

Un nuevo Algoritmo Genético Basado en un Sistema Multiagente para la Programación de la Producción en un Taller de Flujo Híbrido

Pedro Gómez

Centro de Investigación de Gestión e Ingeniería de la Producción (CIGIP). Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Valencia, Campus de Vera, 46022 Valencia.
pgomez@cigip.upv.es

Resumen

En este trabajo se presenta un marco general para el diseño e implementación de un algoritmo genético mediante la utilización de un Sistema Multiagente. El objetivo de dicho algoritmo es abordar el problema de la programación de la producción en un taller de flujo híbrido con y sin tiempos de cambio de partida dependientes de la secuencia para minimizar el C_{max} . El procedimiento propuesto tiene una aplicación multisectorial pero se ha realizado desde la perspectiva de la problemática de la industria cerámica. La propuesta se centra exclusivamente en el proceso de transformación de una implementación tradicional del algoritmo genético a una implementación basada en agentes.

Palabras clave: Programación de la Producción, Algoritmo Genético, Taller de flujo híbrido

1. Introducción

Este trabajo presenta una propuesta para la implementación de un nuevo algoritmo genético (AG) basado en un sistema multiagente (SMA) para la programación de la producción en un taller de flujo híbrido con e etapas. El objetivo es generar un programa para un conjunto de n trabajos con el mínimo makespan posible.

Concretamente se desarrollará un marco de trabajo para la implementación de un AG utilizando un SMA, dando a conocer una arquitectura para el mismo y un conjunto de directivas que constituyen el marco de trabajo. Queda fuera del dominio del trabajo la parametrización tradicional del AG en aquellos aspectos que no estén relacionados con la propuesta.

En esta comunicación se realizará en primer lugar el planteamiento del problema que se pretende abordar, posteriormente se analizarán las referencias más significativas relacionadas con la programación de la producción basadas en Sistemas Multiagente (SMA), a continuación se definirá la propuesta de un nuevo algoritmo genético basado en SMA, y finalmente se establecerán las conclusiones y trabajos futuros.

2. El problema del Taller de Flujo Híbrido (PTFH)

En el taller de flujo híbrido para n trabajos y e etapas, se supone un conjunto de máquinas alternativas en cada etapa que no tienen porque ser idénticas, denominadas máquinas no relacionadas en la literatura, Mokotoff (2001). Los n trabajos llegan simultáneamente a la cola de entrada al taller, es decir, a la primera etapa, para ser procesados secuencialmente en cada una de las etapas, donde se realiza una operación. Los trabajos son procesados sin interrupción y se asume que no hay problemas de almacenamiento. Los tiempos de procesamiento en cada máquina de cada trabajo son conocidos y deterministas. Las máquinas sólo pueden procesar un trabajo de forma simultánea. El objetivo es asignar los trabajos a una máquina de cada etapa y determinar la fecha de inicio y fin de proceso de cada operación de forma que el tiempo de finalización máximo o makespan sea minimizado.

2.1. Identificación y Aplicación del PTFH en la Empresa Cerámica

El autor de este trabajo ha realizado diversos estudios sobre el subsistema productivo de una industria cerámica. Se puede entender que dicho subsistema se puede modelar como un TFH con tres etapas. Estas etapas son prensas y líneas de esmaltado, horno túnel, y clasificado y embalaje.

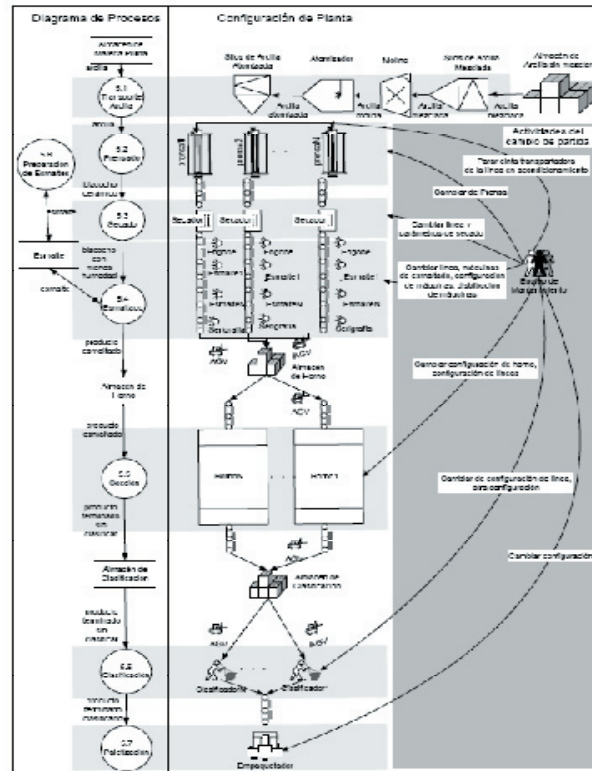


Figura 1. Taller de empresa cerámica (Giret, 2005)

El conjunto de prensas alimentan al conjunto de hornos y estos a su vez al conjunto de clasificadoras embaladoras. Se supone que la capacidad de almacenamiento intermedio y final es suficientemente grande como para no tener que preocuparse de las mismas.

2.2. Análisis del Estado del Arte del PTHF.

El taller de flujo híbrido es cada vez más estudiado existiendo diversas referencias interesantes en la literatura, Reisman et al. (1997). En este trabajo nos centraremos exclusivamente en las referencias que trabajan los tiempos de cambio de partida (TCP) que dependen de la secuencia es decir, en los que los tiempos (o costes) de preparación dependen tanto del trabajo que finaliza en una máquina como del siguiente en ser procesado. Podemos encontrar ejemplos de taller de flujo (no híbrido) con tiempos de cambio de partida dependientes de la secuencia en Corwin et al. (1974) utilizan un Branch and Bound, Gupta et al. (1985) proponen heurísticas para la resolución del problema, Gupta et al. (1986) demuestran que los programas de permutación no son dominantes en estos casos, Gupta (1986) demuestra que el problema es NP-duro, Srikar et al. (1986) propone un modelo PLM, Kochlar et al. (1987) o Szwarc et al. (1987) propone tres heurísticas basadas en el algoritmo de Johnson, Parthasarathy et al. (1997) utiliza un algoritmo basado en recocido simulado que tiene en consideración las restricciones temporales de los trabajos. También existe literatura sobre talleres de flujo híbrido con TCP no dependiente de la secuencia como en Gupta (1988) donde se desarrolla un algoritmo heurístico para un problema con dos etapas incluyendo tiempos de cambio, Gupta et al. (1993) desarrollan una heurística

para casos especiales del problema de dos etapas.

Una complejidad adicional a la hora de abordar el problema representa la consideración de los tiempos de partida dependientes de la secuencia. En este caso las referencias son muy escasas, véase Artigues (1997), Allahverdi et al. (1999), Yang et al. (1999) para un planteamiento general del problema de los tiempos de ajuste, Aggehezal et al. (1995) plantean un algoritmo heurístico en dos fases para resolver el problema de una empresa textil, Vicens et al. (1998) para un aproximación al problema usando simulación, Andrés et al. (1998-a) para un planteamiento de resolución basado en tecnología de grupos, Andrés et al. (1998-b) para la resolución de un taller de dos etapas con un algoritmo genético o Andrés et al. (2001) donde se presenta un algoritmo de recocido simulado.

En Gupta (1986) se ve como el problema del taller de flujo (no híbrido) estático con TCP dependientes de la secuencia es NP-completo y en Brah (1992) se ve como el taller de flujo híbrido o multiprocesador sin cambio de partida es NP-completo, y las soluciones sólo están disponibles para pequeños problemas y casos especiales. La dificultad para resolver este tipo de problemas ha hecho que científicos y usuarios hayan desarrollado y usado métodos heurísticos, que aunque proporcionan resultados sub-óptimos son efectivos y de simple aplicación, incluso con problemas de programación de gran tamaño.

Finalmente, cabe destacar el trabajo de Gupta y Stafford (2006), en el que se realiza un análisis pormenorizado de la investigación realizada en relación al problema del taller de flujo. En él se muestran también las líneas de futuro en el ámbito teórico, computacional y empírico.

2.3. Análisis del Estado del Arte de SMA aplicados a la secuenciación.

Es interesante prestar atención a nuevas formas de abordar el problema de la programación de la producción, y en nuestro caso nos centraremos, en los autores y referencias que utilizan los SMA.

Babayán y He (2000) presentan una metodología general de los sistemas de fabricación basados en agentes mediante teoría de juegos que incorpora la cooperación de los agentes para solucionar el problema de programación de un taller flowshop flexible de n-trabajos y 3-etapas.

Aydin y Fogarty(2004) proponen los equipos "ATeams" de agentes autónomos que cooperan compartiendo soluciones con una memoria común. Son propuestos como medios para solucionar problemas de optimización. Se prueba una arquitectura de ATeam en el problema de programación del taller. Los resultados demuestran que el método puede funcionar, pero que depende de la lista de agentes y de la manera de la cual se maneja la memoria.

Chen y Wang (2007) tiene como objetivo sugerir un marco de colaboración de un sistema basado en agentes inteligentes que pueda manejar una solución de problemas relacionados con proyectos en situaciones reales. El procedimiento de toma de decisión de dos etapas se aplica a programación dinámica en sistemas con toma de decisión bajo incertidumbre y la negociación, también se utiliza para realizar la toma de decisiones y la funcionalidad de colaboración entre agentes.

Yu y Ram (2006) proponen un sistema de secuenciación mediante un sistema multi-agente para talleres de tipo job shop dinámicos. Se presenta una estrategia basada en la división de tareas simulando el comportamiento de algunos insectos.

Wong et al (2006) establecen dos sistemas de secuenciación por agentes. Uno simplemente se basa en la negociación entre agentes-máquina y agentes-trabajo. El segundo incorpora un agente supervisor.

Shin y Jung (2005) proponen un sistema de secuenciación basado en la generación de ofertas entre agentes y su posterior evaluación. Un Mobile agent-based negotiation process (MANPro) se ocupa del mecanismo de negociación. Este artículo se centra en un segundo aspecto, que es como son generadas y evaluadas las ofertas y como los agentes llegan a acuerdos en el MANPro. En primer lugar se propone la arquitectura del sistema y posteriormente se aplica a un caso de secuenciación para un sistema de control de shop floor (SFCS).

Aydin y Oztemel (2000) presentan un sistema de secuenciación dinámico basado en agentes inteligentes. Consiste en dos componentes independientes: el agente y el ambiente simulado (SE). El agente selecciona la regla más apropiada para dar prioridad de acuerdo a las condiciones del taller en tiempo real, mientras que el ambiente simulado realiza las actividades programadas usando la regla seleccionada por el agente.

Usher (2003) estudia un método para realizar el proceso de negociación empleado por un sistema de agentes para mejorar el funcionamiento de un taller. La propuesta aprovecha un período extendido de negociaciones para proporcionar un cuadro más completo de las condiciones del taller para validar las decisiones tomadas por los agentes individuales. El funcionamiento que resulta se prueba en una simulación de un taller Job Shop.

Araúzo et al (2006) expone los fundamentos de estas técnicas relacionadas con las subasta en mercado y su relación con métodos de descomposición de programación matemática, aplicados a la programación de talleres generales.

Fuera de las referencias señaladas no se han encontrado trabajos destacados que combinen problemas de secuenciación y SMA.

3. Metodología Propuesta

Basándose en la propuesta de Holland (1975), para la construcción de algoritmos genéticos en este trabajo se presenta un marco para su diseño e implementación mediante un sistema multiagente (Shen y Norrie (1999)). Respetando las ideas básicas establecidas para el diseño de AG, se trata de repartir las diferentes actividades que se realizan en un AG en roles que serán asignados a diferentes agentes. Este trabajo trata de implementar un AG tradicional incorporando nuevas tecnologías que permitan, mediante la aplicación de nuevos enfoques, la extensión y mejora de dichos algoritmos metaheurísticos.

El objetivo del algoritmo es la generación de programas que minimicen el Cmax o Makespan de un conjunto de trabajos dado.

La representación genética de las soluciones del problema se realizará mediante una representación ordinal, en la que cada individuo está representado por una secuencia de trabajos.

Se deja abierta la posibilidad de que se puedan ejecutar varios algoritmos en paralelo cada uno de ellos con diferentes configuraciones. De esta forma, al mismo tiempo que se podría aprovechar una de las cualidades de los SMA, el sistema adquiere una mayor robustez frente a cambios en la configuración de los datos de entrada.

3.1. La Arquitectura del SMA

En este caso la arquitectura multiagente asume que existen 2 tipos de agentes asociados al algoritmo, denominados *Gestor* y *Familia*, y uno adicional asociado a la coordinación del conjunto de algoritmos que pudieran trabajar en paralelo, llamado *Creador*. Entre todo ellos existe una relación de jerarquía, Ilustración 2. El objetivo de cada agente se describe a continuación.

- *Agente Creador*: Ocupa la posición más alta en la jerarquía. Sus tareas son dos: Servir de interfase con el usuario para leer los datos del problema y generar el número adecuado, previamente establecido, de procedimientos de resolución, seleccionando el mejor resultado.
- *Agente Gestor*: Depende jerárquicamente del Agente Creador, que es quien lo crea. Forma parte del algoritmo, ya que es básicamente el encargado de controlar la evolución del mismo, marcando el inicio y fin de una generación.
- *Agente Familia*: Trabaja en el nivel jerárquico más bajo. Forma parte del algoritmo, y trabaja al ritmo que le marca el Agente Gestor. Su misión es representar un conjunto de soluciones, e intentar modificar la configuración de sus representados para mejorar el Cmax medio del conjunto.

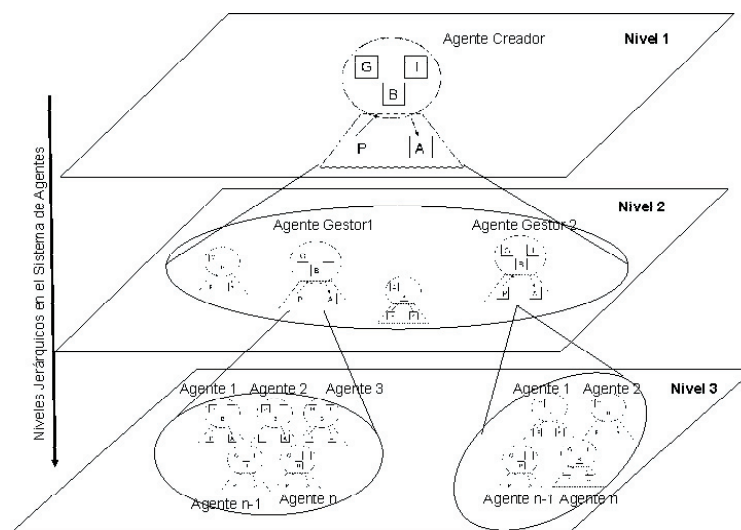


Figura 2. Representación de la jerarquía de Agentes en el SMA (elaboración propia).

3.2. Esquema general del algoritmo.

Como se ha indicado en la introducción, esta propuesta desarrolla un marco de trabajo general. Aspectos como por ejemplo el tipo de cruce, mutación, o n° de iteraciones que se deben realizar dependerán de cada caso, y están relacionadas más con el AG que con su proceso de implementación con un SMA.

A continuación se presenta el esquema general del algoritmo de forma esquemática para posteriormente detallar los aspectos relevantes:

1. El Gestor genera un conjunto de agentes Familia cuyo número depende de la parametrización realizada, y queda a la espera de que estas estén disponibles para iniciar sus actividades.

2. En el momento de su creación, cada Agente Familia genera un conjunto de soluciones que se agrupan, y que constituyen su familia. El objetivo de este agente es aumentar la calidad media de sus soluciones reduciendo el Cmax medio. Inmediatamente después, el Agente familia establece dos hilos de ejecución con el objetivo de mantener en paralelo 2 roles: Subastador y Postor, ver Ilustración 3. Y finalmente, comunica al Agente Gestor su disponibilidad y queda a la espera.
3. Una vez el Agente Gestor sabe que todas las familias están disponibles inicia el proceso, es decir el núcleo del algoritmo genético. El primer paso es seleccionar aleatoriamente a una familia para realizar un cruce y generar nuevos individuos. En este proceso se aplica toda la potencia de los sistemas multiagente.
4. El individuo seleccionado aleatoriamente toma el papel de subastador y comunica al resto de familias su deseo de realizar un cruce con el mejor postor.
5. El resto familias, a través del agente que le representa, pujan por participar en un cruce con un individuo previamente seleccionado.
6. El agente ganador de la puja se hace cargo de los hijos generados. Un hijo se incorporará a una familia si su makespan es menor que el de alguno de los miembros actuales, en caso contrario no se tendrá en cuenta. El agente postor ganador comunica al agente Gestor el final del proceso de cruce.
7. Una vez finalizado el cruce se realiza un proceso de mutación.
8. Finalmente, y antes de volver al paso 3, el Gestor realiza una regeneración de la población si se considera necesario.

Como se puede apreciar, a diferencia de los algoritmos genéticos tradicionales, en esta propuesta cada participante en un proceso de selección para la realización de un cruce posterior, no es una solución sino un conjunto de ellas denominada en este caso familia. Es sumamente interesante destacar que el sistema de familias proporciona a los agentes el objetivo de mejorar los individuos/soluciones que la componen mediante el cruce y sustitución por otros individuos mejores. Como se verá al analizar el proceso de subasta la mejora del makespan medio de una familia dotará a esta de más recursos, y así el agente podrá disponer cada vez de más competitividad, que podrá utilizar en el momento de la puja. Así mismo, se ha proporcionado a los agentes de autonomía necesaria creando funciones de utilidad basadas en la teoría de juegos que evolucionan en función de la participación del agente y del entorno. Así un agente puede tener más o menos aversión al riesgo a lo largo del proceso.

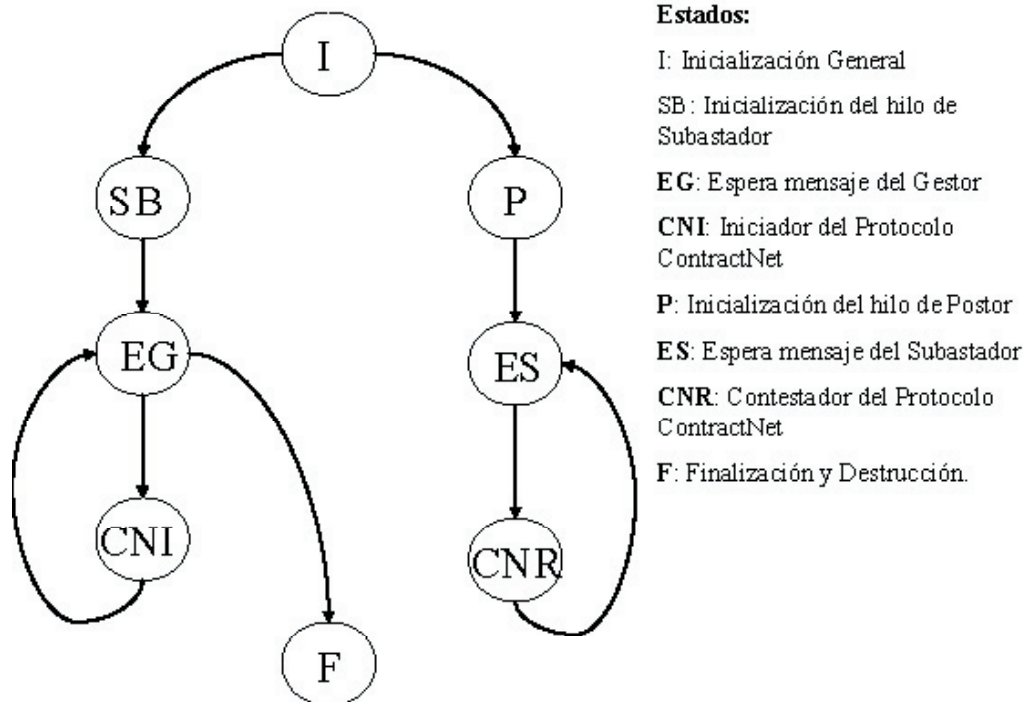


Figura 3. Estados de un Agente Familia (elaboración propia).

3.1. El procedimiento de subasta

Es interesante profundizar un poco más en un aspecto clave y novedoso en este AG. Como es el procedimiento de subasta.

Antes del inicio de la subasta el Gestor a partir de un número constante de puntos asignará a cada agente los que le pertenece. El reparto se hace en función del Cmax medio de cada familia. De esta forma las familias con peores soluciones medias disponen de menos puntos y las de mejores soluciones más puntos. Al disponer de un número constante de puntos y repartirse en cada iteración las familias que hayan mejorado su Cmax medio tienen la oportunidad de modificar su estado en relación al resto. Así mismo, se puede observar como una familia que no mejore va a ir poco a poco perdiendo puntos, ya que estos serán entregados a las familias que si mejoren.

En el procedimiento de subasta el Subastador envía el material genético que identifica a la mejor solución propia al resto de agentes. El resto de agentes procederán a realizar el procedimiento de cruce como si hubiesen sido ya elegidos, obteniendo 2 nuevos individuos (hijos). De esta forma todos los agentes pueden saber cual sería el resultado si ellos fuesen los elegidos y pueden calcular la mejora expresada como un incremento o decremento porcentual de su Cmax medio.

El valor o puja que cada postor envía al subastador es función de la combinación de los puntos o recursos de las familias y de la mejora del Cmax medio si los nuevos hijos se incorporasen. En caso de que la nueva generación no supusiese una mejora la puja sería cero. De igual forma si alguna solución de la nueva generación mejora el mejor Cmax de la población la puja será infinito. Se puede decir que las familias ponen en juego parte de su patrimonio (puntos adquiridos) en función de lo importante que sea su aportación a la población en forma de mejora del Cmax.

$$Puja = Puntos * \frac{(\overline{Cmax_{n-1}} - \overline{Cmax_n})}{Cmax_{n-1}} \quad (1)$$

El objetivo final de un AG es el de obtener el menor Cmax posible para un conjunto de trabajos dado. En general los autores suelen considerar que esta meta se puede alcanzar mediante la combinación de los mejores individuos de forma preferente. Se suele tener en cuenta que si sólo participan algunos individuos, los mejores, en el proceso de cruce se podría converger a un óptimo local, por lo que es necesario intentar obtener nuevas buenas soluciones utilizando de forma esporádica las soluciones no tan buenas. En línea con estas ideas sobre la combinación de individuos en el proceso de cruce se propone utilizar técnicas relacionadas con la teoría de la decisión/juegos, y en concreto con el concepto de utilidad.

Se puede intuir que las familias que más probabilidades tienen de ganar una subasta son las buenas familias (Cmax medio pequeño), o las que mejoran mucho su Cmax medio después de una nueva generación. Pero las familias con muy pocos puntos, ¿serán capaces de aumentar lo suficiente para poder ganar alguna puja? Para intentar potenciar el valor de estas últimas se propone que el valor del incremento o decremento del Cmax medio sea tratado como entrada a una función de utilidad. Las familias con mejor Cmax medio dispondrán de un perfil con aversión al riesgo, mientras que las que disponen de peor Cmax medio tendrán un perfil de jugador arriesgado.

$$Puja = Puntos * U\left(\frac{(\overline{Cmax_{n-1}} - \overline{Cmax_n})}{Cmax_{n-1}}\right) \quad (2)$$

El objetivo es que no sólo se produzcan nuevos individuos en un subconjunto reducido de buenas soluciones, lo que sólo ocasionaría una mejora en ese subconjunto, pero que quizás sólo implique alcanzar un óptimo local. Se ha considerado que quizás sea más interesante que sacrifique dicha mejora para dar cabida a un cambio entre las soluciones menos buenas aumentando el espectro de participantes y reduciendo la posibilidad de caer en óptimos locales.

3.2. El procedimiento de regeneración

Por su puesto, el proceso de subasta que afecta básicamente a la función de selección del AG tradicional debe combinarse con el de regeneración de la población. En este sentido se proponen dos acciones.

- Regeneración en las familias: Supone la modificación de la estructura de las familias mediante el cambio o mutación, según el caso, de alguna solución.
- Regeneración en la población: Implicaría la sustitución de una familia entera que debe ser sustituida por otra nueva.

Se propone que la regeneración del primer tipo se realice de forma periódica y conjuntamente con el proceso de subasta. Se trataría de que cada familia realice un seguimiento de la aportación de los diversos individuos a la subasta, y cada cierto nº de iteraciones realizar una mutación. En el caso de que la situación de la familia sea crítica (Cmax medio muy elevado) se podría intentar la generación de nuevos individuos.

El segundo tipo de regeneración estaría asociada a la evolución del mejor Cmax de la población. Se propone que si éste permanece estancado un número de iteraciones elevado se sustituya parte

de la población por otra nueva. En este caso las familias peores formarían parte importante de las familias a sustituir, aunque no en exclusiva.

4. Conclusiones y trabajo futuro

En el futuro se ha previsto una comparativa entre el algoritmo propuesto y otras propuestas relevantes para el problema indicado con el objeto de establecer la calidad del procedimiento.

Referencias

Aghezzaf E. A., Artiba A., y Elmaghraby S. E., Hybrid FlowShop: an LP based heuristic for planning level problems, ETFA Proceedings, 551-559, 1995.

Allahverdi A., Gupta J., Aldowaisan F., A review of scheduling research involving setup considerations, Omega, 27, 219-239, 1999.

Andrés C., Albarracin J. M., Vicens E., Tormo G., Group Technology in a two-stage flowshop environment: a case study., 16th European Conference on Operation Research, Bélgica, 1998.

Andrés C, Abad, Ros L. y Vicens E., A Genetic Algorithm for Production Scheduling in a two stage Irbid flowshop with sequence dependent setup times, 16th European Conference on Operation Research, Bélgica, 1998-b.

Andrés C., Programación de la producción en Talleres de Flujo Híbrido con Tiempos de Cambio de Partida dependientes de la Secuencia. Métodos y Algoritmos de Resolución. Aplicación a Empresas del Sector Cerámico., Tesis Doctoral, Universidad Politécnica de Valencia, 2001.

Andrés C, Vicens E. y Lario F. C., Algoritmo de recocido simulado para la secuenciación en talleres de flujo híbridos con tiempos de cambio de partida dependientes de la secuencia, XXVI Congreso nacional de estadística e investigación operativa, Úbeda (Jaén), 2001.

Araújo A., Benito de J. J., Sanz P., Olmo de, R., Diseño de subastas para la programación del Job Shop, Actas X Congreso de Ingeniería de Organización, pp.159-160, 2006.

Artigues C., Ordennancement en temps reel d'ateliers avec temps de preparation des ressources, Thèse LAAS, Université Paul Sabatier de Toulouse, 1997.

Aydin, M. E. and T. C. Fogarty. Teams of autonomous agents for job-shop scheduling problems: An experimental study. Journal of Intelligent Manufacturing 15.4 (2004): 455-62.

Aydin, M. E. and E. Oztemel. Dynamic job-shop scheduling using reinforcement learning agents. Robotics and Autonomous Systems 33.2-3 (2000): 169-78.

Babayan, A. and D. He. Solving the n-job 3-stage flexible flowshop scheduling problem using an agent-based approach. International Journal of Production Research 42.4 (2004): 777-99.

Brah S. A., A comparative analysis of due date based job sequencing rules in a flow shop with multiple processors, Production Planning & Control, Vol. 7, n° 4, 362-373, 1996.

Chen, Y. M. and S. C. Wang. Framework of agent-based intelligence system with two-stage decision-making process for distributed dynamic scheduling. Applied Soft Computing 7.1 (2007): 229-45.

Corwin B. D. y Esogbue A. O., Two-machines Flowshop Scheduling problems with Sequence

- Dependent Setup Times: A dynamic programming approach, *Naval Research Logistics Quarterly*, 21, 515-524, 1974.
- Gupta J. N. D. y Darrow W. P., Approximate schedules for the two machine flowshop with sequence dependent setup times, *Indian Journal of Management and Systems*, 1, 6-11, 1985.
- Gupta J. N. D. y Darrow W. P., The two machine sequence dependent flowshop scheduling problem, *European Journal of Operational Research*, 24, 439-446, 1996.
- Gupta J. N. D., Flowshop schedules with sequence dependent setup times, *Journal of Operational Research Society*, 29, 206-219, 1986.
- Gupta J. N. D., Two stage hybrid flowshop scheduling problem., *Journal of Operational Research Society*, 34, 359-364, 1988.
- Gupta J. N. D. y Tunc E. A., Schedules for a two-stage hybrid flowshop with separable setup and removal times. *European Journal of Operational Research*, 1993.
- Gupta J. N. D. y Stafford E. F., Flowshop scheduling research after five decades, *European Journal of Operational Research* 169, pg. 699-711, 2006.
- Holland, J. H., *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.
- Kochlar S. y Morris R. J. T., : Heuristics methods for flexible flow line scheduling., *Journal of Manufacturing Systems*, 6, 299-314, 1987.
- Mokotoff, E. Parallel machine scheduling problems: A survey. *Asia-Pacific Journal of Operational Research*, 18(2): 193-242, 2001.
- Parthasaraty S. y Rajendran C., A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence dependent setup times of jobs: a case study, *Production Planning and Control*, 8, 475-483, 1997.
- Reisman A., Kumar A. y Motwani J., Flowshop scheduling/sequencing research: A statistical review of literatura, 1952-1994, *IEEE transactions on engineering management*, vol. 44, n° 3, 1997.
- Shen W.; Norrie D. H. (1999), *Agent-Based Systems for Intelligent Manufacturing: A State of the Art Survey*, *International Journal of Knowledge and Information Systems*, 1 (2), pp. 129-156. Disponible <http://imsg.enme.ucalgary.ca/publication/abm.htm>.
- Shin, M. and M. Jung. Bid generation and evaluation for MANPro-based real time scheduling. *International Journal of Production Research* 43.18 (2005): 3821-36.
- Usher, J. M. Negotiation-based routing in job shops via collaborative agents. *Journal of Intelligent Manufacturing* 14.5 (2003): 485-99.
- Srikar B. N. y Ghosh S., A MILP Model for the n-job, M stage flowshop with sequence dependent setup times, *International Journal of Production Research*, 24, 1459-1474, 1986.
- Szwarc W. y Gupta J. N. D., A flow shop problems with sequence dependent additive setup times, *Naval Research Quarterly*, 23, 619-627, 1987.

Vicens E., Martinez V., Albaracín J. M., Alfaro J.J., An industrial application for finite capacity production scheduling in a tile industry., 16th European Conference on Operation Research, 1998.

Wong, T. N., et al. Dynamic shopfloor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications* 31.3 (2006): 486-94.

Yang W. y Liao C., Survey of scheduling research involving setup times, *Internacional Journal o Systems Science*, 30, 2, 143-155, 1999.

Yu, X. F. and B. Ram. Bio-inspired scheduling for dynamic job shops with flexible routing and sequence-dependent setups. *International Journal of Production Research* 44.22 (2006): 4793-813.