trabajos) que puede ser, por ejemplo, un horario, planes de aterrizaje o de despegue de aviones, atraques de barcos, etc. Desde el punto de vista matemático, puede entenderse como un arreglo de trabajos y operaciones que deben satisfacer ciertas condiciones de asignación y restricciones, además de satisfacer una condición de la optimalidad en los resultados.

En el *scheduling* hay una gran brecha entre la teoría y la práctica, esto se debe, principalmente al hecho que en la modelación de los problemas no se considera la incertidumbre. Las causas de la incertidumbre se deben a: la falta de información, abundancia de información, conflicto en las evidencias, ambigüedad, medidas, creencias (Zimmermann, 2000). El desarrollo de la lógica difusa, permite considerar la incertidumbre como un elemento muy importante en la formulación de los problemas de asignación. La imprecisión es una incertidumbre de tipo difuso, en el cual se posee información subjetiva del estado de ocurrencia de un determinado suceso, o quizás que el comportamiento de una determinada variable está relacionado con un conjunto de valores numéricos.

En los entornos de *scheduling*, la imprecisión puede aparecer de manera natural, sobre todo en el estado del procesamiento de los trabajos (específicamente su duración), los adelantos, retrasos, los tiempos de culminación, el cumplimiento de una restricción de precedencia entre otras. Resulta natural, por ejemplo, considerar la imprecisión en el tiempo de aterrizaje de un avión en la pista de un aeropuerto, el descargue del contenido de un barco en un muelle, el atraque del mismo. Ambas tareas coinciden en el supuesto que, tanto la pista de aterrizaje y el muelle pueden ser considerados como una máquina simple.

El desarrollo de la lógica difusa, se dio casi en paralelo a los primeros algoritmos o modelos de scheduling, pero la espera por considerar la imprecisión en los entornos de scheduling tuvo que prolongarse hasta los inicios de la década de los 90's.

Sobre fuzzy scheduling, las investigaciones se fundamentan en modelos de optimización, búsqueda de heurísticas e implementación de metaheurísticas eficientes considerando los datos de entrada como variables difusas (Wang y otros, 2002; Ahmadizar y Hosseini, 2011; Ishi y Tada, 1995).

El problema de *fuzzy single machine*, con imprecisión en los tiempos de procesamiento, ha sido abordado mediante un algoritmo que utiliza los grados de posibilidad y la necesidad de un *schedule* (Chanas y Kasperski, 2004). Este algoritmo requiere de todas las combinaciones posibles entre los trabajos y de optimizar sus medidas de posibilidad y necesidad, cuando el número de trabajos crece, se requiere demasiadas operaciones de cálculo.

En este trabajo, se consideran los problemas de *scheduling* en entornos single machine, con imprecisión en el tiempo de procesamiento de los trabajos, sin restricciones en el procesamiento de los trabajos, la función objetivo consiste en minimizar el máximo tiempo de culminación ponderada de los trabajos. Para resolver este problema, se propone un algoritmo, en el cual, la imprecisión en los tiempos de procesamiento de los trabajos se representa mediante números difusos triangulares, y se utiliza una medida de comparación robusta que permite ordenar números difusos para así obtener la secuencia óptima. A diferencia del algoritmo de Chanas y Kaspersky, no se requiere de todas las combinaciones posibles de los trabajos para determinar el *schedule* óptimo.

El algoritmo obtenido, se justifica por la misma naturaleza de los problemas de single machine, los cuales pueden extenderse a problemas de máquinas paralelas u otros entornos de máquinas, donde cada pequeño problema puede ser considerado como uno de single machine.

Metodología

Scheduling Clásico.

En su formulación, un problema de scheduling puede verse sujeto a ciertas restricciones que afecten su operatividad, por esta razón, se han desarrollado diversos métodos y algoritmos para relajar dichas restricciones. Las primeras restricciones se hallan en los diferentes recursos que intervienen, por ejemplo, las máquinas (Lawler, y otros, 1993). Una máquina es aquel recurso, disponible en todo el horizonte de tiempo sobre el cual se elabora la planificación, aunque en los últimos años esta condición clásica de disponibilidad se ha relajado (Potts, y otros, 2009). Las

maquinas desarrollan los trabajos, en algunas ocasiones una sola máquina es suficiente, en otras, un trabajo necesita ser desarrollado por varias máquinas.

Otra restricción, se da en las características de procesamiento y restricciones. En base a la combinación entre máquinas, ambientes de procesamiento y establecimiento de funciones objetivo pudo establecerse una tipología de los problemas de *scheduling*, una de ellas es la notación de Graham (Graham, 1979), $\alpha | \beta | \theta$. Las siguientes tablas presentan las características de cada elemento en esta notación.

Tabla 1. Notaciones para los entornos de máquinas comunes (α). (Alharkan, 2010)

Nombre del entorno	Símbolo / Notación	Descripción
Single machine	1	Una máquina.
Máquinas paralelas idénticas	P_m	P: máquinas paralelas. m: número de máquinas
Máquinas paralelas con diferente velocidad.	Q_m	Q: máquinas paralelas con diferentes velocidades. m: número de máquinas
Flow shop	F_m	F: Flow shop. m: número de máquinas
Job shop	J_m	J: Job shop. m: número de máquinas
Open shop	O_m	O: Open shop. m: número de máquinas

Tabla 2. Notaciones comunes para las características del procesamiento / restricciones (β). (Alharkan, 2010)

Término	Notación	Descripción
Fecha de lanzamiento	r_j	Este término indica que un trabajo no puede iniciar su procesamiento en una máquina antes de su valor r_j .
Preferencia	Prmp	Un trabajo puede interrumpirse durante su procesamiento debido a la llegada de un trabajo de alta prioridad.
Restricciones de precedencia	Prec	Cuando el inicio de un trabajo depende de la culminación de otro.
Averías	Brkdwn	Esto implica que las máquinas no están disponibles de manera continua para el procesamiento.
Recirculación	Recrc	Cuando un trabajo visita una máquinas en más de una vez.
Permutación	Prmu	El orden de procesamiento de todos los trabajos en una máquina se mantiene a lo largo del taller.

Tabla 3. Notaciones comunes para funciones objetivos de scheduling (θ). (Alharkan, 2010)

Función objetivo	Notación
Makespan	$C_{\text{máx}}$
Máximo retraso	$L_{\text{máx}}$
Tiempo de culminación total ponderada	$\sum \omega_j C_j$
Tardanza total ponderada	$\sum \omega_j T_j$
Tiempo de culminación total	$\sum C_j$

En este trabajo, el entorno de máquina es *single machine*. Se conoce un conjunto de n trabajos J_1, \dots, J_n con tiempos de procesamiento P_{ij} a ser procesados por la máquina i , pero como se trata de una sola máquina este índice puede omitirse; además los trabajos tienen pesos ω_j (prioridades que son asignadas bajo ciertas características o condiciones, por lo general determinada por expertos) tal que $\omega_j \geq 0$. El tiempo de culminación del trabajo j en un *schedule* se denota por C_j .

Sea $S = (s(1); s(2); \dots; s(n))$ un *schedule* factible, en la que cada $s(i)$ representa la posición final de un trabajo en la permutación óptima; entonces los tiempos de culminación se definen por (Chanas y Karspersky, 2004):

$$C_j = \sum_{i \in \text{pos}(j)} p_{s(i)} \quad (1)$$

El objetivo de este tipo de problemas consiste en encontrar un *schedule* que minimice $\sum \omega_j C_j$. Usando la notación de Graham, el problema a resolver en este trabajo, se denota por $1 \parallel \sum \omega_j C_j$. Una metodología clásica consiste en el criterio de las proporciones de Smith (Lawler et al, 1993), con la cual es posible encontrar un *schedule* óptimo ordenando de manera no decreciente las proporciones p_j / ω_j , esto viene dado a través del siguiente teorema.

Teorema 1. (Chanas y Kasperski, 2004). Un *schedule* S es óptimo para $1 \parallel \sum \omega_j C_j$ si y sólo si se cumple la siguiente condición:

$$\frac{P_{s(i)}}{\omega_{s(i)}} \leq \frac{P_{s(i+1)}}{\omega_{s(i+1)}} \quad (2)$$

La demostración de este teorema se basa en un intercambio de posiciones del schedule que afectan la optimalidad (Brucker, 1998; Chanas, y otros, 2004). A nivel computacional, este teorema brinda una solución en tiempo polinomial cuyo orden está dado por $O(n \log n)$ (Brucker, 1998).

Fuzzy Scheduling.

Fuzzy scheduling, es un tipo de problemas de scheduling en donde las variables de formulación, funciones objetivo o los ambientes de procesamiento presentan incertidumbre del tipo difusa. Para el desarrollo de este trabajo es necesario presentar algunos resultados básicos de esta teoría.

Definición 1. (Dubois et al, 1978). Un número difuso \tilde{A} es un conjunto $\mu_{\tilde{A}}$ de la recta real \mathfrak{R} , convexo, normalizado tal que:

- a) $\exists x_0 \in \mathfrak{R} / \mu_{\tilde{A}}(x_0) = 1$, que suele llamarse moda.
- b) $\mu_{\tilde{A}}$ es continua a trozos.

todo número difuso está caracterizado por una función de pertenencia $\mu_{\tilde{A}} : \mathfrak{R} \rightarrow [0,1]$ y además toda función de este tipo también genera un número difuso. Consideremos el número difuso $\tilde{A} = (a, c, d, b)$, donde $\mu_{\tilde{A}}(x)$ es el grado de pertenencia de x al número difuso \tilde{A} que tiene la forma:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a \\ f(x), & a \leq x < c \\ 1, & c \leq x \leq d \\ g(x), & d < x \leq b \\ 0, & x > b \end{cases} \quad (3)$$

$f(x)$ es una función semicontinua, estrictamente creciente en $a \leq x < c$ y $g(x)$ otra función estrictamente decreciente en $d < x \leq b$. (ver Figura 1).

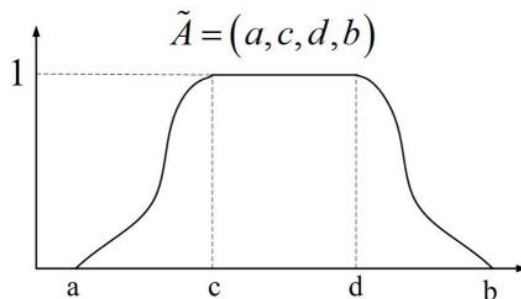


Figura 1. Representación de un número difuso.

Si las funciones semicontinuas son lineales entonces se puede hablar de dos tipos de números difusos: triangulares ($c = d$) y trapezoidales ($c \neq d$).

Dados dos números difusos \tilde{A} y \tilde{B} las operaciones aritméticas difusas se pueden realizar utilizando el principio de extensión propuesto por Zadeh (Zadeh, 1999).

$$\mu_{\tilde{A} * \tilde{B}} = \sup_{z=x*y} \min \{ \mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y) \} \quad (4)$$

En el caso de la multiplicación por un escalar $\lambda \in \mathfrak{R}$ se utiliza la definición:

$$\mu_{\lambda \tilde{A}}(x) = \begin{cases} \mu_{\tilde{A}}\left(\frac{x}{\lambda}\right), & \text{para } \lambda \neq 0. \\ 0, & \text{para } \lambda = 0. \end{cases} \quad (5)$$

Definición 2. (Chanas y Kasperski, 2004). Si $\alpha \in [0, 1]$, el α -corte de un número difuso \tilde{A} es el conjunto:

$$\tilde{A}_\alpha(x) = \begin{cases} \text{sop}(\tilde{A}) & , \text{ para } \alpha = 0 \\ \{x / \mu_{\tilde{A}}(x) \geq \alpha\} & , \text{ para } \alpha \in (0,1] \end{cases} \quad (6)$$

donde $\text{sop}(\tilde{A}) = [\inf \{x / \mu_{\tilde{A}}(x) > 0\}; \sup \{x / \mu_{\tilde{A}}(x) > 0\}] = [a, b]$. Todo α -corte define un intervalo cerrado.

Heilpern define el intervalo esperado de un número difuso \tilde{A} utilizando las siguientes funciones inversas a partir de (3) (Heilpern, 1992):

$$f_{\tilde{A}}^{-1}(\alpha) = \inf \{x / \mu_{\tilde{A}}(x) \geq \alpha\} \quad (7)$$

$$g_{\tilde{A}}^{-1}(\alpha) = \sup \{x / \mu_{\tilde{A}}(x) \geq \alpha\} \quad (8)$$

así, el intervalo esperado denotado por $EI[\tilde{A}]$ viene dado por:

$$EI[\tilde{A}] = [E_1^A, E_2^A] = \left[\int_0^1 f_{\tilde{A}}^{-1}(\alpha) d\alpha, \int_0^1 g_{\tilde{A}}^{-1}(\alpha) d\alpha \right] \quad (9)$$

El valor esperado de un número difuso, denotado por $EV[\tilde{A}]$ es el valor medio entre sus extremos, definido como:

$$EV[\tilde{A}] = \frac{E_1^A + E_2^A}{2} \quad (10)$$

Para dos números difusos \tilde{A} y \tilde{B} el intervalo esperado cumple con las condiciones de linealidad, esto es

$$EI[\tilde{A} + \tilde{B}] = EI[\tilde{A}] + EI[\tilde{B}] \quad (11)$$

$$EI[\lambda\tilde{A}] = \lambda EI[\tilde{A}] \quad (12)$$

Definición 3. (Jiménez, 1996). Para cualquier par de números difusos \tilde{A} y \tilde{B} , el grado en el cual \tilde{A} es mayor o igual que \tilde{B} es el siguiente:

$$M(\tilde{A}; \tilde{B}) = \begin{cases} 0 & , \text{ si } E_2^A - E_1^B < 0 \\ \frac{E_2^A - E_1^B}{E_2^A - E_1^B - (E_1^A - E_2^B)} & , \text{ si } 0 \in [E_1^A - E_2^B, E_2^A - E_1^B] \\ 1 & , \text{ si } E_1^A - E_2^B > 0 \end{cases} \quad (13)$$

donde $[E_1^A, E_2^A]$ y $[E_1^B, E_2^B]$ son los intervalos esperados de los números difusos \tilde{A} y \tilde{B} . Cuando $M(\tilde{A}; \tilde{B}) = 0.5$ se dice que \tilde{A} y \tilde{B} son indiferentes.

Algoritmo para $1 | \sum \omega_j \tilde{C}_j$

La imprecisión en los tiempos de procesamiento se representó mediante números difusos de tipo triangular $\tilde{P} = (p_1, p_2, p_3)$, el cual está determinado por tres cantidades: p_2 es el valor con más posibilidad de ocurrencia, p_1 y p_3 son los valores límites inferior y superior permitidos, respectivamente (ver Figura 2). Por ejemplo, estos valores límite pueden interpretarse como el más pesimista y el más optimista, en función del contexto.

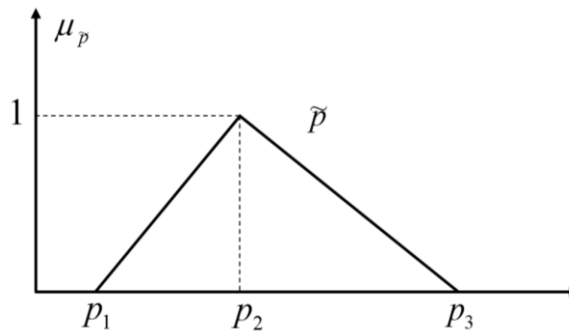


Figura 2. Número difuso triangular

El algoritmo difuso, desarrollado para resolver el problema planteado, se basó en el criterio de la proporción de Smith (Lawler et al, 1993), que resuelve el problema de *scheduling single machine*, pero sin contemplar imprecisión.

Sea $\tilde{p}_j = (p_{j1}, p_{j2}, p_{j3})$ el tiempo de procesamiento de cada trabajo $j = \{1, \dots, n\}$, las proporciones $\tilde{\rho}_j = \frac{\tilde{p}_j}{\omega_j}$ se obtuvieron partir de (5). Para determinar el intervalo esperado de $\tilde{\rho}_j$ utilizando (7) y (8) se calculó

$$f_{\tilde{\rho}_j}^{-1}(\alpha) = \frac{p_{j1} + \alpha(p_{j2} - p_{j1})}{\omega_j} \quad (14)$$

$$g_{\tilde{\rho}_j}^{-1}(\alpha) = \frac{p_{j3} - \alpha(p_{j3} - p_{j2})}{\omega_j} \quad (15)$$

Se utilizó el intervalo esperado de cada proporción como medida de comparación para ordenar los trabajos en forma no decreciente. Para cada par de proporciones, la medida en que $\tilde{\rho}_j$ es mayor que $\tilde{\rho}_k$ se determinó usando (13).

$A = (a_{ij})_{n \times n}$ es una matriz cuadrada que se obtuvo aplicando la medida de comparación entre todas las proporciones obtenidas, es decir

$$a_{ij} = \begin{cases} M(\tilde{\rho}_i, \tilde{\rho}_j), & \text{si } M(\tilde{\rho}_i, \tilde{\rho}_j) > 0.5 \\ 0, & \text{otro caso} \end{cases} \quad (15)$$

Fue necesario generar un indicador para esta matriz que establezca una relación directa con el schedule óptimo. Para ello se obtiene una secuencia de escalares λ_i , tal que:

$$\lambda_i = \sum_{j=1}^n a_{ij} \quad (16)$$

Mediante el ordenamiento de los escalares λ_i en un vector, se determina el schedule óptimo para el problema

$$1 \mid \left| \sum \omega_j \tilde{C}_j \right|$$

El procedimiento para determinar el schedule óptimo está dado por el siguiente algoritmo:

ALGORITMO PARA $1 \mid \left| \sum \omega_j \tilde{C}_j \right|$

Entrada. n : número de tareas,

$\tilde{p}_1, \dots, \tilde{p}_n$: Tiempos de procesamiento difuso de los trabajos

$\omega_1, \dots, \omega_n$: Pesos de los trabajos

Salida. S : Schedule factible, $\sum \omega_j \tilde{C}_j$

Paso 1: Calcular las proporciones entre los trabajos.

Para $i = 1$ hasta n **hacer**

$$\tilde{\rho}_i = \frac{\tilde{p}_i}{\omega_i};$$

$$EI[\tilde{\rho}_i] = [E_1^{\tilde{\rho}_i}, E_2^{\tilde{\rho}_i}];$$

Fin_para

Paso 2: Aplicar la medida de comparación entre las proporciones de los trabajos

Para $i = 1$ hasta n **hacer**

Para $j = 1$ hasta n **hacer**

Si $E_1^{\tilde{\rho}_i} - E_2^{\tilde{\rho}_j} < 0$ **entonces**

$$M[\tilde{\rho}_i, \tilde{\rho}_j] = 0$$

Si no

Si $0 \in [E_1^{\tilde{\rho}_i} - E_2^{\tilde{\rho}_j}, E_2^{\tilde{\rho}_i} - E_1^{\tilde{\rho}_j}]$ **entonces**

$$M[\tilde{\rho}_i, \tilde{\rho}_j] = \frac{E_2^{\tilde{\rho}_i} - E_1^{\tilde{\rho}_j}}{E_2^{\tilde{\rho}_i} - E_1^{\tilde{\rho}_j} - (E_1^{\tilde{\rho}_i} - E_2^{\tilde{\rho}_j})}$$

Si no

$$M[\tilde{\rho}_i, \tilde{\rho}_j] = 1$$

Fin_si

Fin_si

Fin_para

Fin_para

Paso 3: Generar la matriz de comparaciones y calcular los valores $\lambda(i)$

Para $i = 1$ hasta n **hacer**

```

 $\lambda(i) = 0;$ 
Para  $j = 1$  hasta  $n$ 
    Si  $M[\tilde{\rho}_i, \tilde{\rho}_j] > 0.5$  entonces
         $A(i, j) = M[\tilde{\rho}_i, \tilde{\rho}_j]$ 
    Si no
         $A(i, j) = 0$ 
    Fin_si
         $\lambda(i) = \lambda(i) + A(i, j)$ 
Fin_para
Fin_para
    
```

Paso 4: Ordenar en forma ascendente los elementos del vector $\lambda = (\lambda(1); \dots; \lambda(n))$ y guardarlo en el vector $V = (v(1); \dots; v(n))$.

Paso 5: Obtención del schedule óptimo

Hacer $S = (s(1); \dots; s(n))$, con $s(k) = pos(v(k))$, $k = 1, \dots, n$,

donde $pos(v(k))$ indica la posición de $v(k)$ en el vector λ .

Paso 6: Se calculan los tiempos de culminación.

Para $j = 1$ hasta n **hacer**

Calcular \tilde{C}_j usando (1)

Fin_para

Paso 7: Calcular la función objetivo $\sum \omega_j \tilde{C}_j$

Paso 8: Imprimir S , $\sum \omega_j \tilde{C}_j$

Resultados y discusión

El algoritmo se implementó en C++. Consideremos el siguiente caso de estudio, un problema de 10 trabajos a ser procesados en una máquina, con sus respectivos tiempos de procesamiento difusos y pesos (ver Tabla 4)

Tabla 4. Caso de estudio con 10 trabajos

Trabajo	Tiempo de procesamiento \tilde{P}_j	Pesos ω_j
1	(12,14,17)	5
2	(54,63,72)	2
3	(40,46,49)	8
4	(85,98,111)	1
5	(38,46,51)	8
6	(27,30,32)	5
7	(94,96,98)	2

8	(13,15,16)	3
9	(63,71,79)	4
10	(29,32,35)	6

Por ejemplo, para el trabajo 1, la duración del tiempo de procesamiento estará entre las 12 y 17 unidades de tiempo, con mayor posibilidad en las 14 unidades.

Al ejecutar el algoritmo, mediante el paso 1, se obtuvo las proporciones de los trabajos. En el paso 2, se les aplicó la medida de comparación difusa. Con el paso 3, se obtuvo el vector $\lambda = (0; 7; 2,8039; 9; 2,74603; 4,5; 8; 1; 6; 1,9167)$. De acuerdo al paso 4, con los $\lambda(i)$ ordenados en forma ascendente, se formó el vector

$V = (\lambda(1); \lambda(8); \lambda(10); \lambda(5); \lambda(3); \lambda(6); \lambda(9); \lambda(2); \lambda(7); \lambda(4)) = (0; 1; 1,9167; 2,746; 2,8039; 4,5; 6; 7; 8; 9)$

De acuerdo al paso 5, el *schedule* óptimo que minimiza el máximo tiempo de culminación ponderada es:

$$S = (1; 8; 10; 5; 3; 6; 9; 2; 7; 4)$$

En la Tabla 5, obtenida del paso 6, se muestran los tiempos de culminación para cada trabajo.

Tabla 5. Tiempos de culminación para cada trabajo

Trabajo	Tiempo de procesamiento \tilde{P}_j	Tiempo de culminación \tilde{C}_j
1	(12, 14, 17)	(12, 14, 17)
2	(54, 63, 72)	(276, 317, 351)
3	(40, 46, 49)	(132, 153, 168)
4	(85, 98, 111)	(455, 511, 560)
5	(38, 46, 51)	(92, 107, 119)
6	(27, 30, 32)	(159, 183, 200)
7	(94, 96, 98)	(370, 413, 449)
8	(13, 15, 16)	(25, 29, 33)
9	(63, 71, 79)	(222, 254, 279)
10	(29, 32, 35)	(54, 61, 68)

El *schedule* obtenido para el caso de estudio se representó mediante el diagrama de Gantt difuso (ver Figura 4)

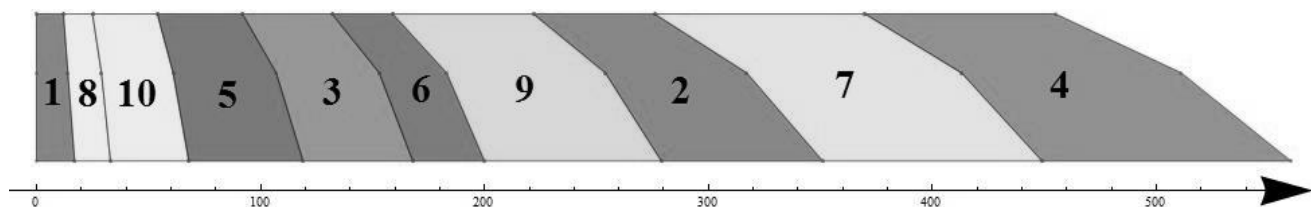


Figura 3. Diagrama de Gantt difuso para el caso de estudio

El máximo tiempo de culminación ponderada quedó minimizado por el *schedule S* y expresado en el número difuso triangular.

$$\sum_{j=1}^{10} w_j \tilde{C}_j = (5681, 6505, 7164)$$

Es decir, al procesar los 10 trabajos en una máquina, el máximo tiempo de culminación ponderada puede ocurrir entre 5 681 y 7 164 unidades de tiempo, con más posibilidad de ocurrencia en 6 505 unidades de tiempo. Los tiempos de culminación también son de manera triangular, esto se evidencia en la deformación en los lados del polígono en la representación clásica del diagrama de Gantt. Puesto que se consideró la imprecisión en el tiempo de procesamiento de los trabajos, el resultado de este algoritmo también presenta un *makespan* robusto, el cual quedó determinado por el tiempo de culminación del trabajo con mayor proporción, para el caso de estudio, el trabajo 4.

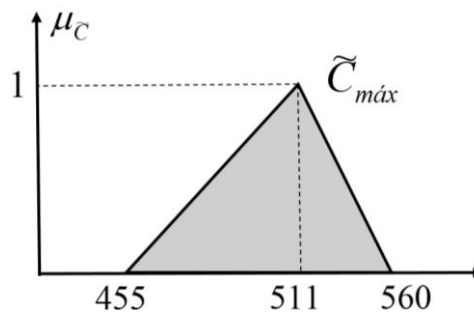


Figura 4. Makespan difuso para el caso de estudio

La robustez del makespan permite soportar la variabilidad de los tiempos de procesamiento de los trabajos en el *schedule* óptimo, siendo el makespan con más posibilidad de ocurrencia de 511 unidades de tiempo.

Se ha tomado como caso de estudio un problema con 10 trabajos, pero el algoritmo se comporta de manera similar para una cantidad mayor de trabajos, demostrando en todas ellas su eficiencia para calcular el máximo tiempo de culminación ponderada y, sobre todo, la facilidad con la que se manipula la imprecisión en estos entornos.

Conclusiones

El algoritmo, permite manejar la imprecisión en los tiempos de procesamiento de los trabajos en un problema de single machine donde intervienen prioridades o pesos en los trabajos. La solución óptima obtenida se refleja en un *schedule* robusto que soporta variaciones en los tiempos de procesamiento. De esta manera se proporciona una nueva metodología que puede modelar algunos problemas en donde una máquina sea capaz de procesar todos los trabajos

que se presentan en un taller. En trabajos futuros se pretende 1) Implementar un algoritmo eficiente para problemas de single machine con restricciones de precedencia en el procesamiento de los trabajos, el cual corresponde a un problema NP-hard. 2) Extender los algoritmos difusos para single machine, a problemas de máquinas paralelas u otros entornos de máquinas.

Agradecimientos

Para la realización de este trabajo se ha contado con ayuda del Fondo para la Innovación Ciencia y Tecnología del Perú FINCyT (Proyecto PIBA-2-P-069-14).

Referencias

- AHMADIZAR, F.; HOSSEINI, L. Single-machine scheduling with a position-based learning effect and fuzzy processing times. *The International Journal of Advanced Manufacturing Technology*, 2011, 56(5): p. 693-698.
- ALHARKAN, I. Algorithms for sequencing and scheduling. Industrial Engineering Department, King Saud University, Riyadh, Saudi Arabia, 2010.
- BRUCKER, P. *Scheduling Algorithms*. s.l.: Springer-Verlag, Osnabrück, Germany, 2004.
- CADENAS, J. M.; VERDEGAY, J. L. *Modelos de optimización con datos imprecisos*. Universidad de Murcia, Servicio de Publicaciones, España, 1999.
- CHANAS, S.; KASPERSKI, A. Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates. *Engineering Applications of Artificial Intelligence*, 2001, 14(3): p. 377-386.
- CHANAS, S; KASPERSKI, A. Possible and necessary optimality of solutions in the single machine scheduling problem with fuzzy parameters. *Fuzzy Sets and Systems*, 2004, 142(3): p. 359-371.
- DUBOIS, D.; PRADE, H. Operations on fuzzy numbers. *International Journal of systems science*, 1978, 9(6), p. 613-626.
- GRAHAM, R. L., et al. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 1979, 5, p. 287-326.
- HEILPERN, S. The expected value of a fuzzy number. *Fuzzy sets and Systems*, Wroclaw, Polonia, 1992. págs. 81-86.

ISHII, H.; TADA, M. Single machine scheduling problem with fuzzy precedence relation. *European Journal of Operational Research*, 1995, 87(2): p. 284-288.

JIMÉNEZ, M. Ranking fuzzy numbers through the comparison of its expected intervals. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, España, 1996. p. 379-388.

LAWLER, E. L., et al. Sequencing and scheduling: Algorithms and complexity. *Handbooks in operations research and management science*, 1993, 4, p. 445-522.

POTTS, C. N.; STRUSEVICH, Vitaly A. Fifty years of scheduling: a survey of milestones. *Journal of the Operational Research Society*, 2009, 60(1): p. 41-68.

WANG, C, et al. The single machine ready time scheduling problem with fuzzy processing times. *Fuzzy sets and systems*, 2002, 127(2): p. 117-129.

ZADEH, L. A. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1999, 100, p. 9-34.

Z

I

M

M

E

R

M

A

N

N

,

H

.

A

n

a

Editorial "Ediciones Futuro"

Universidad de las Ciencias Informáticas. La Habana, Cuba

rcci@uci.cu

l

i