

Tipo de artículo: Artículo original
Temática: Ingeniería y gestión de software
Recibido: 02/12/2016 | Aceptado: 04/05/2017

Sistema clasificador borroso basado en algoritmos genéticos para evaluar el estado de ejecución de proyectos

Fuzzy classifier system based on genetic algorithms to evaluate the state of project execution

Carlos Rafael Rodríguez Rodríguez^{1*}, Marieta Peña Abreu¹, Gilberto Fernando Castro Aguilar², Pedro Yobanis Piñero Pérez¹

¹ Universidad de Ciencias Informáticas. Carretera a San Antonio de los Baños, Km. 2 ½, Torrens, Boyeros, La Habana, Cuba. CP 19370. {crodriguezr, mpabreu, ppp}@uci.cu.

² Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil. Urbanización Belo Horizonte, Mz. 52 V. 26 - Km 11.5 vía a la Costa. Guayaquil – Guayas – Ecuador. gilberto.castroa@ug.edu.ec

* Autor para correspondencia: crodriguezr@uci.cu

Resumen

Las organizaciones suelen orientar sus objetivos hacia la gestión mediante proyectos. Para controlar la ejecución de los mismos, las organizaciones emplean diversas herramientas para facilitar la toma de decisiones. Sin embargo, aún son insuficientes en contextos con incertidumbre en la información y condiciones cambiantes en los estilos de gestión. Para afrontar esa situación es recomendable el uso de técnicas de *soft computing*. Como contribución a ello, en este trabajo se propone un método para construir un sistema clasificador borroso para evaluar el estado de ejecución de proyectos. Se experimenta con técnicas basadas en algoritmos genéticos; se analiza su forma de operación y los atributos que emplean. Se realiza una validación cruzada aleatoria con 20 iteraciones y 3 clasificadores utilizando una base de conocimientos de 204 proyectos. Para determinar la técnica de mejores resultados se calculan las métricas: cantidad de reglas generadas, porcentaje de clasificaciones correctas, cantidad de falsos positivos, cantidad de falsos negativos, error cuadrático medio, raíz del error cuadrático medio y error porcentual de la media absoluta simétrica. El principal aporte práctico es la integración a la biblioteca *AnalysisPro* de las técnicas de *soft computing* seleccionadas, mediante funciones implementadas en lenguaje PL/R que utilizan como dependencia el paquete FRBS.

Palabras clave: aprendizaje automático, evaluación de proyectos, sistema clasificador borroso, *soft computing*

Abstract

Organizations often guide their objectives towards project management. Organizations employ many tools to ease decision-making during project execution control. However, they are still insufficient in contexts with uncertainty in information and changing management style conditions. It is advisable to use soft computing techniques to face this situation. As a contribution in this context, this paper proposed a method to build a fuzzy classifier system to evaluate the state of project execution. An experiment is conducted with techniques based on genetic algorithms. Their working methods and parameters used are analyzed. A random cross-validation is made with 20 iterations and 3 classifiers using a base of knowledge with 204 projects. In order to determine the technique of better results, it calculates the metrics: quantity of generated rules, percent correct classifications, quantity of false positives, quantity of false negatives, Mean Square Error, Root Mean Square Error and Symmetric Mean Absolute Percentage Error. The main practical contribution is the integration of selected soft computing techniques into the AnalysisPro library, using functions implemented in PL/R language that use the FRBS package as a dependency.

Keywords: *fuzzy classifier system, learning machine, project evaluation, soft computing*

Introducción

La búsqueda de formas más eficaces de gestión es una necesidad que enfrentan las organizaciones. Diversos autores (Pressman, 2010; Delgado, et al., 2011; Blanco, 2011; Lage, 2013) reconocen las ventajas de orientar a proyectos los objetivos de una organización. Para el éxito de un proyecto se requiere tomar decisiones oportunamente mediante el seguimiento y control. Para ello la evaluación sistemática de indicadores es decisiva (Delgado, et al., 2011). El uso de indicadores que avalen el desempeño empresarial es de gran significado para la empresa actual (Cuesta, et al., 2014).

La Gestión de Proyectos es una actividad compleja en la que resulta necesario aplicar técnicas matemáticas especializadas, unidas a las Tecnologías de la Información y las Comunicaciones (TIC) (Blanco, 2011; Lugo, et al., 2011). Esa necesidad aumenta en entornos con grandes volúmenes de datos por procesar (White, 2012). En un estudio a 124 herramientas de Gestión de Proyectos, se obtuvo que: el 56 % no genera reportes, ni permite hacer análisis, el 21 % no ofrece funcionalidades para la planificación y el 26 % no facilita la gestión de recursos (Lugo, 2015).

Las herramientas más profesionales evalúan indicadores bien definidos, lo que facilita la detección de desviaciones. No obstante, implementan métodos tradicionales para el análisis de información; en contextos donde se identifican los siguientes aspectos: presencia de datos mezclados, incertidumbre, ruido provocado por errores de medición y la

apreciación de las personas, vaguedad en los conceptos, condiciones cambiantes en el entorno de ejecución de los proyectos y evaluación de los proyectos de forma estática y basada en el conocimiento de expertos (Bermúdez, 2015).

Varias investigaciones emplean técnicas de minería de datos, inteligencia artificial y *soft computing* para resolver algunos de esos aspectos en diversas áreas de la Gestión de Proyectos (Rodríguez, 2016). Sin embargo, la mayoría de estos aportes presenta alguna de estas dos limitaciones: las reglas de inferencia borrosas se especifican manualmente por expertos o, no están disponibles en herramientas de Gestión de Proyectos. Contribuciones más recientes (Castro, et al., 2016; Peña, et al., 2016; Torres, et al., 2016) no tratan la evaluación del estado de ejecución de proyectos.

Si la base de reglas se construye solo a partir del conocimiento de expertos, el sistema borroso (SB) podría no funcionar bien. Los expertos podrían equivocarse: al localizar determinados puntos característicos en las funciones de pertenencia, respecto al número de reglas o acerca de la no visibilidad de determinadas áreas del espacio de búsqueda (Piñero, 2005). Mientras que la segunda limitación dificulta la aplicación generalizada de esos aportes en escenarios reales de Gestión de Proyectos de manera coherente, efectiva y con un mínimo de esfuerzo por parte de los decisores.

En adición a esto, la búsqueda de formas más eficaces de gestión en la que se encuentran inmersas constantemente las organizaciones, las provee de un nivel de madurez ascendente. En este contexto pueden cambiar tanto los atributos que se controlan, como el rigor y la profundidad de los análisis. La utilización de un SB estático en este entorno es insuficiente, por su incapacidad para adaptarse a los cambios ocurridos en el medio donde se gestionan los proyectos.

Para esta problemática se ofreció una solución (Bermúdez, 2015) que utiliza técnicas para resolver problemas de regresión. Los resultados que se obtienen en la evaluación de proyectos mejoran en comparación con los de la técnica empleada en Xedro-GESPRO 13.05. No obstante, cada una de las técnicas de aprendizaje automático presenta potencialidades y limitaciones al aplicarlas en un dominio específico; por lo que de acuerdo con el teorema *No free lunch* (Wolpert, 1996) se recomienda experimentar con diferentes enfoques.

En este trabajo se propone un método para construir un sistema clasificador borroso (SCB) para evaluar el estado de ejecución de proyectos. Mediante el aprendizaje automático de reglas borrosas con técnicas de *soft computing* se evitará la limitación de utilizar reglas especificadas por expertos. La integración de estas técnicas a la biblioteca *AnalysisPro* facilitará su uso en herramientas de Gestión de Proyectos. A diferencia del trabajo anterior, se utilizan las técnicas de *soft computing* para resolver problemas de clasificación que ofrece el paquete FRBS y dentro de ellas las basadas en algoritmos genéticos. La propuesta permite construir un SCB capaz de adaptarse al nivel de madurez de la organización, con un enfoque de mejora continua y retroalimentación. El SCB mejorará los resultados de la evaluación del estado de ejecución de proyectos en comparación con el SB que utiliza Xedro-GESPRO 13.05.

Materiales y Métodos

Para la conducción del trabajo se utilizaron varios métodos de investigación. Entre los teóricos el **histórico-lógico** y el **dialéctico** permitieron realizar una revisión crítica de las principales investigaciones referidas al tema, determinar los principales precedentes de la investigación y la novedad de la misma. A través del **hipotético-de ductivo** se identificó la problemática existente y su posible solución; además se planteó el objetivo de la investigación. Por medio del **analítico-sintético** se descompuso el problema en partes más pequeñas para estudiarlo mejor; luego a través de la síntesis se formuló la propuesta de solución. Mediante la **modelación** se especificaron las actividades del método propuesto. Dentro de los empíricos el **experimental** permitió aplicar el método propuesto a una muestra seleccionada. Finalmente, el método matemático **estadística descriptiva** posibilitó el análisis de los resultados del experimento.

Método para construir un sistema clasificador borroso para la evaluación de proyectos

El método propuesto define cinco actividades (ver Figura 1) que permiten construir un SCB para evaluar el estado de ejecución de proyectos. Este proceso debe repetirse con la frecuencia que la alta gerencia determine. Se recomienda una periodicidad entre seis meses y un año. Lo que permitirá que el SCB se ajuste a los cambios en los estilos de gestión, en un ambiente de mejora continua. Más adelante, se describen las actividades.

La biblioteca *AnalysisPro* (LIGP, 2015) se adopta como base para aplicar el método. Esta integra funcionalidades y varias técnicas para el análisis de datos. Tiene como principal ventaja su alto nivel de integración a nivel de funciones con bases de datos. Implementa funciones programadas en PL/PGSQL y PL/R integrables en PostgreSQL. Utiliza como dependencias el paquete FRBS (Riza, et al., 2015) y funciones nativas de R (R Core Team, 2015). Como aporte práctico de este trabajo se integran a ella mediante funciones implementadas en PL/R las técnicas de *soft computing* basadas en algoritmos genéticos que se seleccionaron.

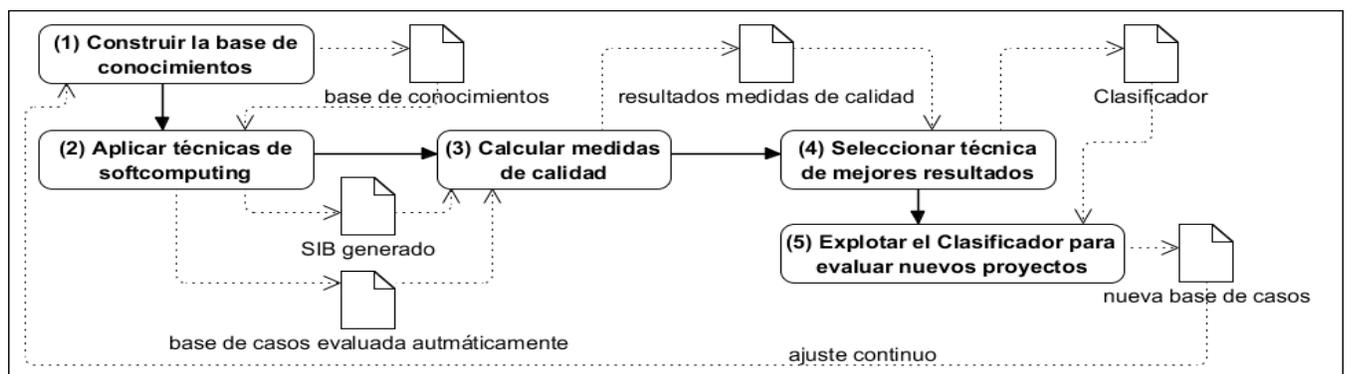


Figura 1. Método para construir un SCB para evaluar el estado de ejecución de proyectos (del autor).

Descripción de las actividades del método:

Actividad 1: Para construir la base de conocimientos debe realizarse las siguientes tareas: 1) selección de los indicadores que se valorarán, 2) recolección de los valores de los indicadores para los proyectos analizados, 3) limpieza de los datos y 4) evaluación por expertos del estado de los proyectos según los valores de los indicadores. El resultado es una base de conocimientos consistente que será la entrada de la actividad 2.

Actividad 2: Por medio de las nuevas funciones integradas a *AnalysisPro* se aplican las técnicas de *soft computing* seleccionadas (se describen más adelante). Para ello se diseñan experimentos de validación cruzada aleatoria con k iteraciones y n clasificadores, a través de funcionalidades propias de *AnalysisPro*. Para este proceso se emplean como dependencia las funciones *frbs.learn* (para el aprendizaje) y *predict* (para las pruebas) del paquete FRBS. Como resultado se almacenan los modelos generados por las técnicas, lo que incluye: estructura de las reglas, variables lingüísticas para cada indicador, los parámetros que se utilizaron y la cantidad de reglas generadas.

Actividad 3: A partir de los datos almacenados en la actividad 2, se calculan siete métricas para evaluar el desempeño del SCB construido por cada técnica en cada una de las k iteraciones. Los resultados de las métricas también se almacenan para utilizarlos en la actividad 4. Las métricas computadas se describen a continuación:

- Cantidad de reglas generadas (CR): representa la cantidad de reglas de cada SCB (se expresa en unidades).
- Porcentaje de clasificaciones correctas (% CC) formalizado en la ecuación 1:

$$\% CC = \frac{100 * \text{cant. casos bien clasif.}}{\text{total de casos}} \quad (1)$$

- Cantidad de falsos positivos (FP): resultado automático que evalúa el estado de ejecución de un proyecto de Bien cuando en realidad está evaluado de Mal (se expresa en unidades).
- Cantidad de Falsos negativos (FN): resultado automático que evalúa el estado de ejecución de un proyecto de Mal cuando en realidad está evaluado de Bien (se expresa en unidades).
- Error cuadrático medio (MSE) formalizado en la ecuación 2:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{real} - y_{pred})^2 \quad (2)$$

- Raíz del error cuadrático medio (RMSE) formalizado en la ecuación 3:

$$RMSE = \sqrt{MSE} \quad (3)$$

- Error porcentual de la media absoluta simétrica (SMAPE) formalizado en la ecuación 4:

$$100 * \frac{1}{n} \sum_{i=1}^n \frac{|y_{real} - y_{pred}|}{(y_{real} + y_{pred})/2} \quad (4)$$

En las ecuaciones (2 - 4): n es la cantidad de casos, y_{real} es el valor de salida esperada y y_{pred} es la salida calculada.

Actividad 4: Se ejecutan pruebas estadísticas de rigor a partir de los resultados de la actividad anterior, para identificar diferencias significativas entre las técnicas y formar grupos convenientes. Para realizar estas pruebas se utilizan las funciones *shapiro.test(data)*, *friedman.test(data)* y *wilcox.test(data[,a],data[,b],paired=TRUE)* del paquete estadístico R incluidas en *AnalysisPro*. Finalmente, se selecciona y obtiene el SCB, que mejor representa el estado actual de la organización en cuanto a su madurez y estilo de gestión. La madurez y estilo de gestión están implícitos en la base de conocimientos según los indicadores que se consideran y la evaluación que dan los expertos.

Actividad 5: Se emplea el SCB para evaluar periódicamente el estado de ejecución de los proyectos. Esto ocurrirá de manera continua hasta que la alta gerencia considere que el SCB debe ser ajustado a la nueva realidad de la organización. Durante ese periodo, los resultados de la evaluación se almacenan en una base de datos. Al decidirse que es necesario el ajuste del SCB, la nueva base de conocimientos se construirá a partir de los datos almacenados. Esto último asegura la mejora continua del SCB.

Seguidamente se describen las tres técnicas de *soft computing* basadas en algoritmos genéticos citadas en la Actividad 2. Estas técnicas se incluyen en el paquete FRBS, cuyos autores las consideran entre las más extendidas para construir y usar sistemas basados en reglas difusas (Riza, et al., 2015). Dentro de las cinco técnicas de clasificación que ofrece FRBS, en trabajos previos (Rodríguez, 2016) se constató que estas tres arrojan mejores resultados para este contexto.

Sistema borroso basado en aprendizaje cooperativo - competitivo genético (*gfs.gccl*)

En la técnica *gfs.gccl* (Ishibuchi, et al., 1999) un cromosoma describe cada regla difusa y utiliza valores enteros para representar los antecedentes. En la parte consecuente de las reglas difusas, se aplica el método heurístico para generar automáticamente la clase. La evaluación se calcula para cada regla, lo que significa que el rendimiento no se basa en todo el conjunto de reglas. Para manejar datos de alta dimensión, la técnica utiliza etiquetas "*don't care*" en los conjuntos borrosos de los antecedentes. La secuencia de actividades que sigue la técnica es la siguiente:

Paso 1: Generar una población inicial de N_{pop} reglas difusas especificando al azar los conjuntos borrosos de los antecedentes de cada regla. La clase consecuente y el grado de certeza se determinan por un procedimiento heurístico.

Paso 2: Clasificar todos los casos de entrenamiento dados, con las reglas difusas de la población actual; luego calcular la aptitud de cada regla.

Paso 3: Generar N_{rep} nuevas reglas difusas aplicando operadores genéticos a la población actual. La clase consecuente y el grado de certeza de cada nueva regla se determinan utilizando el mismo procedimiento heurístico.

Paso 4: Reemplazar las N_{rep} reglas difusas que menor valor de aptitud tienen en la población actual con las nuevas reglas generadas.

Paso 5: Terminar si se cumple la condición de parada preestablecida, si no regresar al paso 2.

Para aplicar esta técnica deben especificarse los valores de los siguientes parámetros: datos de entrenamiento, rango de los datos, número de clases, número de etiquetas lingüísticas, tamaño de la población, probabilidad de cruzamiento, probabilidad de mutación, y número máximo de generaciones.

Sistema híbrido genético de los enfoques Michigan y Pittsburgh (*fh.gbml*)

La técnica *fh.gbml* (Ishibuchi, et al., 2005) propone un algoritmo híbrido de los enfoques Michigan y Pittsburgh para el aprendizaje automático basado en genética. La estructura del algoritmo se basa en la del enfoque Pittsburgh, pero incorpora el estilo Michigan para modificar parcialmente cada individuo (conjunto de reglas). De esa manera se combinan las ventajas de cada enfoque para obtener un sistema de mejor rendimiento en la clasificación. El propósito de usar el estilo Michigan es aprovechar su alta capacidad para encontrar de manera eficiente buenas reglas difusas. Mientras que el estilo Pittsburgh se emplea dada su habilidad para encontrar una buena combinación de reglas difusas, es decir, optimizar el sistema. La secuencia de actividades que sigue la técnica es la siguiente:

Paso 1: Generar una población inicial de n individuos, donde cada individuo es un conjunto de m reglas difusas.

Paso 2: Calcular la actitud de cada individuo de la población actual.

Paso 3: Generar $n-1$ nuevos individuos mediante operadores genéticos de la misma manera que el enfoque Pittsburgh. Aplicar una iteración simple del enfoque Michigan a cada nuevo individuo con una probabilidad predeterminada.

Paso 4: Añadir el mejor individuo de la población actual a los $n-1$ nuevos individuos generados para formar la próxima población.

Paso 5: Retornar al paso 2 si la condición de parada previamente definida no se cumple.

Para aplicar esta técnica deben especificarse los valores de los siguientes parámetros: datos de entrenamiento, rango de los datos, número de clases, tamaño de la población, máximo número de reglas, probabilidad de cruzamiento, probabilidad de mutación, número máximo de generaciones, número máximo de iteraciones, probabilidad de ocurrencia de una etiqueta "don't care" y probabilidad de ocurrencia del GCCL.

Algoritmo de aprendizaje estructural en ambientes de vaguedad (*slave*)

La técnica *slave* (González, et al., 2001) se basa en el enfoque de aprendizaje iterativo de reglas (IRL). Esta técnica utiliza una sola regla difusa en cada ejecución del algoritmo genético. Con el fin de eliminar las variables irrelevantes en una regla, *slave* tiene una estructura compuesta de dos partes: la primera para representar la relevancia de las variables y la segunda para definir los valores de los parámetros. Utiliza códigos binarios como representación de la población y aplica los operadores genéticos (selección, cruzamiento y mutación) sobre esa población. La mejor regla se obtiene calculando el grado de consistencia y completitud. La secuencia de actividades de la técnica es la siguiente:

Paso 1: Construir una regla difusa mediante el procedimiento de un algoritmo genético.

Paso 2: Adicionar la regla al conjunto final de reglas

Paso 3: Chequear y penalizar la regla

Paso 4: Si se cumple la condición de parada, devolver el conjunto de reglas como solución, si no regresar al paso 1.

Para aplicar esta técnica deben especificarse los valores de los siguientes parámetros: datos de entrenamiento, rango de los datos, número de clases, número de etiquetas lingüísticas, tamaño de la población, probabilidad de cruzamiento, probabilidad de mutación, número máximo de generaciones, número máximo de iteraciones, límites inferior y superior del umbral y factor de cubrimiento (épsilon).

Materiales y métodos utilizados en el experimento

Para el experimento se utiliza una base de conocimientos de 204 proyectos (terminados o en ejecución) gestionados con la herramienta Xedro-GESPRO 13.05. Los proyectos pertenecen a la Universidad de Ciencias Informáticas, certificada con el nivel 2 de CMMI. Cada proyecto tiene como atributos los valores de cinco indicadores propuestos en (Lugo, 2015). Esos valores se calcularon cada semana automáticamente, lo que asegura su confiabilidad y actualización con un mínimo de esfuerzo y errores. Estos indicadores son:

- Índice de Rendimiento de la Ejecución (IRE). Adquiere valores entre 0 y 2.
- Índice de Rendimiento de la Planificación - con adelantos (IRPa). Adquiere valores entre 0 y 2.
- Índice de Rendimiento de la Eficacia (IREF). Adquiere valores entre 0 y 1.
- Índice de Calidad del Dato (ICD). Adquiere valores entre 0 y 1.
- Índice de Rendimiento de los Recursos Humanos (IRRH). Adquiere valores entre 0 y 100.

Además, cada proyecto tiene la evaluación dada por los expertos según el estado de los indicadores. Esa evaluación

presenta la siguiente distribución: 63 evaluados de Bien (31 %), 92 de Regular (45 %) y 49 de Mal (24 %). Ninguno de los casos contiene valores ausentes ni fuera de rango.

Para el desarrollo del experimento se aplica una validación cruzada aleatoria con 20 iteraciones y 3 clasificadores, este último parámetro representa las tres técnicas de *soft computing* a analizar. Por cada iteración, el juego de datos se divide aleatoriamente en un conjunto de entrenamiento con el 75 % de los casos y otro de prueba con el 25 % restante. Se analizarán las métricas descritas en la actividad 2 del método propuesto: CR, % CC, FP, FN, MSE, RMSE y SMAPE. Los parámetros con los que cada técnica se configura se especifican en la Tabla 1. Algunos se fijaron dada la naturaleza del problema y otros se determinaron luego de varias pruebas para trabajos previos.

Tabla 1. Parámetros configurados en cada técnica para el experimento (del autor).

Técnica	Parámetros (nombre = valor)
<i>gfs.gccl</i>	num_class = 3; num_labels = 3; popu_size = 20; persen_cross = 0,9; persen_mutant = 0,3; max_gen = 50
<i>fh.gbml</i>	num_class = 3; popu_size = 50; max_num_rule = 25; persen_cross = 0,9; persen_mutant = 0,3; max_gen = 100; p_dcare = 0,5; p_gccl = 1
<i>Slave</i>	num_class = 3; num_labels = 3; max_iter = 20; k_lower = 0,25; k_upper = 0,75; persen_cross = 0,9; persen_mutant = 0,3; max_gen = 50; epsilon = 0,1

Para comparar los resultados de las técnicas respecto a cada una de las métricas se comprobó el supuesto de normalidad de las muestras mediante el test Shapiro-Wilk. Este arrojó que por cada métrica al menos una técnica posee valores de probabilidad (p-valor) inferiores a 0,05, lo que rechaza la idea de distribución normal con un 95 % de confianza. Esto sugiere la utilización de pruebas no paramétricas para comprobar las diferencias significativas. Como las muestras fueron obtenidas a partir de una misma base de conocimientos se consideran relacionadas, por lo que se utilizan los test de Friedman y Wilcoxon. Para ambos test se adopta un umbral de 0,05 para determinar la diferencia significativa. En el test de Wilcoxon se utiliza el método de Monte Carlo con un 99 % en el intervalo de confianza. En los casos que el test de Friedman encuentre diferencias significativas entre las técnicas, se forman grupos. En ellos se cumple que: las técnicas de un mismo grupo no tienen diferencias significativas entre ellas, las técnicas en los grupos menores (Grupo1 < Grupo2 < ...) reportan mejores resultados respecto a la métrica analizada.

Resultados y discusión

En la Figura 2 se ofrecen los resultados de cada técnica para la métrica CR. Mediante el test de Friedman se encontraron diferencias significativas (p-valor = 1,361e-07). Al aplicar el test de Wilcoxon los grupos resultantes fueron: Grupo1: *slave*, *fh.gbml*; Grupo2: *gfs.gccl*. Esto significa que las técnicas *slave* y *fh.gbml* ofrecen los mejores

resultados. El estadígrafo descriptivo muestra que *slave* tiene una media de 9,3 inferior a la de *fh.gbml* que es de 11,5. Sin embargo la desviación estándar tiene un comportamiento opuesto, *fh.gbml* tiene 1,5 mientras *slave* alcanza 4,3.

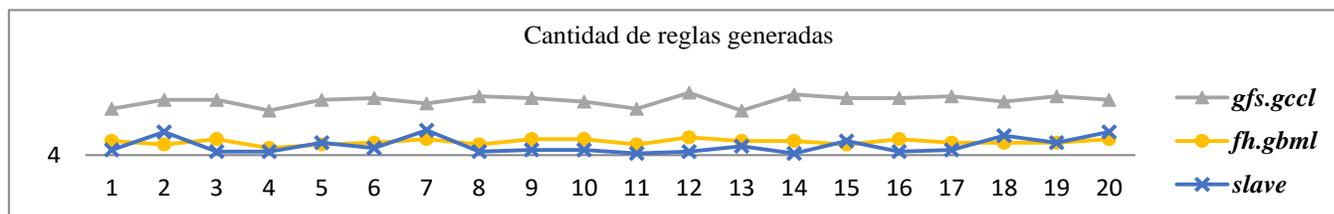


Figura 2. Cantidad de reglas generadas en 20 iteraciones sobre el conjunto de entrenamiento (del autor).

Nota: La leyenda de esta figura será empleada en las restantes.

Esta métrica no influye directamente en la calidad de la evaluación, pero sí en el tiempo necesario para ejecutar el SCB en el momento de la evaluación. A mayor cantidad de reglas, el SCB demora más tiempo para evaluar nuevos proyectos. Por lo que es recomendable que, entre SCB con similar rendimiento se utilice el que menor número de reglas contenga. En el ámbito de este trabajo se utilizarían *slave* o *fh.gbml*.

La Figura 3 muestra los resultados de cada técnica para la métrica % CC. Para el conjunto de entrenamiento el test de Friedman encontró diferencias significativas (p-valor = 0,0002324). Al aplicar Wilcoxon se obtuvieron los grupos: Grupo1: *fh.gbml*; Grupo2: *gfs.gccl*, *slave*; por lo que *fh.gbml* es significativamente superior. El estadígrafo descriptivo de *fh.gbml* muestra una media de 92,25 % y una desviación estándar de 3,38. En el conjunto de prueba el test de Friedman no encontró diferencias significativas. No obstante, *fh.gbml* obtuvo la media más alta con 87,84 %.

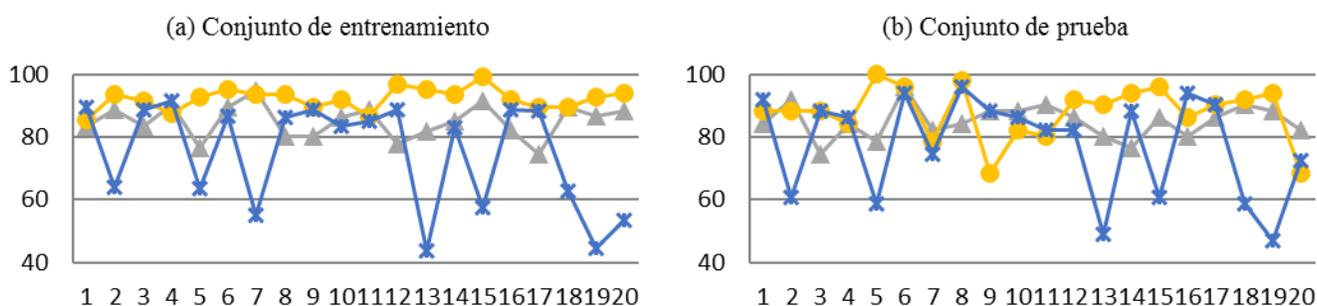


Figura 3. Porcentaje de clasificaciones correctas en 20 iteraciones para ambos conjuntos (del autor).

Esta variable es, a consideración de los autores, la más representativa de la calidad del SCB construido, dado que está directamente ligada a la certeza con la que los decisores podrán usar los resultados. En ambos conjuntos la técnica

fh.gbml obtiene los mejores valores de media: 92,25 % en entrenamiento y 87,84 % en pruebas.

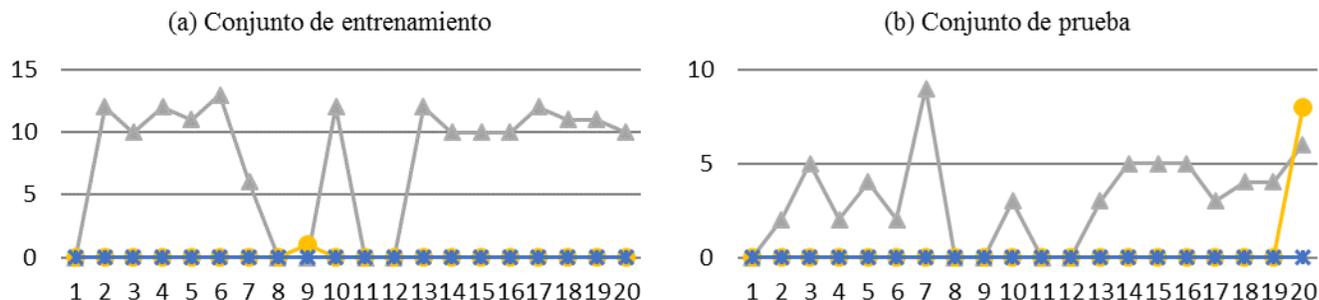


Figura 4. Cantidad de falsos positivos en 20 iteraciones para ambos conjuntos (del autor).

En la Figura 4 se muestran los resultados de cada técnica para la métrica FP. En ambos conjuntos el test de Friedman encontró diferencias significativas (p -valor = $1,874e-06$ en entrenamiento y p -valor = $2,575e-06$ en pruebas). Wilcoxon agrupó las técnicas de la misma manera: Grupo1: *slave*, *fh.gbml*; Grupo2: *gfs.gccl*. Por lo que *slave* y *fh.gbml* obtienen los menores valores y su diferencia no es significativa. Clasificar proyectos como FP constituye un riesgo negativo que puede impactar las principales áreas de la Gestión de Proyectos (alcance, tiempo, costo, calidad). Por lo que se sugiere utilizar las técnicas *slave* o *fh.gbml* que permiten reducir su probabilidad de ocurrencia.

Al analizar los valores de cada técnica para la métrica FN se observaron sobre ambos conjuntos resultados muy similares. El test de Friedman no encontró diferencias significativas entre los resultados sobre ninguno de los dos conjuntos. Por ello fue necesario evaluar si los valores arrojados eran satisfactorios o no. La técnica *gfs.gccl* no clasifica ningún caso como FN. Sin embargo, las técnicas *fh.gbml* y *slave* (que obtuvieron los mejores resultados al analizar los FP) si obtienen valores de FN. La técnica *slave* tuvo una media de 0,15 en entrenamiento y de 0,10 en pruebas; *fh.gbml* tuvo una media de 0,30 para ambos conjuntos. Obtener FN también es un resultado poco deseable, aunque en menor medida que los FP. Estos casos también representan un riesgo que desencadena acciones para solucionar un problema que realmente no existe y por tanto significa el empleo innecesario de recursos y tiempo.

Sobre los resultados de las técnicas para la métrica MSE (Figura 5), el test de Friedman encontró diferencias estadísticamente significativas en ambos conjuntos (p -valor = 0,0001174 en entrenamiento y p -valor = 0,003929 en pruebas). El test de Wilcoxon para el conjunto de entrenamiento agrupó las técnicas así: Grupo1: *fh.gbml*; Grupo2: *slave*, *gfs.gccl*. Lo que significa que *fh.gbml* obtuvo resultados significativamente mejores, con una media de 0,084 y una desviación estándar de 0,039. Para el conjunto de prueba los grupos quedaron como sigue: Grupo1: *fh.gbml*,

slave; Grupo2: *gfs.gccl*. Esto evidencia que no existió diferencia estadísticamente significativa entre *fh.gbml* y *slave*; no obstante, *fh.gbml* mostró valores más bajos de media (0,13) y desviación estándar (0,09).

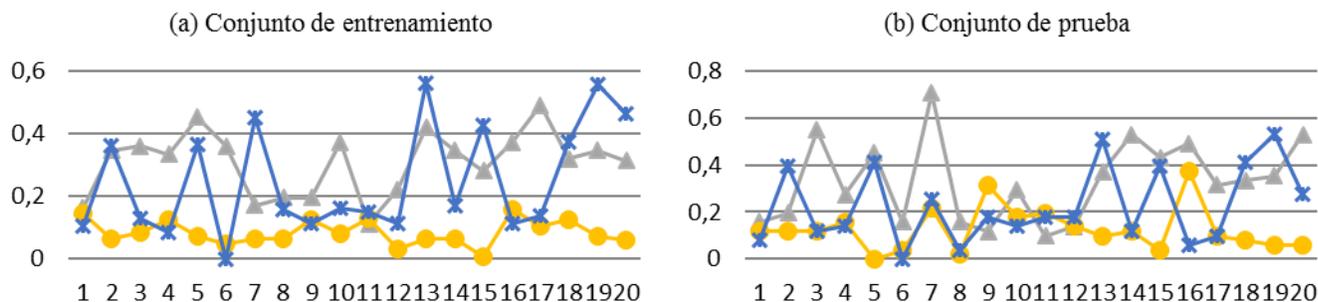


Figura 5. Resultados de la métrica MSE en 20 iteraciones para ambos conjuntos (del autor).

En la Figura 6 se representan los valores de las técnicas para la métrica RMSE. En ambos conjuntos se encontraron diferencias significativas mediante el test de Friedman (p-valor = 0,0001174 en entrenamiento y p-valor = 0,003929 en pruebas), por lo que se aplicó el test de Wilcoxon. Para el entrenamiento los grupos formados fueron: Grupo1: *fh.gbml*; Grupo2: *slave*, *gfs.gccl*. Por lo que se puede afirmar que *fh.gbml* obtiene los valores menores, con una media de 0,28 y una desviación estándar de 0,74. En el conjunto de prueba las técnicas se agruparon así: Grupo1: *fh.gbml*, *slave*; Grupo2: *gfs.gccl*. Lo que demuestra que no hay diferencias significativas entre *fh.gbml* y *slave*. No obstante, el estadígrafo descriptivo arroja que *fh.gbml* alcanza valores más bajos de media (0,33) y desviación estándar (0,14).

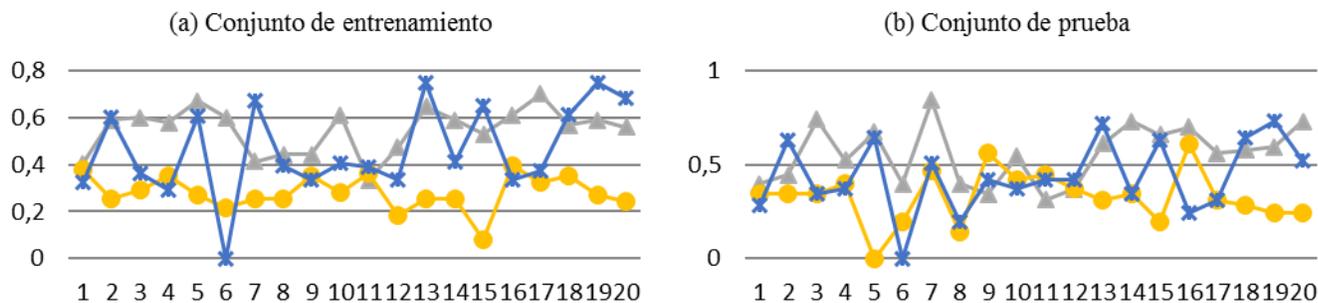


Figura 6. Resultados de la métrica RMSE en 20 iteraciones para ambos conjuntos (del autor).

Finalmente la Figura 7 muestra los resultados de la métrica SMAPE. Para el conjunto de entrenamiento el test de Friedman encontró diferencias significativas con p-valor = 5,545e-05. Al aplicar el test de Wilcoxon los grupos formados fueron: Grupo1: *fh.gbml*; Grupo2: *gfs.gccl*, *slave*. Lo que indica que *fh.gbml* tiene resultados estadísticamente mejores, con una media de 1,2 y una desviación estándar de 0,56. Para el conjunto de prueba no se encontraron diferencias estadísticamente significativas; los valores de media y desviación estándar fueron: *slave*

(2,90; 1,82), *fh.gbml* (1,74; 1,27) y *gfs.gccl* (2,88; 1,07).

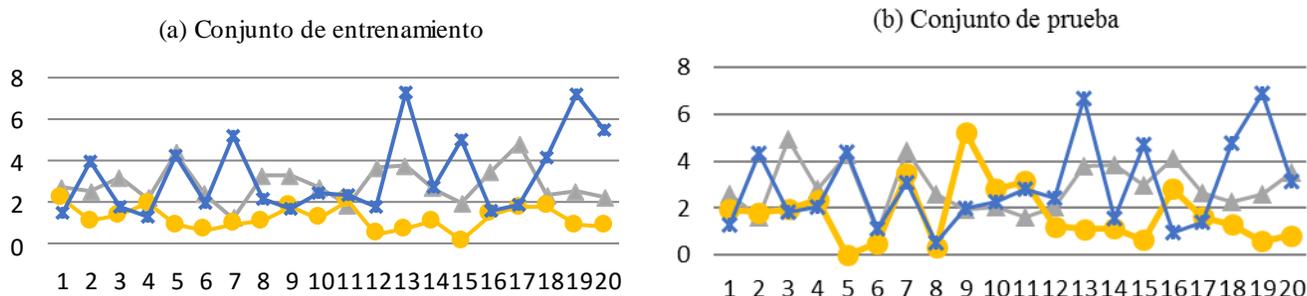


Figura 7. Resultados de la métrica SMAPE en 20 iteraciones para ambos conjuntos (del autor).

Al analizar los resultados de las siete métricas, se observa que la técnica *fh.gbml* se ubicó en el grupo 1 para todas las métricas y se considera, por tanto, la que mejor desempeño tuvo en el experimento. La técnica *slave* se ubicó en el grupo 2 solo en cuatro observaciones, en las demás se ubicó en el grupo 1.

El SCB que se obtiene está formado por 13 reglas y se construyó con funciones de pertenencia triangulares, como t-norma y s-norma el par (product-max). Como función de implicación se empleó Zadeh. En la Tabla 2 se muestran los conjuntos borrosos para cada una de las 14 etiquetas lingüísticas de las cinco variables de entrada. Las reglas que se obtuvieron muestran la estructura que se presenta a continuación:

IF ire is v.1_a.12 and irpa is dont_care and icd is v.3_a.1 and irrh is v.4_a.10 and iref is v.5_a.1 then eval is 1

Tabla 2. Conjuntos borrosos del SCB que se obtuvo (del autor).

	a.1	a.2	a.3	a.4	a.5	a.6	a.7	a.8	a.9	a.10	a.11	a.12	a.13	a.14
Punto_A	0	0	0	0	0,5	0	0	0,33	0,66	0	0	0,25	0,5	0,75
Punto_B	0	1	0	0,5	1	0	0,33	0,66	1	0	0,25	0,5	0,75	1
Punto_C	1	1	0,5	1	1	0,33	0,66	1	1	0,25	0,5	0,75	1	1

Con el objetivo de comparar el desempeño del SCB que se obtiene, con algunas propuestas anteriores, se realiza un experimento consistente en evaluar toda la base de conocimiento utilizando: el SCB obtenido en este trabajo con la técnica *fh.gbml*, el SB de Xedro-GESPRO 13.05 y el SCB obtenido con la técnica *anfis* en (Bermúdez, 2015). Se analizan los valores en por ciento de las métricas (CC, FP y FN) y los valores de las métricas MSE, RMSE y SMAPE. En la Tabla 3 se muestra para cada sistema el valor del criterio analizado y su orden (*ranking*) según ese criterio.

Luego de realizar el experimento, se pudo constatar que el *scb_fh.gbml* consiguió resultados superiores a los de

sb_gespro13.05 y *scb_anfis*. Se demuestra que al utilizar el sistema *scb_fh.gbml*, existe un incremento en la cantidad de proyectos clasificados correctamente y una disminución de los restantes criterios analizados. Esto sugiere una mejora en la calidad de la evaluación del estado de ejecución de proyectos.

Tabla 3. Resultados de la comparación del SCB que se obtuvo con sus precedentes (del autor).

Sistemas	% CC		% FP		% FN		MSE		RMSE		SMAPE	
	valor	lugar										
<i>scb_fh.gbml</i>	98,53	1	0	1	0	1	0,01	1	0,12	1	0,25	1
<i>sb_gespro13.05</i>	47,49	3	17,64	3	32,84	3	0,65	3	0,80	3	65,35	3
<i>scb_anfis</i>	95,66	2	0	1	2,94	2	0,03	1	0,14	2	2,35	2

El resultado de este trabajo ofrece un sistema clasificador borroso para evaluar el estado de ejecución de proyectos atendiendo a los principales indicadores de esa disciplina. Este sistema se ha integrado a la biblioteca *AnalysisPro* lo que facilita su uso desde las herramientas de Gestión de Proyectos que se emplean en el sistema empresarial cubano; por lo que supone la posibilidad de contribuir a la implementación de los lineamientos 7, 8, 119, 122 y 124 aprobados en el 7mo. Congreso del Partido Comunista de Cuba.

Conclusiones

En el presente trabajo se ha propuesto un método para construir un SCB basado en algoritmos genéticos para evaluar el estado de ejecución de proyectos. Atendiendo a los resultados obtenidos se arriba a las siguientes conclusiones:

- La mayoría de los aportes anteriores sobre el uso de la *soft computing* en la toma de decisiones en la Gestión de Proyectos utilizan sistemas borrosos estáticos especificados por expertos y no se integran con herramientas informáticas, lo que dificulta su uso efectivo y generalizado.
- El método propuesto facilita el tratamiento de la incertidumbre contenida en la información y la adaptación del SCB a los cambios en los estilos de gestión de la organización, en un ambiente de mejora continua.
- Las técnicas utilizadas aprenden reglas automáticamente, no especificadas por expertos. Su integración a la biblioteca *AnalysisPro* constituye una oportunidad de mejora para las herramientas de Gestión de Proyectos.
- La técnica *fh.gbml* fue la única que se ubicó en el grupo 1 para todas las métricas analizadas durante la experimentación y se considera, por tanto, la de mejor desempeño.
- El SCB que se obtuvo mejora los resultados de la evaluación del estado de ejecución de proyectos en comparación con el SB que utiliza Xedro-GESPRO 13.05.
- Como trabajos futuros se plantea utilizar técnicas de aprendizaje no supervisado y comparar los resultados.

Referencias

- BERMÚDEZ, ANIE. 2015. *Sistema basado en técnicas de soft computing para la evaluación de la ejecución de proyectos*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2015. pág. 80, Tesis de Maestría.
- BLANCO, LÁZARO J. 2011. *La informática en la dirección de empresas*. La Habana : Editorial Félix Varela, 2011. ISBN 978-959-07-1629-4.
- CASTRO, G., PÉREZ, I., PIÑERO, P.Y. AND GARCÍA, R. 2016. Aplicación de la minería de datos anómalos en organizaciones orientadas a proyectos. *Revista Cubana de Ciencias Informáticas*. 2016. Vol. 10, Especial UCIENCIA. ISSN: 2227-1899.
- CUESTA, ARMANDO AND VALENCIA, MARINO. 2014. *Indicadores de Gestión del Capital Humano y del Conocimiento en la empresa*. La Habana : Editorial Academia, 2014. ISBN 978-959-270-310-0.
- DELGADO, R., GARCÍA, J., DELFINO, A. AND MEDINA, M. 2011. *La Dirección Integrada de Proyecto como Centro del Sistema de Control de Gestión en el Ministerio del Poder Popular*. Caracas : CENDA, 2011.
- GONZÁLEZ, A. AND PÉREZ, R. 2001. Selection of relevant features in a fuzzy genetic learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2001. Vol. 31, 3, pp. 417-425.
- ISHIBUCHI, H., NAKASHIMA, T. AND MURATA, T. 1999. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*. 1999. Vol. 29, 5, pp. 601 - 618.
- ISHIBUCHI, H., YAMAMOTO, T. AND NAKASHIMA, T. 2005. Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*. 2005. Vol. 35, 2, pp. 359-365.
- LAGE, AGUSTÍN. 2013. Los procesos de dirección en la empresa de alta tecnología. [ed.] Aldo Gutiérrez. *La Economía del Conocimiento y el Socialismo*. La Habana : Editorial Academia, 2013, 9.
- LIGP. 2015. *Biblioteca para el Análisis de Datos AnalysisPro*. Laboratorio de Investigaciones en Gestión de Proyectos, Universidad de las Ciencias Informáticas. La Habana : s.n., 2015.

- LUGO, J.A. AND GARCÍA, A.M. 2011. Colección automática de métricas hacia un repositorio de mediciones. *Revista Facultad Ingeniería Universidad de Antioquia*. 2011. Vol. D, 58, pp. 199-207. ISSN 0120-6230.
- LUGO, JOSÉ A. 2015. *Modelo para el control de la ejecución de proyectos basado en soft computing*. Universidad de Ciencias Informáticas. La Habana : s.n., 2015. pág. 147, Tesis doctoral.
- PEÑA, M., GARCÍA, R., RODRÍGUEZ, C.R., AND PIÑERO, P.Y. 2016. Criterios económicos borrosos para el análisis de factibilidad de proyectos de software en ambientes de incertidumbre. *Revista Cubana de Ciencias Informáticas*. 2016. Vol. 10, Especial UCIENCIA. ISSN: 2227-1899.
- PIÑERO, P. Y. 2005. *Modelo para el aprendizaje y la clasificación automática basado en técnicas de soft computing*. Santa Clara : Universidad Martha Abreu, 2005. Tesis doctoral.
- PRESSMAN, R. S. 2010. *Software Engineering: A practitioner's approach*. 7th edition. New York : McGraw-Hill, 2010.
- R CORE TEAM. 2015. *R: A Language and Environment for Statistical Computing*. Viena, Austria : R Foundation for Statistical Computing, 2015.
- RIZA, L.S., BERGMEIR, CH., HERRERA, F. AND BENÍTEZ, J.M. 2015. frbs: Fuzzy Rule-based Systems for Classification and Regression Tasks. *Journal of Statistical Software*. 2015. Vol. 65, 1, pp. 1-30.
- RODRÍGUEZ, C. 2016. *Aprendizaje de reglas difusas para la evaluación de proyectos basado en técnicas de clasificación*. Universidad de Ciencias Informáticas. La Habana : s.n., 2016. pág. 78, Tesis de maestría.
- TORRES, S., ALDANA, M.L., PIÑERO, P.Y. AND PIEDRA, L.A. 2016. Red neuronal multicapa para la evaluación de competencias laborales. *Revista Cubana de Ciencias Informáticas*. 2016. Vol. 10, Especial UCIENCIA. ISSN: 2227-1899.
- WHITE, T. 2012. *Hadoop: The definitive guide*. 3th edition. California : O'Reilly Media, 2012.
- WOLPERT, D. 1996. The Lack of A Priori Distinctions Between Learning Algorithms. s.l : Massachusetts Institute of Technology, 1996. Vol. 8, 7, pp. 1341-1390. ISSN: 0899-7667.