

**METHODOLOGY FOR DEVELOPING A SCHEDULER FOR A FLEXIBLE
MANUFACTURING CELL USING HEURISTIC SEARCH****METODOLOGÍA PARA EL DESARROLLO DE UN PLANIFICADOR DE
TAREAS PARA UNA CELDA FLEXIBLE DE MANUFACTURA UTILIZANDO
BÚSQUEDA HEURÍSTICA**

Esp. Camilo Arturo Lagos Mora*, **MSc. José Tomas Buitrago Molina****
PhD. Ricardo Timarón***

* **Departamento de Sistemas, Universidad de Nariño**, Pasto, Colombia, Estudiante
Maestría Universidad del Valle, E-mail: clagos@udenar.edu.co

** **Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle**, Cali,
Colombia, E-mail: buitrago@univalle.edu.co

*** **Departamento de Sistemas, Universidad de Nariño**, Pasto, Colombia,
E-mail: ritimar@udenar.edu.co

Abstract: The principal difficulty which the planning of tasks faces for the development of pieces or products in a cell of flexible manufacture (FMC) is the complexity of the relations of his elements and the quantity of pieces tried in it that, though it is minor that those who are made of cells of dedicated use, it does not stop being a problem at the moment of planning the actions for the accomplishment of each one of them. In this article one proposes a methodology of planning of tasks for a flexible cell of manufacture using heuristic search with partial solutions, which it is a modification of the technique of search first the best, and it is based that to produce n pieces of ideal form is necessary to pass for the production of $n-1$ ideal pieces. This concept allows to eliminate all the unnecessary states to produce n pieces departing from the partial solution of the production of $n-1$ pieces and assuming therefore the solution of them $n-1$ pieces as initial state for the production of n -pieces.

Keywords: Scheduler, flexible manufacturing cell, state-space, heuristic search, heuristic function.

Resumen: La principal dificultad a la que se enfrenta la planificación de tareas para el desarrollo de piezas o productos en una celda de manufactura flexible (FMC) es la complejidad de las relaciones de sus elementos y la cantidad de piezas procesadas en ella que, aunque es menor que las que se fabrican en celdas de uso dedicado, no deja de ser un problema a la hora de planificar las acciones para la realización de cada una de ellas. En este artículo se propone una metodología de planificación de tareas para una celda flexible de manufactura utilizando búsqueda heurística con soluciones parciales, que es una modificación de la técnica de búsqueda primero el mejor, y se basa en que para producir n piezas de forma óptima es necesario pasar por la producción de $n-1$ piezas óptimas. Este concepto permite eliminar todos los estados innecesarios para producir n piezas partiendo de la solución parcial de la producción de $n-1$ piezas y asumiendo por lo tanto la solución de las $n-1$ piezas como estado inicial para la producción de n -piezas.

Palabras clave: Planificador, celda de manufactura flexible, espacio de estados, búsqueda heurística, función heurística.

1. INTRODUCCIÓN

Una de las principales actividades al querer lograr un objetivo es la planificación de las acciones que se deben desarrollar en su consecución. Esto se cumple no solo en la industria si no también en la vida diaria. Un ejemplo típico es la planificación de tareas para el desarrollo de piezas o productos en una celda de manufactura flexible (FMC). La principal dificultad a la que se enfrenta ésta tarea es la complejidad de las relaciones de sus elementos y la cantidad de piezas procesadas en ella que, aunque es menor que las que se fabrican en celdas de uso dedicado, no deja de ser un problema a la hora de planificar las acciones para la realización de cada una de ellas.

Se estima que aproximadamente el 90% de las piezas producidas en una fábrica se hacen en lotes de 50 o menos. Del tiempo que pasan estas en el taller, la mayoría es perdido en mover y esperar, resultando en una baja productividad, encontrando como resultados costos de producción de 10 a 100 veces más que en la producción masiva [19].

El objetivo de la planificación de tareas en FMC es tratar de alcanzar y superar la eficiencia de producción de los sistemas de producción masiva de tal manera que los costos sean iguales o inferiores a estos.

Existen varios trabajos realizados enfocados a desarrollar algoritmos de planificación de tareas para celdas de manufactura y diferentes tipos de procesos, los cuales han mostrado sus bondades para esta tarea como el de Chen y Song 2008 [3] en el cual plantean una estrategia híbrida de dos etapas para la minimización del *makespan*. Lee y otros [5] propusieron un esquema para procesos *flow shop* con permutación para m-máquinas utilizando el algoritmo *branch-and-bound*. Rego y Duarte [13] desarrollaron un algoritmo que combina el clásico algoritmo de cuello de botella y un procedimiento de búsqueda dinámico y adaptativo de vecindad para problemas *job shop*. Rad, Ruiz y Boroojerdian [10] proponen 5 métodos nuevos que mejoran la heurística NHE la cual ha sido considerada como una heurística de alta eficiencia para *flowshop*, y así se podría hablar de otras técnicas [1, 4], donde su principal objetivo es la minimización del tiempo de producción o minimización del *makespan* como es más comúnmente conocido [16].

No existe mucho trabajo en tratar de desarrollar una metodología, en donde se puedan implementar estos trabajos relacionados con anterioridad.

Este trabajo se centra en el desarrollo de una metodología para el desarrollo de un planificador de tareas que permita implementar y probar una diversidad de algoritmos y principalmente funciones heurísticas sobre un esquema de una búsqueda heurística.

2. CONCEPTOS PRELIMINARES

Dentro de los conceptos preliminares a tener en cuenta para la comprensión de la metodología se tienen:

- *Espacio de estados*: Es un modelo que permite representar cualitativamente las situaciones que se pueden presentar durante la búsqueda de una solución a un problema. Las variables específicas de este modelo son el estado inicial y el estado objetivo, que representan las situaciones en la cual comienza el problema y donde termina el problema respectivamente.
- *Búsqueda heurística*: Es una técnica de búsqueda que permite encontrar un camino hacia una solución de forma eficiente debido a la incorporación de una función heurística.
- *Función heurística*: Es una función de evaluación que permite ponderar a un estado para determinar su bondad [6].

3. METODOLOGÍA PARA LA PLANIFICACIÓN DE TAREAS EN UNA FMC

La metodología planteada para la solución del problema de planificación en FMC sigue los siguientes pasos:

3.1 Formular el modelo matemático para la celda de manufactura

Un aspecto importante a la hora de querer planificar las actividades en una FMC es su modelaje. El modelo será vital a la hora de querer aplicar la técnica de inteligencia artificial escogida para realizar la planificación. En lo que se refiere ya directamente al modelo propuesto, se tiene en cuenta los elementos mínimos necesarios para la planificación adecuada de las tareas de una celda de manufactura y que además permita una interacción fácil con otros elementos, de tal forma

que permitan simular de forma más fiel a la realidad el comportamiento de esta. Los elementos tenidos en cuenta son:

- Estado de la máquina (ocupada o no).
- Tiempo en que una máquina empieza a realizar una tarea.
- Tiempo en el que la máquina termina de realizar la tarea.
- Tipo de pieza que entra y tipo de pieza que sale de una máquina herramienta
- Número de la pieza
- Almacenes de entrada y salida

Para un robot también es importante definir las máquinas con las que interactúa.

- Máquina origen.
- Máquina destino

Teniendo en cuenta estos aspectos, un modelo básico tendrá la siguiente forma:

$$\begin{aligned} &M(\text{Ocp}, \text{Ti}, \text{Tf}, \text{Pin}, \text{Pout}, \text{Pieza}), \\ &R(\text{Ocp}, \text{Ti}, \text{Tf}, \text{Morg}, \text{Mdt}, \text{Pin}, \text{Pout}, \text{Pieza}), \\ &A(\text{In}, \text{Pi}, \text{Pe}, \text{Pp}), \text{Aout}(\text{Pi}, \text{Pe}, \text{Pp}) \end{aligned}$$

Donde:

- M es una máquina herramienta de la celda de manufactura,
- R es un robot de la celda de manufactura,
- Ain es el almacén de entrada,
- Aout es el almacén de salida.

Los parámetros asociados a cada uno de los ítems representan lo siguiente:

- Ocp: Especifica si la máquina está ocupada o no (se podría utilizar para esta variable 0 si está libre o 1 si está ocupada).
- Ti: Es el tiempo en que la operación se comienza a realizar.
- Tf: Es el tiempo que la operación termina.
- Pin: Es la pieza de entrada.
- Pout: Es la pieza de salida.
- Pieza: Es el nombre de la pieza procesada (consecutivo).
- Morg: Máquina herramienta de la cual tomará la pieza el robot.
- Mdt: Máquina herramienta a la cual entregará la pieza el robot.
- Pi: Son las piezas iniciales en el almacén de entrada y en el almacén de salida.
- Pe: Son las piezas existentes actualmente en el almacén de entrada o las piezas que han sido terminadas en el almacén de salida.
- Pp: Son el número de piezas que se deben procesar.

La representación del estado puede ser ampliada para de tal manera que permita especificar elementos adicionales como la trayectoria de algunas máquinas, que permita evitar posibles colisiones con los elementos de la celda.

3.2 Definir el problema de producción como una búsqueda en un espacio de estados

Para definir el problema como una búsqueda en un espacio de estados se debe:

1. *Definir el espacio de estados:* El espacio de estados igual al modelo de la celda de manufactura.
2. *Definir el estado inicial:* Donde se reemplazan las condiciones iniciales del problema en el espacio de estados.
3. *Definir estado final:* Donde se reemplazan las condiciones finales del problema en el espacio de estados. Se asume como una situación en la que se tenga las piezas deseadas procesadas.
4. *Definir operadores:* Los operadores permiten cambiar de estado y están definido por las tareas que pueden realizar cada una de las máquinas. Por ejemplo, se podría definir el siguiente operador

Si la máquina herramienta M1 está libre y el robot R1 está libre y el número de piezas requeridas no se cumple, el robot R1 toma una pieza del almacén 1 para llevarlo a M1. Dicho matemáticamente se tendrá:

Si $M1(\text{Ocp})=0$ y $R1(\text{Ocp})=0$ y $A(\text{In})>0$ y $(A(\text{Pp})-A(\text{Pe})) < \text{PiezasProducir}$,

Entonces $R1(1, \text{Ti}=\text{mayor}[R1(\text{Tf}), M1(\text{Tf})], \text{Tf}=\text{Ti}+\text{T1}, \text{Morg}=\text{A1}, \text{Mdst}=\text{M1}), A(\text{Pe})=A(\text{Pe})-1$.

Claro está que los operadores deberán expresarse de manera general de tal forma que puedan ser aplicados a una gran cantidad de celdas.

3.3 Definir la función heurística para que guíe la búsqueda

La función heurística tiene como fin asignar o quitar puntos a un estado de acuerdo a sus características.

En lo que se refiere a la función heurística, existen muchas estrategias para tratar de obtener el mejor camino posible de una forma más directa, como lo es la “distancia cuadrática”, división del problema, entre otros [12, 14], y tratar de reducir al máximo la explosión combinatoria, claro está que la eficacia de cada uno de ellos dependen del problema particular. Para este trabajo se utilizó

como estrategia el mayor tiempo empleado y la profundidad del camino seguido, dando como resultado la siguiente función heurística.

$$F_h = T_f \text{ máximo} - P_{fd} * (30 \text{ a } 70\% \text{ tiempo tarea más rápida})$$

Donde:

T_f máximo: Es el mayor valor T_f (tiempo en que termina el proceso) de todas las máquinas de la FMC.

P_{fd} : Es la profundidad de la ruta seguida

Para problemas poco complejos o con un bajo número de piezas, una heurística expresada como sigue es adecuada.

$$F_h = T_f \text{ máximo}$$

3.4 Utilizar una técnica de búsqueda para solucionar el problema

En este paso simplemente se selecciona una técnica de búsqueda heurística en la cual se implementa la función heurística antes definida y se procede a encontrar un camino desde el estado inicial hasta el estado objetivo.

Uno de los mayores problemas a los que se enfrenta en esta etapa al tratar de desarrollar un planificador de tareas es la explosión combinatoria, lo que hace muchas veces imposible la utilización de algoritmos matemáticos que buscan optimizar el tiempo empleado. Debido a esto han surgido algunas técnicas como: Gráfica de Precedencias [11] que permite definir planes en los cuales se contempla situaciones en las que se pueden realizar varias tareas paralelamente pero no situaciones en donde diferentes máquinas pueden realizar la misma tarea. Otras técnicas que intentan combatir este problema, son las de inteligencia artificial, como las redes neuronales, algoritmos genéticos [8] y la búsqueda heurística como *enfriamiento simulado* [7], en la cual se evita la explosión combinatoria siguiendo un único camino a la vez, pero implica una rigurosa ordenación de las reglas del modelo de acuerdo a su importancia en la solución, *búsqueda tabú* [17], entre otras. Otro aspecto importante, es que la búsqueda en un espacio de estados es bastante general y permite la inclusión de otras técnicas en su estructura, considerándolas como heurísticas que permitirán orientar la búsqueda.

Dentro de los métodos de búsqueda heurística, el método *primero el mejor*, es uno de los mejores, pues combina las ventajas que posee las búsquedas

por profundidad y la búsqueda por anchura y la estrategia de soluciones parciales permite combatir la explosión combinatoria.

Blai y Hector [2] en su estudio realizado compararon el rendimiento de los planificadores de búsqueda heurística y en especial los que utilizan el algoritmo de búsqueda primero el mejor, frente a otros planificadores específicos como GRAPHPLAN y demostraron su gran competitividad de este frente a los demás.

A pesar de que a estos tipos de búsqueda se les critica por obtener buenas soluciones y no óptimas, esto depende principalmente del tipo de función heurística empleada. Obviamente que una función heurística orientada a tener una buena solución será más eficiente en términos de tiempo y costo computacional que una heurística orientada a obtener la mejor solución.

Dentro del método de búsqueda primero el mejor existen dos variaciones: los grafos "O" y el algoritmo A^* , de estos dos siendo el mejor el algoritmo A^* pues se tendrá en cuenta el costo del camino seguido como una forma adicional para reducir la explosión combinatoria.

Para esta metodología se desarrolló una técnica de búsqueda heurística denominada BPMSP, la cual es una modificación de la técnica de búsqueda primero el mejor, propuesta para este trabajo y que permite reducir la explosión combinatoria.

Algoritmo propuesto:

- a. Comenzar definiendo el estado inicial y el estado final de la FMC.
- b. Hacer al estado inicial el estado Actual.
- c. Definir el número de piezas para la solución parcial.
- d. Mientras que el número de piezas producidas del estado actual sea menor que el número de piezas a producir haga:
 - d.1 Mientras que el número de piezas producidas del estado actual sea menor que el número de piezas a producir en la solución parcial haga:
 - d.1.1 Si el estado actual no ha sido explorado antes hacer:
 - d.1.1.1 Ingresar al estado actual en una estructura denominada *explorados*.
 - d.1.1.2 Generar los sucesores del estado actual y guardarlos en una estructura denominada *abiertos* ordenándolos de acuerdo a su

importancia (menor valor heurístico).

d.1.1.3 Sacar el mejor estado de *abiertos* y convertirlo en el estado actual.

d.2 Si el número de piezas producidas del estado actual es menor que el número de piezas a producir hacer:

d.2.1 Construir la solución parcial desde la lista de cerrados

d.2.2 Tomar de la solución parcial el estado en el cual ingreso el último elemento a la FMC, eliminando los sucesores a este y convertirlo en el nuevo estado inicial para la producción de la siguiente solución parcial.

d.2.3 Limpiar la lista de abiertos y cerrados.

d.2.4 Pasar los estados que quedaron de la solución parcial a la lista de cerrados.

3.5 Generar el plan de tareas utilizando la solución anteriormente obtenida

Encontrado un camino desde el estado inicial al estado objetivo solo resta tomarlo y extraer de este las máquinas que interactuaron en la planificación, definiendo sus tiempos de inicio y terminación y sus máquinas de origen y destino. Esta información puede ser consignada en un archivo o hoja de cálculo, la cual puede ser fuente de información para el control de la celda.

4. RESULTADOS

Para probar la validez del modelo se desarrollo el software denominado K'Axllay, en el cual se implementaron varios modelos de celdas de manufactura, con los siguientes resultados:

a. Se implementó la celda de manufactura de la figura 1.

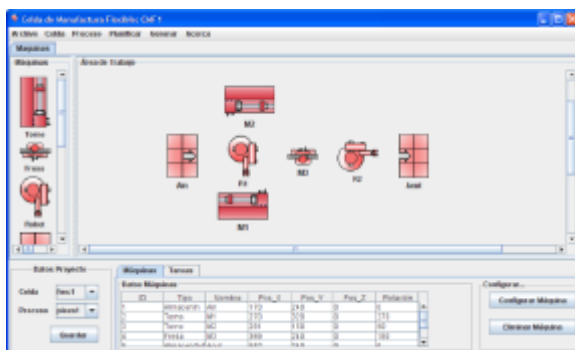


Fig. 1. Celda de manufactura flexible

Donde M1 y M2 son máquinas que pueden realizar la misma tarea. El modelo para la FMC será el siguiente:

M1(Ocp,Ti,Tf,Pin,Pout,Pieza),

M2(Ocp,Ti,Tf,Pin,Pout,Pieza),

M3(Ocp,Ti,Tf, Pin,Pout,Pieza),

R1(Ocp,Ti,Tf,Morg,Mdtn,Pin,Pout,Pieza),

R2(Ocp,Ti,Tf, Morg,Mdtn,Pin,Pout,Pieza),

Ain(Pi,Pe,Pp), Aout(Pi,Pe,Pp)

El conjunto de operaciones a realizar se muestran en la tabla 1, el cual es un tipo de problema flow shop donde cada trabajo o tarea tiene su ruta predeterminada a seguir y las operaciones deben seguir un orden dado [9], pero considerando máquinas paralelas.

Tabla 1. Conjunto de operaciones para fabricar una pieza en la FMC de la figura 1.

Trabajos	Operaciones	Mejecedora	Monigen	Mdestino	Tiempo
J1	O ₁₁	R1	Ain	M1	10
	O ₁₁	R1	Ain	M2	18
	O ₁₂	M1	0	0	20
	O ₁₂	M2	0	0	15
	O ₁₃	R1	M1	M3	15
	O ₁₃	R1	M2	M3	10
	O ₁₄	M3	0	0	25
	O ₁₅	R2	M3	Aout	10

El resultado obtenido al aplicar la metodología propuesta y que se implemento en un software para la fabricación de 3 piezas se muestra en la tabla 2.

Tabla 2. Actividades que debe desarrollar las máquinas de la celda para la producción de 3 piezas

Planificación para la producción de 3 piezas								
Máquina	Ocupada	Tiempo ini	Tiempo fin	Máquina o	Máquina d	Pieza In	Pieza Out	PiezaNo.
R1	1	0	10	Ain	M1	1	1	1
M1	1	10	30	0	0	1	11	1
R1	1	10	28	Ain	M2	1	1	2
M2	1	28	43	0	0	1	11	2
R1	1	30	45	M1	M3	11	11	1
M3	1	45	70	0	0	11	112	1
R1	1	45	55	Ain	M1	1	1	3
M1	1	55	75	0	0	1	11	3
R1	1	60	70	M2	M3	11	11	2
M3	1	70	95	0	0	11	112	2
R2	1	70	80	M3	Aout	112	1	1
R1	1	80	95	M1	M3	11	11	3
M3	1	95	120	0	0	11	112	3
R2	1	95	105	M3	Aout	112	1	2
R2	1	120	130	M3	Aout	112	1	3

b. Se implementó la celda de manufactura de la figura 2, el cual es un ejemplo de problema de planificación *open shop*.

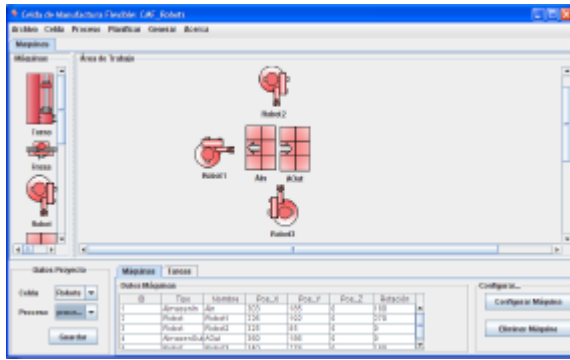


Fig. 2. Celda de manufactura flexible (open shop)

En este tipo de problemas todas las máquinas pueden realizar todas las operaciones y las operaciones pueden empezar en cualquier orden.

La tabla 3 muestra las operaciones y los tiempos para la celda de la figura 2.

Tabla 3. Tabla de operaciones y tiempos respectivos para la FMC de la figura 2

Trabajos	Operaciones	M1	M2	M3
J1	O ₁₁	5	3	2
	O ₁₂	1	2	2

La solución al problema de planificación para la producción de 5 piezas puede verse en la tabla 4 y en la figura 3.

Tabla 4. Conjunto de operaciones para fabricar una pieza en la FMC de la figura 2

Planificación para la producción de 5 piezas						
Máquina	Ocupada	Tiempo inicial	Tiempo final	Máquina orig	Máquina dest	PiezaNo.
Robot2	1	0	3	Aln	Robot1	2
Robot1	1	0	1	Aln	Robot3	1
Robot3	1	1	3	Robot1	AOut	1
Robot1	1	2	3	Aln	Robot3	3
Robot2	1	3	6	Aln	Robot1	4
Robot3	1	3	5	Robot1	AOut	3
Robot1	1	3	4	Robot2	AOut	2
Robot1	1	4	5	Aln	Robot3	5
Robot3	1	5	7	Robot1	AOut	5
Robot1	1	6	7	Robot2	AOut	4

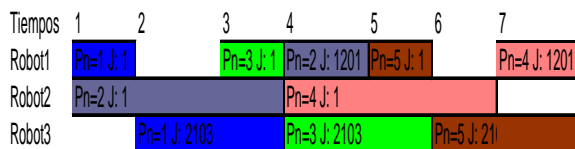


Fig. 3. Diagrama de barras para la distribución de tiempos y operaciones de las máquinas para la fabricación de 5 piezas en la celda de la figura 2.

Como se puede observar los resultados obtenidos son buenos, lo que valida la utilidad de la metodología y su utilidad como herramienta para el desarrollo de una FMC y para su análisis.

5. CONCLUSIONES

La metodología propuesta permite la implementación de diversos tipos de problemas de planificación de una manera rápida y sencilla, probando así la validez de esta metodología.

La metodología propuesta y la herramienta software desarrollada es general y puede utilizarse como una herramienta de investigación en la cual se pueden validar diversas propuestas de técnicas de búsqueda heurística y de funciones heurísticas, así como analizar el comportamiento de las FMC.

Un aspecto importante de este trabajo es que se logró reducir la explosión combinatoria, la cual crece considerablemente al tener más elementos en la FMC que puedan realizar la misma tarea (mayor número de combinaciones), con la estrategia de soluciones parciales propuesta.

La manufactura integrada por computadora (CIM) puede beneficiarse de la tarea desarrollada por el algoritmo propuesto, ayudando especialmente a la planificación de procesos asistido por computadora (CAPP), pues le permitirá establecer la manera más adecuada y eficiente el conjunto de operaciones o actividades que debe desarrollar la FMC para la fabricación de las piezas.

REFERENCIAS

- [1]. Bellanger, A., y A. Oulamara. «Scheduling hybrid flowshop with parallel batching machines and compatibilities.» *Computers & Operations Research* 36 (2009): 1982 - 1992.
- [2]. Bonet, Blai, y Hector Jennifer. «Planning as Heuristic Search.»
- [3]. Chen, Feng, y Kailei Song. «Minimizing makespan in two-stage hybrid cross docking scheduling problem.» *Computers & Operations Research*, n° 36 (2008): 2066-2073.
- [4]. Erhan Kesena, Saadettin, M.Duran Toksarib, Zülal Güngör, y Ertan Güner. «Analyzing the behaviors of virtual cells (VCs) and traditional manufacturing systems: Ant colony optimization (ACO)-based metamodels.» *Computers & Operations Research*, n° 36 (2009): 2275-2285.
- [5]. Lee, Wen-Chiung, Chin-Chia Wu, Yu-Hsiang Chung, y Han-Chu Liu. «Minimizing the total completion time in

- permutation flow shop with machine-dependent job deterioration rates.» *Computers & Operations Research*, n° 36 (2009): 2111-2121.
- [6]. Nilsson, Nils J. *Inteligencia Artificial. Una nueva síntesis, Primera edición*. Madrid: McGraw Hill, 2001.
- [7]. Onwubolu, G. C., Y. M. Mutingi. «A genetic algorithm approach to cellular manufacturing systems.» *Computers & Industrial Engineering*, 2001.
- [8]. Pan, Jason Chao-Hsien, y Han-Chiang Huang. «A hybrid genetic algorithm for no-wait job shop scheduling problems.» *Expert Systems with Applications* (Elsevier), n° 36 (2008): 5800-5806.
- [9]. Pinedo, Michael L. *Scheduling. Theory, Algorithms, and Systems*. New York: Springer, 2008.
- [10]. Rad, Shahriar Farahmand, Rubén Ruiz, y Naser Boroojerdian. «Newhigh performing heuristics for minimizing makespan in permutation flowshops.» *Omega*, n° 37 (2009): 331 – 345.
- [11]. RAMOS, Carlos, Joao ROCHA, Y Zita VALE. «On the complexity of precedence graphs for assembly and task planning.» *Computers in Industry*, n° 36 (1998).
- [12]. Ramos, Silvia A. «ApuntesHeurísticas.pdf.» Modelos y Optimización I. Noviembre de 2007.
- [13]. Rego, Cesar, y Renato Duarte. «A filter-and-fan approach to the job shop scheduling problem.» *European Journal of Operational Research*, n° 194 (2009): 650–662.
- [14]. Soto, Jesús. «T2-III.» *Inteligencia artificial e ingeniería de conocimiento - estrategias de resolución de problemas: búsqueda informada*.
<http://docs.google.com/gview?a=v&q=cache:o45KBMaOrhAJ:imai-software.com/moodle/file.php/2/T2-III.pdf+tipos+de+heuristica&hl=es&gl=co> (último acceso: 21 de 10 de 2008).
- [15]. T'kindt, Vincent, y Jean-Charles Billaut. *Multicriteria Scheduling Theory, Models and Algorithms*. Segunda edición. Berlin: Springer, 2006.
- [16]. Universidad de Alicante. Departamento de ciencias de la computación e inteligencia artificial. *Aplicaciones: Scheduling*. 2004. <http://www.dccia.ua.es/dccia/inf/assignaturas/MTPSR/sesion9-alu.pdf> (último acceso: 7 de 2 de 2009).
- [17]. Wang, Xianpeng, y Tang Lixin. «A tabu search heuristic for the hybrid flowshop scheduling with finite intermediate buffers.» *Computers & Operations Research* (Elsevier) 36 (2007): 907-918.
- [18]. Winston, Henry Patrick. *Inteligencia Artificial, Tercera edición*. Wilmington, Delaware: Addison-Wesley, 1994.
- [19]. Zubair, M. Mohamed. «"A flexible approach to (re)configure Flexible Manufacturing Cell".» *European Journal of Operational Research*, 1995: 566.