

Tipo de artículo: Artículo original
Temática: Inteligencia Artificial
Recibido: 01/09/2020 | Aceptado: 20/11/2020

Aplicación de sistemas neuroborrosos en la clasificación de reportes en problemas de secuenciación

Application of neuro-fuzzy systems in the classification of reports in scheduling problems

Jessica Coto Palacio ^{1,2} <https://orcid.org/0000-0002-1977-145X>

Yailen Martínez Jiménez ^{2,3*} <https://orcid.org/0002-1223-0589>

Ann Nowé ³ <https://orcid.org/0000-0001-6346-4564>

¹ UEB Hotel Los Caneyes. jcotopalacio@gmail.com

² Universidad Central “Marta Abreu” de Las Villas. yailenm@uclv.edu.cu, jcoto@uclv.cu

³ Vrije Universiteit Brussel. ann.nowe@ai.vub.ac.be, ymartine@vub.ac.be

*Autor para la correspondencia. (yailenm@uclv.edu.cu)

RESUMEN

La secuenciación de tareas es un área muy amplia en la cual muchos investigadores se han enfocado en los últimos años. En las empresas generalmente esta planificación se realiza de forma manual o semiautomática. En este trabajo se propone la aplicación de sistemas neuroborrosos en la clasificación de reportes en problemas de secuenciación, paso necesario para identificar en que recurso se procesará el

reporte y posteriormente conformar la secuencia o planificación de trabajo del día. Para la clasificación de los reportes que arriban al sistema se utilizan cuatro algoritmos neuroborrosos. Los experimentos muestran que el algoritmo que mejores resultados obtiene es IVTURS, y las reglas difusas obtenidas por el mismo son analizadas para arribar a conclusiones sobre la distribución de reportes entre los recursos.

Palabras clave: clasificación de reportes; secuenciación de tareas; sistemas neuroborrosos.

ABSTRACT

Scheduling is a very broad area in which many researchers have focused in the past years. In most of the companies this process is usually done manually or semi automatically. In this work we propose the application of neuro-fuzzy systems in the classification of reports in scheduling problems, a necessary step to identify in which resource the report will be processed, in order to build the work sequence or schedule for the day. For the classification of the reports arriving to the system four neuro-fuzzy algorithms are used. The experiments show that the algorithm that obtains the best results is IVTURS, and the fuzzy rules obtained are analyzed to arrive to conclusions regarding the distributions of reports among the resources.

Keywords: report classification; scheduling problems; neuro-fuzzy systems.

Introducción

Los problemas de secuenciación están presentes en todas aquellas situaciones donde es necesario ejecutar un conjunto de tareas y para esto se requiere la asignación de las mismas a los recursos disponibles en intervalos de tiempo. En (Zhang, 1996) se define la secuenciación de tareas como el proceso de seleccionar planes alternativos y asignar recursos y tiempos a un conjunto de actividades en un plan. Debido a esto, las asignaciones deben obedecer ciertas restricciones que reflejan las relaciones temporales entre las actividades y las limitaciones de los recursos compartidos. La mayoría de las investigaciones en el área de la

secuenciación de tareas se han enfocado en el desarrollo de procedimientos exactos para la generación de una solución base asumiendo que se tiene toda la información necesaria y un ambiente determinístico (Herroelen, y otros, 2005). Este enfoque es conocido en la literatura como problemas de secuenciación de tareas en ambientes de tipo offline.

En ambientes online, por otra parte, una secuencia de trabajos $\sigma = \{J_1, J_2, \dots, J_n\}$ tiene que procesarse en un número determinado de máquinas. Los trabajos arriban al sistema uno a uno, y cuando un nuevo trabajo llega tiene que ser inmediatamente despachado hacia una de las máquinas, sin tener conocimiento sobre trabajos futuros. El objetivo es optimizar una función objetivo determinada. Los problemas de secuenciación pueden clasificarse teniendo en cuenta el ambiente de las máquinas, las características de los trabajos y la función objetivo. Esta clasificación se conoce comúnmente como $\alpha|\beta|\gamma$ (Graham, y otros, 1979), donde α representa el ambiente de las máquinas, β las características de los trabajos y γ el criterio a optimizar.

Según el ambiente de las máquinas el escenario más simple es en el que se cuenta con un solo recurso, pues los trabajos tienen una sola operación a ser procesada y solo existe una máquina que pueda ejecutarla. Cuando existen múltiples máquinas el entorno se torna más complicado, ya que pueden ser idénticas o pueden diferir en velocidad. Los posibles ambientes con máquinas paralelas se resumen de la siguiente forma (Martínez-Jiménez, 2012):

1. Máquinas Paralelas Idénticas: Procesan los trabajos a la misma velocidad.
2. Máquinas Paralelas Diferentes: El tiempo de procesamiento depende de la máquina.
3. Máquinas Paralelas no relacionadas: El tiempo de procesamiento depende de la máquina y del trabajo.

El presente trabajo se centra en modelar un problema de clasificación que se corresponde con un ambiente de máquinas paralelas no relacionadas, donde se cuenta con diferentes máquinas para procesar los reportes, cada una tiene características diferentes, y cada reporte puede ser procesado solo por una máquina en específico. Hay dos formas posibles de solucionar este tipo de problemas, utilizando enfoques jerárquicos o integrados. El primero divide el problema en enrutamiento (asignando una máquina a cada reporte) y

secuenciación, mientras que el segundo considera ambos pasos al mismo tiempo. En nuestro caso se siguen las ideas de los enfoques jerárquicos y la propuesta de solución se basa en la utilización de varios enfoques de sistemas neuroborrosos existentes en la literatura para asignar una máquina a cada reporte y dar esa asignación como entrada a un algoritmo de secuenciación.

El resto del artículo está estructurado de la siguiente manera: La próxima sección presenta una descripción general de los problemas de secuenciación en ambientes con máquinas paralelas no relacionadas y posibles métodos de solución, y hace una descripción general del problema a resolver. Posteriormente se describen los algoritmos neuroborrosos aplicados a la problemática existente, y por último se muestra la configuración de los experimentos, así como los resultados obtenidos. El trabajo termina con conclusiones sobre los resultados obtenidos por los algoritmos neuroborrosos al aplicarse a los conjuntos de datos de clasificación de reportes.

Clasificación de reportes en problemas de secuenciación

En los problemas de secuenciación de tareas de la vida cotidiana se procesan trabajos siguiendo un determinado orden, de forma tal que se logre optimizar los tiempos en los que transitan por el sistema. La ejecución de los reportes generados por los clientes de los supermercados a través de compras por diferentes vías, es un problema de secuenciación de tareas. Muchos de los reportes deben cumplir una serie de condiciones y el sistema debe ser capaz de priorizar su ejecución en caso de ser necesario.

Existe una cantidad de m tipos de máquinas, donde cada tipo de máquina cuenta con un número determinado de recursos. También se tienen n reportes en lotes, agrupados por el tipo de reporte, los cuales deben ser procesados por un tipo de máquina. Estos reportes cuentan además con un idioma, un tiempo medio de procesamiento, una cantidad determinada de pedidos y un tiempo total de procesamiento que está dado por la cantidad de pedidos y la máquina especializada en este. Los reportes son independientes, por tanto, pueden ser procesados simultáneamente siempre que haya capacidad disponible. Si las máquinas

están ocupadas, los reportes deben esperar que exista disponibilidad. Entre los recursos no existe ningún tipo de relación.

La Tabla 1 muestra un pequeño ejemplo donde se puede apreciar el formato de los datos que necesitan ser procesados. En el sistema existen 9 tipos de máquinas que procesan reportes con características específicas, pero además se cuenta con una décima que es capaz de asumir cualquier tipo de reportes, es decir, a la máquina número 10 se le puede asignar reportes que no necesiten tratamiento especial alguno y reportes de las otras 9 en caso de que estas sufran algún tipo de avería.

Tabla 1 - Ejemplo de datos.

Idioma	Duración Total (seg)	Número de ejecuciones	Promedio de tiempo de ejecución (seg)	Máquina
Inglés	229455	103	2227,715621	M7
Español	90256	51	1769,719275	M10
Español	15905	9	1767,186111	M10
Francés	233587	136	1717,549338	M7
Español	24852	17	1461,874765	M10
Inglés	151902	111	1368,487595	M7
Alemán	134127	102	1314,971363	M7
Español	451647	351	1286,744875	M10
Francés	125216	102	1227,607529	M7
Español	12962	11	1178,324091	M10
Español	15069	13	1159,134077	M10
Inglés	81479	73	1116,143932	M10
Alemán	126054	114	1105,740026	M7
Español	10986	10	1098,599200	M10
Español	81323	79	1029,408329	M10

En este trabajo se cuenta con tres bases de casos con 150 lotes de reportes cada una, y cada base cuenta con la información en el formato mostrado en la tabla anterior. En el primer juego de datos las 10 máquinas están trabajando, por lo que cada reporte es asignado a su correspondiente recurso, y en los otros dos existen máquinas sin funcionar, uno con nueve recursos funcionando y el otro con solo seis. Como se mencionó

anteriormente, al resolver un problema de este tipo se pueden usar dos alternativas, dividir el problema en ruteo y secuenciación, o resolver ambas cosas a la vez. En este caso para construir una solución al problema de secuenciación se ha optado por realizar inicialmente el paso de ruteo, por tanto, para cada reporte se debe decidir qué recurso lo procesará, y para eso se aplican los algoritmos neuroborrosos que se detallan en la próxima sección.

Sistemas Neuroborrosos

La creciente necesidad de desarrollar sistemas inteligentes adaptativos para solucionar problemas del mundo real ha hecho que la fusión entre las redes neuronales artificiales y los sistemas de inferencia borrosos haya atraído un creciente interés por parte de investigadores de diversas áreas científicas (Gajate-Martín, 2011). Los sistemas neuroborrosos surgen a principios de los noventa combinando las ventajas de las redes neuronales artificiales y de los sistemas de inferencia borrosos (Jang, 1993).

Es necesario aclarar que no basta que una red neuronal y un sistema de inferencia borroso sean usados juntos para estar en presencia de un modelo neuroborroso. Un sistema neuroborroso es una forma de crear un sistema de inferencia borroso utilizando algún tipo de heurística o método de aprendizaje inspirado en los mecanismos de aprendizaje usados en las redes neuronales (Nauck, 2000). Una de las ventajas fundamentales es que las redes neuronales aprenden “desde cero” a través del ajuste de las interconexiones entre sus distintas capas. Sin embargo, uno de los mayores problemas de estas técnicas es que el conocimiento aparece de forma implícita en forma de “caja negra” y suelen fallar cuando se producen comportamientos fuera de la región de entrenamiento (Gajate-Martín, 2011).

Por otra parte, los sistemas de inferencia borrosos pueden utilizar el conocimiento humano mediante el almacenamiento de información en forma de bases de reglas y conjuntos de datos, realizando posteriormente razonamientos borrosos para inferir el valor de salida. Estas técnicas permiten expresar incertidumbre en los sistemas basados en reglas y poseen además robustez intrínseca. Sin embargo, la obtención de las reglas borrosas y de las funciones de pertenencia correspondientes depende en gran medida

del conocimiento que se tenga a priori del sistema objeto de estudio. Esto se plasma, entre otros factores, en la dificultad que existe para ajustar los parámetros de las funciones de pertenencia de manera óptima (Grau-García, y otros, 2012). La fortaleza principal de los métodos neuroborrosos es que son aproximadores universales que ofrecen reglas interpretables (Jin, 2000).

Además de las ventajas anteriormente detalladas, debemos mencionar que en el lugar donde se aplicarán los resultados de este trabajo existían investigaciones precedentes donde se utilizaban los sistemas de inferencia borrosos, y se solicitó que las primeras pruebas se realizaran utilizando este tipo de sistemas, de ahí que en esta investigación se decida utilizar los cuatro algoritmos que se describen a continuación:

1. FURIA (Fuzzy Unordered Rule Induction Algorithm): Este algoritmo constituye una ampliación o extensión de RIPPER (Cohen, 1995), un algoritmo de aprendizaje de reglas del estado del arte, preservando sus conocidas ventajas, como por ejemplo, los conjuntos de reglas simples y comprensibles. En particular, FURIA aprende reglas borrosas en lugar de reglas convencionales, y conjuntos de reglas no ordenadas en lugar de listas de reglas. Además, para tratar con ejemplos no cubiertos hace uso de una regla eficiente de estiramiento. Los resultados experimentales presentados en (Hühn, y otros, 2009) muestran que FURIA supera significativamente al algoritmo RIPPER original, así como a otros clasificadores como por ejemplo el C4.5, en términos de precisión de clasificación.
2. IVTURS (Sanz, y otros, 2013): Un sistema de clasificación lingüística basado en reglas difusas, el cual se basa en un nuevo método de razonamiento difuso evaluado por intervalos con ajuste y selección de reglas. Este proceso de inferencia utiliza funciones de equivalencia restringidas por intervalo para aumentar la relevancia de las reglas en las cuales la equivalencia de los grados de membresía por intervalos de los patrones y los grados ideales de membresía son mayores, lo cual es un comportamiento deseable. Adicionalmente, su construcción parametrizada permite calcular la función óptima para cada variable, lo que podría implicar una mejora potencial en el comportamiento del sistema. Además, este ajuste de la equivalencia se combina con la selección de reglas para disminuir la complejidad del sistema. IVTURS se compone de tres etapas: 1) La

generación de un sistema de clasificación inicial basada en reglas difusas (IV-FRBCS). Para hacer esto, primero se aprende la base de reglas utilizando el algoritmo de aprendizaje de reglas difusas conocido como FARC-HD (modelo de clasificación basado en reglas de asociación difusa para problemas de alta dimensión) y luego se modelan las etiquetas lingüísticas con IVFS; 2) La aplicación de un IV-FRM y 3) Un paso de optimización que utiliza la sinergia entre el ajuste de la equivalencia y la selección de reglas.

3. SLAVE (Structural Learning Algorithm in Vague Environment): es un algoritmo de aprendizaje inductivo que utiliza conceptos basados en la teoría de la lógica difusa y utiliza un enfoque iterativo para aprender con algoritmos genéticos. Este método es un enfoque alternativo de los enfoques clásicos de Pittsburgh y Michigan. El algoritmo de aprendizaje extrae un conjunto de reglas difusas de una serie de ejemplos. Este proceso se desarrolla a través un método iterativo en el que se selecciona una regla cada vez. SLAVE utiliza un algoritmo genético para seleccionar la regla que mejor representa el sistema, esta se incorpora en el conjunto final de reglas. Con el fin de obtener nuevas representaciones, la regla anteriormente obtenida se penaliza (mediante la eliminación de los ejemplos incluidos en esta regla) y se repite el proceso. Este esquema iterativo se repite hasta que el conjunto de reglas obtenidas representa adecuadamente los ejemplos en el conjunto de entrenamiento, retornando el conjunto de reglas como la solución al problema (Gonzalez, y otros, 2001). Se han propuesto modificaciones a este algoritmo, que incluyen nuevos operadores genéticos para reducir el tiempo necesario para aprender y mejorar la comprensión de las reglas obtenidas. Además, se han propuesto nuevas formas de penalizar las reglas en el enfoque iterativo que permite mejorar el comportamiento del sistema (Gonzalez, y otros, 1999).
4. NSLV(González, y otros, 2009): Es un algoritmo de aprendizaje de reglas difusas basado en el uso de una estrategia de cobertura secuencial. Extrae iterativamente una regla única que se agrega al conjunto de reglas. La selección de la mejor regla en cada iteración se lleva a cabo por medio de una Algoritmo Genético. El algoritmo devuelve la base de reglas final obtenida al terminar el proceso de aprendizaje. NSLV es una versión mejorada de SLAVE2 que evita el sesgo generado por la selección de la clase en el proceso iterativo. Definitivamente, NSLV es la evolución natural de los algoritmos SLAVE y SLAVE2 descritos en (Garcia, y otros, 2014).

Como se mencionó anteriormente, el objetivo general de estos métodos es encontrar el conjunto de reglas difusas que mejor se ajuste a los datos de entrenamiento. En este trabajo se evalúa el desempeño de los cuatro métodos neuroborrosos descritos ante el problema de clasificación de reportes en problemas de secuenciación. Una vez aplicados estos algoritmos se obtiene una asignación de reportes a recursos, y es posible entonces pasar a realizar la secuenciación que no es más que encontrar el orden adecuado en que deben ejecutarse los reportes dentro de cada recurso, con el objetivo de minimizar el tiempo de completamiento. En la próxima sección se muestra un ejemplo de reglas obtenidas a partir de uno de los conjuntos de datos con los que se realizan los experimentos.

Resultados y discusión

Para la ejecución de los algoritmos neuroborrosos descritos en el epígrafe anterior se utiliza la herramienta Keel (Alcalá-Fdez, y otros, 2011), herramienta implementada en Java, de código abierto que permite resolver problemas de regresión, clasificación, agrupamiento, entre otros. Para esto cuenta con una gran colección de algoritmos de extracción del conocimiento, técnicas de pre-procesamiento, algoritmos de aprendizaje basados en inteligencia computacional, algoritmos evolutivos de aprendizaje de reglas basados en diferentes enfoques, etc.

Como paso inicial se importaron los datos en el Keel en formato xls y se aplicaron las respectivas divisiones o particiones que requiere el programa. Se utilizó validación cruzada en diez particiones (10 fold cross-validation). Los resultados de los experimentos en cuanto a la clasificación se muestran en la siguiente tabla.

Tabla 2. Resultados obtenidos por los algoritmos para tres juegos de datos.

Instancia \ Algoritmo	FURIA	IVTURS	NSLV	SLAVE
Reportes 10 máquinas	0.945	0.971	0.957	0.944
Reportes 9 máquinas	0.923	0.952	0.932	0.922

Reportes 6 máquinas	0.890	0.907	0.879	0.889
---------------------	-------	--------------	-------	-------

Según los resultados obtenidos y como se muestra en la tabla anterior, en todos los casos el algoritmo IVTURS obtiene los mejores resultados al tener una mayor calidad en las reglas difusas encontradas. Dichas reglas pueden ser interpretadas de manera tal que se obtenga una clasificación lo más certera posible en cuanto a que tipos de reportes están destinados a que máquina, teniendo en cuenta las características de los reportes. De la ejecución de los algoritmos analizados en este artículo se obtienen dos tipos de ficheros como resultados, uno que define las particiones difusas consideradas por el algoritmo para cada variable, y otro archivo que describe el conjunto final de reglas difusas obtenidas. En la Figura 1 se muestra un ejemplo de las particiones de las variables “número de ejecuciones” y “duración total”, para el caso en que todas las máquinas están funcionando.

```
@Number of Labels in Variable 2: 5
DuracionTotal(seg):
L_0(5): (-970281.75, 489.0, 971259.75, -1455667.125, 489.0, 1456645.125)
L_1(5): (489.0, 971259.75, 1942030.5, -484896.375, 971259.75, 2427415.875)
L_2(5): (971259.75, 1942030.5, 2912801.25, 485874.375, 1942030.5, 3398186.625)
L_3(5): (1942030.5, 2912801.25, 3883572.0, 1456645.125, 2912801.25, 4368957.375)
L_4(5): (2912801.25, 3883572.0, 4854342.75, 2427415.875, 3883572.0, 5339728.125)

@Number of Labels in Variable 3: 5
numeroDeEjecuciones:
L_0(5): (-1723.25, 2.0, 1727.25, -2585.875, 2.0, 2589.875)
L_1(5): (2.0, 1727.25, 3452.5, -860.625, 1727.25, 4315.125)
L_2(5): (1727.25, 3452.5, 5177.75, 864.625, 3452.5, 6040.375)
L_3(5): (3452.5, 5177.75, 6903.0, 2589.875, 5177.75, 7765.625)
L_4(5): (5177.75, 6903.0, 8628.25, 4315.125, 6903.0, 9490.875)
```

Fig. 1 - Ejemplo de particiones creadas por el algoritmo IVTURS.

Estas particiones, teniendo en cuenta los valores de dichas variables, pueden clasificarse en “muy bajo”, “bajo”, “medio”, “alto” y “muy alto”, en orden de L_0 (5) a L_4(5) respectivamente. En el caso de los idiomas las particiones serían: L_0(6)=inglés, L_1(6)=español, L_2(6)=francés, L_3(6)=alemán, L_4(6)=portugués, L_5(6)=dutch.

Utilizando las particiones mostradas anteriormente el algoritmo obtiene un conjunto de 12 reglas difusas de las cuales se ejemplifican cuatro en la Figura 2.

```
1: PromedioDeTiempoDeEjecucion(seg) IS L_1(5): M10 CF: [0.8967, 0.9462]
2: Idioma IS L_1(6): M10 CF: [0.9530201342281881, 1.0]
3: numeroDeEjecuciones IS L_0(5): M10 CF: [0.9040841529288194, 0.9153015549864002]
4: Idioma IS L_5(6) AND PromedioDeTiempoDeEjecucion(seg) IS L_0(5): M2 CF: [0.4594, 0.5698]
```

Fig. 2 - Ejemplo de reglas difusas creadas por el algoritmo.

Utilizando el conjunto de 12 reglas difusas creadas por el algoritmo IVTURS se pueden establecer algunas asociaciones, por ejemplo, si analizamos la regla número 2, de acuerdo a las particiones creadas por el algoritmo, la misma establece que si el idioma es L_1(6), definido como español, entonces la máquina a la que se asigna el reporte es M10 con un factor de certidumbre entre 0.95 y 1. Un análisis similar puede realizarse con la regla 3, la cual establece que si el número de ejecuciones del reporte es bajo, entonces la máquina a la cual se envía es igualmente la número 10 con un factor de certidumbre entre 0.90 y 0.915. El conocimiento que se puede extraer de las reglas permite realizar una asignación apropiada de reportes a máquinas o recursos, con vistas a desarrollar posteriormente un proceso de secuenciación favorable en términos de tiempo de ejecución de las tareas a desarrollar.

Conclusiones

En este trabajo se ha presentado el uso de sistemas neuroborrosos para la clasificación de reportes en ambientes de secuenciación, lo cual constituye un paso importante para el desarrollo del proceso de secuenciación de las tareas a ejecutar por los recursos disponibles. El uso de algoritmos neuroborrosos permite obtener un conjunto de reglas difusas para la interpretación de los resultados del modelo.

Los resultados experimentales mostraron que los cuatro métodos utilizados presentan buen rendimiento en cuanto a exactitud de la clasificación. Específicamente el algoritmo IVTURS obtuvo los mejores resultados para los tres juegos de datos utilizados en la experimentación.

Al analizar las reglas difusas obtenidas por este algoritmo se puede arribar a conclusiones importantes sobre la clasificación de los reportes teniendo en cuenta el idioma o el número de ejecuciones del mismo que se solicitan, este conocimiento puede ayudar en la toma de decisiones y en la utilización de los recursos de las empresas de forma eficiente.

Referencias

- Alcala-Fdez, J, Et Al. “Keel Data-Mining Software Tool: Data Set Repository, Integration Of Algorithms And Experimental Analysis Framework.” *Journal Of Multiple-Valued Logic And Soft Computing*, 2011, 17(2-3): 255–87.
- Sanz, A., Fernandez, J. A. And Bustince, H. “Ivturs: A Linguistic Fuzzy Rule-Based Classification System Based On A New Interval-Valued Fuzzy Reasoning Method With Tuning And Rule Selection.” *Ieee Transactions On Fuzzy Systems* 2013, 21: 13.
- Cohen, W. “Fast Effective Rule Induction.” In *In Proceedings Of The Twelfth International Conference On Machine Learning*, 1995, Morgan Kaufmann, 115–23.
- Gajate Martin, A. “Modelado Y Control Neuroborroso De Sistemas Complejos. Aplicación A Procesos De Mecanizado De Alto Rendimiento.” In *Vii Simposio Cea De Control Inteligente*, 2011.
- Garcia, D., Gonzalez, A. And Perez, R. “Overview Of The Slave Learning Algorithm: A Review Of Its Evolution And Prospects.” *International Journal Of Computational Intelligence Systems*, 2014, 7(6): 1194–1221.
- Gonzalez, A., And Perez, R. “Slave: A Genetic Learning System Based On An Iterative Approach.” *Ieee Transactions On Fuzzy Systems*, 1999, 7:2: 176–92.

- Gonzalez, A., And Perez, R.. “Selection Of Relevant Features In A Fuzzy Genetic Learning Algorithm.” Ieee Transactions On Systems, Man, And Cybernetics, 2011, Part B: Cybernetics 31:3: 417–25.
- Gonzalez, A., And Perez, R. “Improving The Genetic Algorithm Of Slave.” Mathware & Soft Computing 2009, 16: 59–70.
- Graham, R. L., Lawler, L. E. Et Al. “Optimization And Approximation In Deterministic Sequencing And Scheduling: A Survey.” Annals Of Discrete Mathematics 1979, 5: 287–326.
- Grau Garcia, I., Napoles, G. Et Al. 2012. “Aplicación De Sistemas Neuroborrosos A Problemas Resistencia Antiviral Del Vih.” Revista Cubana De Ciencias Informáticas (Rcci), 2012, 6: 11.
- Herroelen, W., And Leus, R.. “Project Scheduling Under Uncertainty: Survey And Research Potentials.” European Journal Of Operational Research, 2005, 165(2): 289–306.
- Hühn, J., And Hüllermeier, E. “Furia: An Algorithm For Unordered Fuzzy Rule Induction.” Data Mining Knowledge Discovery, 2009, 19: 293–319.
- Jang, J. S. R. “Anfis: Adaptive-Network-Based Fuzzy Inference System.” Ieee Transactions On Systems, Man And Cybernetics 1993, 23: 665–85.
- Jin, Y. “Fuzzy Modeling Of High-Dimensional Systems: Complexity Reduction And Interpretability Improvement.” Ieee Transactions On Fuzzy Systems 2000, 8(2): 212–21.
- Martínez Jiménez, Y. “A Generic Multi-Agent Reinforcement Learning Approach For Scheduling Problems.” Phd Thesis, Vrije Universiteit Brussel, 2012.
- Nauck, D. “Data Analysis With Neuro - Fuzzy Methods.” Fakultat Fur Informatik: Phd Thesis, 154 Pages, 2000.
- Zhang, W. Architecture “Reinforcement Learning For Job Shop Scheduling”. Oregon State University. PHD THESIS, 1996.

Conflicto de interés

No existe

Contribuciones de los autores

Jessica Coto Palacio: Contribución relacionada con la implementación y el desarrollo de los experimentos realizados, la obtención de los juegos de datos, el análisis estadístico y la redacción del manuscrito.

Yailen Martínez Jiménez: Contribución relacionada con la conceptualización, análisis formal y asesoría en temas de secuenciación de tareas y análisis estadístico.

Ann Nowé: Contribución relacionada con la conceptualización, el desarrollo del marco teórico, asesoría en temas de clasificación y secuenciación.