Universidad de Oviedo
Programa de Doctorado en Informática

Aprendizaje de estrategias inteligentes para la optimización energética en dispositivos heterogéneos de computación

Alberto Cocaña Fernández

Directores: Dr. José Ranilla Pastor
Dr. Luciano Sánchez Ramos

Esta Memoria de Tesis se presenta como requisito para optar al grado de *Doctor en Informática*

Oviedo Mayo 2017

University of Oviedo
PhD Program in Computer Science

Energy optimization of heterogeneous computing devices through learning of intelligent strategies

Alberto Cocaña Fernández

Advisors: Dr. José Ranilla Pastor
Dr. Luciano Sánchez Ramos

This dissertation is submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy in Computer Science

Oviedo May 2017

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: Aprendizaje de estrategias inteligentes para la optimización energética en dispositivos heterogéneos de computación	Inglés: Energy optimization of heterogeneous computing devices through learning of intelligent strategies

2.- Autor	
Nombre: Alberto Cocaña Fernández	DNI/Pasaporte/NIE:
Programa de Doctorado: Doctorado en Informática	
Órgano responsable: Centro Internacional de Postgrado	

FOR-MAT-VOA-010-BIS-2

RESUMEN (en español)

El nivel tecnológico actual ha permitido el desarrollo de sofisticados dispositivos cuyas capacidades de cómputo, especialización y bajos costes han fomentado su adopción tanto en la rutina diaria como en la resolución de problemas fundamentales de las ciencias y la ingeniería, la provisión de servicios o la extracción de conocimiento, convirtiéndose de esta forma en uno de los pilares de la sociedad moderna. La evolución de estos dispositivos hardware y del software ejecutado sobre los mismos, se ha centrado exclusivamente en el rendimiento, lo que es insostenible debido a los efectos colaterales tanto energéticos, como económicos o medioambientales, entre otros. Por ello, en esta disertación se han estudiado las principales limitaciones inherentes al enfoque tradicional sustituyéndolo por un nuevo paradigma de optimización multiobjetivo que tiene en cuenta los efectos colaterales y costes incurridos, en pro de construir soluciones software e infraestructuras de cómputo sostenibles. En concreto, se puso el foco sobre dos aspectos de gran impacto: la eficiencia energética en los grandes sistemas de cómputo y la eficiencia computacional de los algoritmos y técnicas de aprendizaje automático.

El primer aspecto fue abordado mediante el desarrollo de mecanismos de adaptación dinámica para los clústeres de Computación de Alto Rendimiento, optimizando conjuntamente el rendimiento computacional y los efectos derivados de los elevados consumos de energía. En concreto, se diseñó un mecanismo reactivo basado en un Sistema Borroso Genético Híbrido aprendido por medio de Algoritmos Evolutivos Multiobjetivo. Validado experimentalmente sobre un escenario real, este mecanismo mejoró notablemente los ahorros de energía alcanzados con los Sistemas Basados en el Conocimiento propuestos por otros autores, a la vez que se alinea de forma precisa con los criterios definidos para la operación del clúster. Este mecanismo se implementó por medio de una herramienta software registrada y distribuida libremente, para permitir un uso productivo del mismo. También se trabajó sobre la mejora de la eficiencia ecológica, poniendo el foco sobre una optimización del rendimiento y del impacto medioambiental del ciclo de vida de los elementos del clúster. Posteriormente, para mejorar los resultados en escenarios donde la carga de trabajo fluctúa con una estacionalidad previsible, se diseñó un mecanismo proactivo que reajusta el clúster optimizando los recursos en base a una predicción de la carga de trabajo futura, y de acuerdo con una función de utilidad que modela los criterios de operación establecidos. Este mecanismo proactivo también fue objeto de dos contratos de investigación y de transferencia de conocimiento.

El segundo aspecto se trató mediante el diseño de algoritmos de aprendizaje automático capaces de optimizar conjuntamente precisión y costes asociados con la clasificación de las instancias. En concreto, se diseñó un clasificador multietapa sensible al coste, basado en Reglas Borrosas y aprendido por medio de Programación por Recocido Simulado Multiobjetivo. Ese nuevo clasificador se validó experimentalmente en dispositivos portátiles, donde se pudo cuantificar el efecto positivo de su eficiencia computacional sobre la duración de las baterías, frente a otras alternativas de clasificadores disponibles en la literatura de aprendizaje automático.

RESUMEN (en Inglés)

Technological advances have led to the development of sophisticated computing devices, whose processing capabilities, degree of specialisation, and low acquisition costs, fostered their irruption into the daily routine as well as being key in solving fundamental problems in science or engineering, providing general-purpose IT services and enabling knowledge extraction from raw data, thus becoming a pillar of modern society. The evolution of hardware and software has focused exclusively in the pursuit of raw performance, what is inherently unsustainable due to its collateral effects whether these are power, economical o environmental-related. Because of this, the goal of this dissertation is to tackle the main limitations implicit in the traditional development approach by superseding it with a new paradigm of multiobjective optimization aware of all collateral effects and costs involved, in the pursuit of building sustainable software solutions and computing infrastructures. Specifically, two problems were researched: the energy efficiency of large computing systems and the computational efficiency of machine learning techniques and algorithms.

The first problem was addressed through the development of mechanisms for the dynamic allocation of computing resources in High Performance Clusters, jointly optimising computational performance and power consumption. Precisely, a new reactive decision-making mechanism based on Hybrid Genetic Fuzzy Systems and learned by means of Multiobjective Evolutionary Algorithms was designed to improve energy efficiency in real-word clusters. Experimental validation showed that this new mechanism archived greater power savings than those proposed in the literature, while complying with a set of subject preferences defined for the cluster operation in terms of service quality and reliability. This new mechanism was implemented in a fully-functional, registered and publicly accessible software tool enabling its productive use. Then, efforts were focused on improving ecological efficiency by balancing performance and the environmental impact related to the life cycle of the cluster's computing and support equipment. Lastly, a proactive mechanism was designed to improve energy efficiency in scenarios where the workload features foreseeable stationary fluctuations. This mechanism optimises resource allocation over a temporal horizon, having the future workload forecast by a model of the cluster environment, and assessing every potential decision with a utility function leaned through multiobjective optimization, and tasked with modelling the subjective preferences regarding cluster operation. This proactive mechanism was also the object two knowledge transfer contracts.

The second problem was solved by designing machine learning algorithms capable of jointly optimising accuracy and classification-related costs. Specifically, a cost-conscious Multistage Fuzzy Rule-Based Classifier leaned by means of Multiobjective Simulated Annealing Programming was designed to achieve computational and cost-efficient classification. Experimentation done with battery-powered devices showed how this new classifier achieved better results in terms of classification performance and battery life than the alternatives proposed by other authors.

Resumen

El nivel tecnológico actual ha permitido el desarrollo de sofisticados dispositivos cuyas capacidades de cómputo, especialización y bajos costes han fomentado su adopción tanto en la rutina diaria como en la resolución de problemas fundamentales de las ciencias y la ingeniería, la provisión de servicios o la extracción de conocimiento, convirtiéndose de esta forma en uno de los pilares de la sociedad moderna. La evolución de estos dispositivos hardware y del software ejecutado sobre los mismos, se ha centrado exclusivamente en el rendimiento, lo que es insostenible debido a los efectos colaterales tanto energéticos, como económicos o medioambientales, entre otros. Por ello, en esta disertación se han estudiado las principales limitaciones inherentes al enfoque tradicional sustituyéndolo por un nuevo paradigma de optimización multiobjetivo que tiene en cuenta los efectos colaterales y costes incurridos, en pro de construir soluciones software e infraestructuras de cómputo sostenibles. En concreto, se puso el foco sobre dos aspectos de gran impacto: la eficiencia energética en los grandes sistemas de cómputo y la eficiencia computacional de los algoritmos y técnicas de aprendizaje automático.

El primer aspecto fue abordado mediante el desarrollo de mecanismos de adaptación dinámica para los clústeres de Computación de Alto Rendimiento, optimizando conjuntamente el rendimiento computacional y los efectos derivados de los elevados consumos de energía. En concreto, se diseñó un mecanismo reactivo basado en un Sistema Borroso Genético Híbrido aprendido por medio de Algoritmos Evolutivos Multiobjetivo. Validado experimentalmente sobre un escenario real, este mecanismo mejoró notablemente los ahorros de energía alcanzados con los Sistemas Basados en el Conocimiento propuestos por otros autores, a la vez que se alinea de forma precisa con los criterios definidos para la operación del clúster. Este mecanismo se implementó por medio de una herramienta software registrada y distribuida libremente, para permitir un uso productivo del mismo. También se trabajó sobre la mejora de la eficiencia ecológica, poniendo el foco sobre una optimización del rendimiento y del impacto medioambiental del ciclo de vida de los elementos del clúster. Posteriormente, para mejorar los resultados en escenarios donde

la carga de trabajo fluctúa con una estacionalidad previsible, se diseñó un mecanismo proactivo que reajusta el clúster optimizando los recursos en base a una predicción de la carga de trabajo futura, y de acuerdo con una función de utilidad que modela los criterios de operación establecidos. Este mecanismo proactivo también fue objeto de dos contratos de investigación y de transferencia de conocimiento.

El segundo aspecto se trató mediante el diseño de algoritmos de aprendizaje automático capaces de optimizar conjuntamente precisión y costes asociados con la clasificación de las instancias. En concreto, se diseñó un clasificador multietapa sensible al coste, basado en Reglas Borrosas y aprendido por medio de Programación por Recocido Simulado Multiobjetivo. Ese nuevo clasificador se validó experimentalmente en dispositivos portátiles, donde se pudo cuantificar el efecto positivo de su eficiencia computacional sobre la duración de las baterías, frente a otras alternativas de clasificadores disponibles en la literatura de aprendizaje automático.

Abstract

Technological advances have led to the development of sophisticated computing devices, whose processing capabilities, degree of specialisation, and low acquisition costs, fostered their irruption into the daily routine as well as being key in solving fundamental problems in science or engineering, providing general-purpose IT services and enabling knowledge extraction from raw data, thus becoming a pillar of modern society. The evolution of hardware and software has focused exclusively in the pursuit of raw performance, what is inherently unsustainable due to its collateral effects whether these are power, economical o environmental-related. Because of this, the goal of this dissertation is to tackle the main limitations implicit in the traditional development approach by superseding it with a new paradigm of multiobjective optimization aware of all collateral effects and costs involved, in the pursuit of building sustainable software solutions and computing infrastructures. Specifically, two problems were researched: the energy efficiency of large computing systems and the computational efficiency of machine learning techniques and algorithms.

The first problem was addressed through the development of mechanisms for the dynamic allocation of computing resources in High Performance Clusters, jointly optimising computational performance and power consumption. Precisely, a new reactive decision-making mechanism based on Hybrid Genetic Fuzzy Systems and learned by means of Multiobjective Evolutionary Algorithms was designed to improve energy efficiency in real-word clusters. Experimental validation showed that this new mechanism archived greater power savings than those proposed in the literature, while complying with a set of subject preferences defined for the cluster operation in terms of service quality and reliability. This new mechanism was implemented in a fully-functional, registered and publicly accessible software tool enabling its productive use. Then, efforts were focused on improving ecological efficiency by balancing performance and the environmental impact related to the life cycle of the cluster's computing and support equipment. Lastly, a proactive mechanism was designed to improve energy efficiency in scenarios where the workload features foreseeable stationary fluctuations.

This mechanism optimises resource allocation over a temporal horizon, having the future workload forecast by a model of the cluster environment, and assessing every potential decision with a utility function leaned through multiobjective optimization, and tasked with modelling the subjective preferences regarding cluster operation. This proactive mechanism was also the object two knowledge transfer contracts.

The second problem was solved by designing machine learning algorithms capable of jointly optimising accuracy and classification-related costs. Specifically, a cost-conscious Multistage Fuzzy Rule-Based Classifier leaned by means of Multiobjective Simulated Annealing Programming was designed to achieve computational and cost-efficient classification. Experimentation done with battery-powered devices showed how this new classifier achieved better results in terms of classification performance and battery life than the alternatives proposed by other authors.

Índice

Índice de Figuras

Índice de Tablas

Lista de Abreviaturas

ACPI Interfaz Avanzada de Configuración y Energía (*Advanced Configuration and Power Interface*)

ANNs Redes Neuronales Artificiales (*Artificial Neural Networks*)

BSD Distribución de Software de Berkeley (*Berkeley Software Distribution*)

CFS *Correlation-based Feature Selection*

CMS Clúster de Modelización Científica de la Universidad de Oviedo

DVFS Escalado Dinámico del Voltaje y la Frecuencia (*Dynamic Voltage and Frequency Scaling*)

FLOP Operación en Coma Flotante (*Floating-point Operation*)

FLOPS Operaciones en Coma Flotante por Segundo (*Floating-point Operations Per Second*)

FRBC Clasificador Basado en Reglas Borrosas (*Fuzzy Rule-Based Classifier*)

GBML Aprendizaje Automático Basado en Genéticos (*Genetics-based Machine Learning*)

HARs Sistemas de Reconocimiento de la Actividad Humana (*Human Activity Recognition*)

HGFS Sistema Borroso Genético Híbrido (*Hybrid Genetic Fuzzy System*)

HPC Computación de Alto Rendimiento (*High Performance Computing*)

IRPCG Grupo de Recuperación de Información y Computación Paralela (*Information Retrieval and Parallel Computing Group*)

KBS Sistemas Basados en el Conocimiento (*Knowledge-Based System*)

MFCCs *Mel Frequency Cepstral Coefficients*

MFRBC Clasificador Multietapa Basado en Reglas Borrosas (*Multistage Fuzzy Rule-Based Classifier*)

MI Información Mutua (*Mutual Information*)

MOEAs Algoritmos Evolutivos Multiobjetivo (*Multiobjective Evolutionary Algorithms*)

MOSA-P Programación por Recocido Simulado Multiobjetivo (*Multiobjetive Simulated Annealing Programming*)

mRMR *minimum Redundancy Maximum Relevance*

NSGA-II *Non-dominated Sorting Genetic Algorithm-II*

OGE *Oracle Grid Engine/Open Grid Engine*

PBS *Portable Batch System*

QoS Calidad de Servicio (*Quality of Service*)

RMS Sistema de Gestión de Recursos (*Resource Management System*)

SEC Reconocimiento y Clasificación de Entornos Sonoros (*Sound Environment Classification*)

SGE *Sun Grid Engine/Son of Grid Engine*

SVMs Máquinas de Vectores de Soporte (*Support Vector Machines*)

TI Tecnologías de la Información

TORQUE *Terascale Open-source Resource and QUEue Manager*

TSK Takagi-Sugeno-Kang

Parte I: Memoria de la Tesis

1
Introducción

El nivel tecnológico actual ha permitido el desarrollo de sofisticados dispositivos computacionales, cuyas elevadas capacidades de cómputo, especialización y bajos precios de adquisición han fomentado su irrupción en un amplio abanico de escenarios, alcanzado de esta forma un extraordinario nivel de implantación en la sociedad. La adopción de estos dispositivos en la rutina diaria mediante el empleo de *smartphones*, *wearables* o dispositivos médicos, así como en los entornos académicos e industriales para la resolución de problemas fundamentales de las ciencias y la ingeniería, la provisión de servicios de Tecnologías de la Información (TI), o la extracción de conocimiento mediante la explotación de las fuentes de datos disponibles, ha generado una extrema dependencia de los mismos, constituyendo de esta forma uno de los pilares de la sociedad moderna.

Tradicionalmente, la evolución tecnológica de estos dispositivos se ha vinculado directamente con el rendimiento computacional, a menudo medido en términos de velocidad de procesamiento[1] o *throughput*[2] y con independencia de los costes energéticos en los que se incurre. Un enfoque basado únicamente en el rendimiento absoluto es inexorablemente insostenible debido a que conduce a unos consumos energéticos elevados, con el consecuente aumento de los costes de operación, huellas de carbono, concentración de calor y reducción de la fiabilidad, así como restricciones notables en la autonomía de los dispositivos ligados a fuentes de energía autónomas. No obstante, las limitaciones de este paradigma de optimización monoobjetivo no se circunscriben únicamente al hardware de los dispositivos heterogéneos de cómputo, sino también al diseño de los algoritmos que se ejecutan sobre estos dispositivos. Un ejemplo claro de esto es el caso de los sistemas de aprendizaje automático para el procesamiento intensivo de volúmenes de datos complejos. Estas técnicas de clasificación o predicción adoptadas en campos como la astronomía, biología, climatología, medicina, finanzas o economía, se desarrollan y evolucionan exclusivamente en base a su precisión (rendimiento de clasificación o de predicción), lo que tiende a limitar su escalabilidad cuando se abordan grandes conjuntos de datos por su reducida eficiencia computacional [126], o su aplicabilidad cuando se implementan en dispositivos embebidos de baja potencia.

Por estos motivos, las investigaciones basadas en el enfoque tradicional de obtener soluciones óptimas sólo en términos de rendimiento, sin tener en cuenta los efectos colaterales que ello ocasiona, han ido perdiendo terreno en favor de nuevos enfoques multiobjetivo, donde la eficiencia energética y computacional es uno de gran importancia. En concreto, las investigaciones descritas en esta disertación se enmarcan dentro de un paradigma de optimización conjunta del rendimiento computacional o de clasificación y de la eficiencia energética, con el objetivo de construir soluciones software e infraestructuras de cómputo sostenibles, abordando para ello importantes restricciones como son los costes de operación, la fiabilidad, la durabilidad o autonomía de las baterías, entre otros.

[1] La velocidad de procesamiento mide el tiempo de ejecución de un programa concreto. Se utiliza para evaluar la capacidad de un sistema para el procesamiento de tareas individuales [89, 108]

[2] La tasa de ejecución o *throughput* es una medida de la cantidad de tareas (transacciones, peticiones, trabajos, etc.) procesadas por unidad de tiempo [59, 89, 108]. Un ejemplo de este tipo de métricas es el número de Operaciones en Coma Flotante por Segundo (FLOPS, *Floating-point Operations Per Second*).

En esta disertación se explorarán dos aspectos en los que fundamentalmente se materializan las limitaciones del paradigma tradicional. El primero es la eficiencia energética de las grandes infraestructuras de cómputo, donde se desarrollarán soluciones para optimizar el equilibrio entre el rendimiento computacional absoluto y los efectos derivados de los grandes consumos de energía eléctrica y del impacto medioambiental. El segundo es la eficiencia computacional de los algoritmos y técnicas de aprendizaje automático, donde se optimizará el equilibrio entre la precisión de clasificación y de predicción, en conjunto con el efecto que esto tiene sobre el hardware subyacente en el que se ejecutan, así como de su escalabilidad a grandes conjuntos de datos. En concreto, se abordará el caso concreto de la implementación de algoritmos de clasificación sobre dispositivos médicos portátiles de baja potencia y autonomía, limitada ésta por la capacidad de sus baterías.

1.1. Optimización de los centros de computación basada en la eficiencia energética

Los centros de procesamiento de datos y de supercomputación son un elemento esencial de las sociedades modernas dado que sustentan la mayoría de servicios de TI ofrecidos a empresas y ciudadanos. Gracias a la consolidación y centralización de los procesadores y redes de comunicaciones de alto rendimiento, proporcionan las infraestructuras físicas y lógicas clave para la provisión de servidores web y de aplicaciones, plataformas de comercio electrónico, bases de datos y sistemas corporativos, almacenamiento, sistemas de explotación de datos, o los recursos de cómputo de altas prestaciones utilizados para abordar los *Grand Challenges*[3] de la ciencias y la ingeniería.

La versatilidad de estas instalaciones de cómputo, combinadas con la siempre creciente demanda de servicios de TI y el sustancial consumo de energía asociado a las mismas, hace que se hayan convertido en uno de los consumidores de electricidad con mayor tasa

[3] Los *Grand Challenges* son problemas fundamentales de la ciencia y la ingeniería con amplias aplicaciones, y cuya solución sólo es posible con el empleo de recursos de computación de altas prestaciones. Ejemplo de estos problemas son el desarrollo de nuevos materiales, la predicción del cambio climático, el diseño y fabricación de semiconductores, el diseño de medicamentos, etc. [96]

de crecimiento en los países desarrollados [52]. De hecho, el consumo de electricidad en Estados Unidos aumentó de 61.000 millones de kilovatios-hora (kWh) en 2006 a 91.000 millones de kWh en 2013, y se estima que alcance los 140.000 millones de kWh en 2020 [52, 118]. Sin embargo, se debe destacar que esta gran demanda de energía no sólo tiene un impacto económico significativo para los proveedores de servicios que operan estas instalaciones [55, 114], sino también un gran impacto medioambiental con una huella de carbono equivalente a la industria de la aviación [61], y que se espera que alcance las 340 millones de toneladas métricas de CO_2 en 2020 [58]. Por estos motivos hay una necesidad imperativa de mejorar la eficiencia energética de estas instalaciones para mejorar su sostenibilidad, reduciendo su impacto medioambiental así como los costes de operación, y mejorar la fiabilidad de sus componentes hardware.

En la última década se han llevado a cabo numerosas investigaciones para la mejora de la eficiencia de computación, en lo que a menudo se denomina *green computing*, siguiendo múltiples enfoques que pueden ser clasificados taxonómicamente en dos categorías: estáticos y dinámicos [119]. Los enfoques estáticos abordan el problema mediante el desarrollo de nuevo hardware con menor consumo energético, bien a través de CPUs de bajo consumo como el IBM PowerPC A2 del IBM BlueGene/Q [67, 72], o a través de dispositivos de cómputo diseñados para optimizar FLOPS/vatio como los coprocesadores Intel Xeon Phi o las GPUs. Por otro lado, los enfoques dinámicos se centran en reajustar los recursos de cómputo disponibles para adecuarlos a la carga de trabajo existente en cada momento, reduciendo la velocidad de procesamiento o apagando los recursos no utilizados para ahorrar energía. Ejemplo de esto es la técnica de Escalado Dinámico del Voltaje y la Frecuencia (DVFS, *Dynamic Voltage and Frequency Scaling*) consistente en reducir el voltaje y frecuencia de las CPUs cuando están infrautilizadas [32, 33, 60, 62, 68–70, 90], los planificadores de trabajos (*job schedulers*) que reducen los consumos energéticos asociados a las comunicaciones entre nodos o centros de cómputo en los *grids*[4] [135, 136], o los métodos que tienen en cuenta

[4] Los *grids* son un modelo de computación distribuida basado en recursos computacionales geográficamente dispersos, pertenecientes a dominios administrativos diferentes, y que se utilizan conjuntamente a través de una capa software que virtualiza sus recursos para presentarlos de forma transparente a los usuarios [49].

la eficiencia de refrigeración de las diferentes zonas de los centros de cómputo para reducir los costes de refrigeración [25, 111].

En esta disertación se pondrá el foco directamente sobre la adaptación dinámica de los recursos de los clústeres de computadores, elementos que constituyen las infraestructuras físicas predominantes en los centros de procesamiento de datos y de supercomputación modernos [8, 9]. Estas técnicas consisten fundamentalmente en reconfiguración de los nodos de cómputo del clúster (modificación del estado global ACPI *Gx* del dispositivo[5]), adaptando los recursos disponibles con los demandados en cada momento por la carga de trabajo recibida, ahorrando energía cuando el clúster está infrautilizado. Esta técnica tiene amplia aceptación y se ha empleado tanto en clústeres de balanceo de carga (*Load-Balancing clusters*) [27, 50, 56, 84, 102] como en clústeres de Computación de Alto Rendimiento (HPC, *High Performance Computing*) [22, 54, 81, 130], y en los hipervisores comerciales VMware vSphere y Citrix XenServer [12, 15, 35, 121].

En estos trabajos se propone como norma general, y especialmente en el caso de los clústeres HPC, una serie de mecanismos de toma de decisiones para ajustar el clúster mediante Sistemas Basados en el Conocimiento (KBS, *Knowledge-Based System*) que modelan conocimiento experto por medio de un conjunto de reglas, y que determinan el número de nodos que debe estar encendido en cada momento. No obstante, los trabajos realizados hasta el momento presentan dos limitaciones fundamentales:

1. Los Sistemas Basados en el Conocimiento propuestos hasta el momento están definidos de forma independiente a las características de clúster o al escenario de carga. Es decir, están basado en un conjunto de reglas fijas dictadas por un experto humano, y cuya capacidad de adaptación a un escenario u otro se

[5] La especificación Interfaz Avanzada de Configuración y Energía (ACPI, *Advanced Configuration and Power Interface*) [113] define cuatro estados globales de energía para un sistema compatible: *G0*, *G1*, *G2* y *G3*. En el estado *G0 Working* el dispositivo está operativo y ejecutando instrucciones. En el estado *G1 Sleeping* el dispositivo consume una cantidad reducida de energía, aparentando estar apagado para el usuario y sin ejecutar hilos en el modo usuario, pero permite una vuelta al estado *G0* con una baja latencia y sin necesidad de reiniciar el Sistema Operativo. En el nivel *G2 Soft Off* el dispositivo consume una mínima cantidad de energía, no ejecuta código en modo usuario ni en modo sistema, y requiere una mayor latencia para volver al estado *G0*, así como un reinicio del Sistema Operativo. En el nivel *G3 Mechanical Off* no hay corriente eléctrica a través de los circuitos, y se puede desensamblar el equipamiento sin dañar el hardware.

limita al ajuste de los parámetros de funcionamiento. Esta particularidad reduce notablemente la capacidad de adaptación y optimalidad del ahorro energético logrado en los escenarios del mundo real a los que se presenten estos sistemas. Todavía no se ha explorado la posibilidad de aprender sistemas cuya base de reglas dependa del escenario al que se deban adaptar, así como al equilibrio deseado entre el rendimiento del clúster y los costes asociados al mismo, incluyendo el consumo energético o el efecto sobre la fiabilidad del hardware que tienen las reconfiguraciones (encendidos y apagado) de los nodos de cómputo.

2. Los mecanismos de adaptación del clúster propuestos siguen una estrategia puramente reactiva, en la que se toman decisiones en función del estado actual del clúster. Sin embargo, no se ha explorado la posibilidad de tomar decisiones en base a un modelo de predicción de la carga de trabajo futura del clúster, lo que permitiría lograr un mayor grado de adaptación y mejores resultados para aquellos patrones cambiantes pero predecibles de las cargas de trabajo.

1.2. Optimización de las técnicas de aprendizaje automático basada en la eficiencia computacional

El rápido desarrollo de las redes de comunicaciones y de las capacidades de recopilación, generación y almacenamiento de los datos ha conducido a un escenarios donde grandes volúmenes de datos se producen cada día desde múltiples fuentes autónomas [34, 126]. A medida que estos grandes y complejos volúmenes de datos se extienden a todos los dominios de la ciencia y la ingeniería, ha surgido una creciente demanda de procesos eficientes de descubrimiento de conocimiento capaces de extraer información útil en tiempo real a partir de datos en bruto por medio de técnicas de clasificación y de predicción [126]. De hecho, los sistemas inteligentes de aprendizaje automático ya han sido adoptados en numerosos campos con un uso intensivo de datos [18]. Sin embargo, el diseño de la mayoría de estos sistemas está basado exclusivamente en el rendimiento de los mismos, medido éste como la precisión de clasificación o de predicción. Esta estrategia de diseño es la responsable de limitar su escalabilidad cuando se deben abordar grandes conjuntos de datos debido a su limitada eficiencia computacional [18, 131].

Este problema se aprecia claramente cuando se trabaja con conjuntos de datos de alta dimensionalidad, consistentes en un gran número de instancias y/o atributos, como son habituales en el procesamiento de imágenes médicas, reconocimiento de texto, o datos genéticos [21, 29]. Por un lado, la presencia de una gran cantidad de atributos irrelevantes o redundantes degradan notablemente el rendimiento de los algoritmos tradicionales de aprendizaje automático [29, 77, 125]. Por otro lado, un gran volumen de instancias y/o atributos afecta negativamente el proceso de aprendizaje por el crecimiento exponencial del espacio de búsqueda, y que también es responsable del incremento de la complejidad de los clasificadores aprendidos, lo que a su vez reduce su escalabilidad y eficiencia [18, 21]. Es más, la búsqueda de técnicas de aprendizaje automático computacionalmente eficientes no se precisa sólo para abordar la alta dimensionalidad, sino que también es esencial en multitud de aplicaciones de clasificación del mundo real que deben tener en cuenta explícitamente el consumo de tiempo, memoria, almacenamiento, energía, batería y/o los costes monetarios. Estos casos son aquellos en los que hay restricciones inherentes al problema a resolver, bien sea por el empleo de dispositivos con escasos recursos de cómputo o baterías de capacidad limitada, por la necesidad de dar una respuesta en tiempos muy reducidos para la operación en tiempo-real del sistema (transacciones con tarjetas de crédito, traducción voz a voz), porque la obtención de datos es intrusiva o tiene altos costes de adquisición (diagnosis médica), o porque el tiempo de CPU es costoso y debe ser presupuestado y controlado, entre otros ejemplos [53, 71, 87, 104, 127].

Dado el marco de esta disertación, el caso de la implementación de sistemas de aprendizaje automático en dispositivos de cómputo de escala reducida es particularmente importante. Este tipo de dispositivos de baja potencia embebidos, móviles o *wearables* han sido sujeto de numerosos investigaciones dado su bajo coste, pequeño tamaño y amplia disponibilidad en el mercado. En particular, han permitido el desarrollo de un amplio abanico de aplicaciones en el campo biomédico, el de las comunicaciones, el empresarial, el de la localización de objetos o personas, o el de las aplicaciones de vigilancia [28, 85, 98, 132]. Un ejemplo de estas aplicaciones es la implementación de Sistemas de Reconocimiento de la Actividad Humanas (HARs, *Human Activity Recognition*)text en *smartphones* o *wearables* para la motorización remota de pacientes, permitiendo una supervisión continua de la salud y el bienestar de personas ancianas [24], la vigilancia de la severidad de los síntomas y de las complicaciones motoras en pacientes con la enfermedad de Parkinson [100], el control en tiempo real de las señales fisiológicas mediante electrocardiografía [101, 109], etc. Sin embargo, los desafíos fundamentales para el desarrollo de estas aplicaciones son las limitadas capacidades de cómputo y restricciones en el consumo de energía de estos dispositivos portátiles alimentados por baterías. Por ello, hay una demanda crucial de clasificadores eficientes energéticamente capaces de reducir los costes computacionales, encontrando un equilibrio optimo entre precisión y consumos de energía, para maximizar la vida útil de las baterías hasta un nivel adecuado para un uso práctico [23, 24, 26, 28, 63, 83, 85, 97–100, 105, 109].

La reducción de la complejidad computacional de los algoritmos de clasificación es un tema que se ha estudiado profundamente a lo largo de los años, y del que existe una amplia literatura al respecto. Un enfoque bien conocido es el de la selección de características o *feature selection*, que consiste en la selección de un subconjunto de los atributos que describen cada instancia de tal forma que se centra la atención del clasificador sobre los atributos más relevantes [82]. Esta técnica se suele utilizar para mejorar la precisión de los clasificadores, así como para aprender predictores más rápidos y eficientes computacionalmente [82, 125]. Aunque se han propuesto numerosos métodos para realizar la selección de características, dado el objeto y alcance de esta disertación, se pondrá especialmente el foco sobre la solución propuesta en [29], ya que los métodos *ranker* y *filter* descritos buscan explícitamente la minimización del coste de adquisición de las características.

Otro enfoque es el de las implementaciones eficientes de clasificadores empleados habitualmente como las Máquinas de Vectores de Soporte (SVMs, *Support Vector Machines*) o las Redes Neuronales Artificiales (ANNs, *Artificial Neural Networks*), con el objetivo de reducir su complejidad computacional. Ejemplos de esto se pueden encontrar con la propuesta de un SVM simplificado en [86], la aproximación de SVMss en [80], o la aproximación del kernel Gaussiano RBF propuesto en [106] para acelerar las evaluaciones de los SVMss.

Las arquitecturas multietapa en las que se combinan una serie de clasificadores independientes de creciente coste computacional en un estructura jerárquica, tales como una cascada o un árbol, pueden mejorar también la eficiencia de los clasificadores tradicionales. La idea es mejorar progresivamente la precisión del clasificador con cada nueva etapa, al precio de un mayor coste computacional [107]. Esta técnica permite que el proceso de clasificación se detenga cuando se haya alcanzado el nivel de certeza deseado, reduciendo de esta forma los costes asociados a la clasificación y extracción de características en el caso de instancias simples, y requiriendo sólo los máximos costes para instancias más complejas, con la consecuente mejora de la eficiencia global. Se pueden ver ejemplos de la aplicación de esta técnica en [104, 107, 115, 120, 128]. Otra técnica similar son los clasificadores evolutivos basados en reglas borrosas, que tienen en cuenta el coste computacional al basarse en cálculos recursivos que permiten realizar conjuntamente el aprendizaje y la inferencia, en detrimento de una menor precisión en las primeras instancias [47, 48, 79, 103].

Los ensembles de clasificadores construidos mediante *bagging*, *boosting* o *stacking* también se pueden simplificar para mejorar su eficiencia. Los ensembles excesivamente grandes y complejos se pueden reducir a sub-ensembles más eficientes con la misma precisión que el original, pero con una fracción de los costes computacionales y de memoria [94, 133]. Se pueden ver ejemplos de esta técnica en [133], [94] y [91].

Todos los métodos descritos hasta este punto buscan mejorar la eficiencia de los clasificadores mediante la reducción de su complejidad computacional con un impacto despreciable o mínimo sobre su precisión. Sin embargo, proporcionar una solución adecuada para cada posible escenario requiere que el problema de optimización se aborde no con carácter monoobjetivo, sino multiobjetivo donde el resultado es un conjunto de equilibrios óptimos entre el rendimiento de clasificación y todos los costes asociados que deban ser tenidos en cuenta. Este planteamiento es clave para poder encontrar una solución adecuada a cualquier problema que se presente mediante el cumplimiento de todos las restricciones inherentes al mismo (tiempos de respuesta, consumos de memoria, operaciones por segundo, etc.) y la satisfacción de las preferencias subjetivas de un experto humano (como el equilibrio óptimo entre precisión y autonomía de la batería, consumos de energía, costes monetarios, etc.). Ejemplos de este enfoque multiobjetivo se pueden encontrar en la literatura sobre selección de características [129], aprendizaje de ensembles [116] y sobre el diseño de clasificadores [19, 20, 57, 64, 73, 74, 76].

A pesar de esto, y de acuerdo con el conocimiento del autor, todavía no se ha presentado un enfoque completo y suficientemente flexible para aprender clasificadores sensibles al coste y adecuados para un empleo en cualquier escenario. El motivo de esto es que ninguno de los métodos propuestos hasta ahora son capaces de encontrar el equilibrio óptimo entre la precisión o rendimiento de clasificación y todos los costes explícitos relevantes para el problema en cuestión (tanto en relación al proceso de inferencia de la clase como a la extracción de las características), tanto si estos costes son de carácter computacional, temporal, energéticos, económicos, o incluso concernientes al grado de agresividad para un paciente.

2

Objetivos

El objetivo principal de esta disertación es la definición, diseño y desarrollo de soluciones software que maximicen la sostenibilidad de las infraestructuras de cómputo subyacentes por medio de un paradigma de optimización conjunta de la Calidad de Servicio (QoS) y de los costes asociados a la misma. Para lograr este objetivo general, se han establecido los siguientes objetivos específicos:

- Estudio del arte sobre las técnicas propuestas por otros autores para le mejora de la eficiencia energética de los clústeres de computadores. Analizar los sistemas de adaptación dinámica de los nodos de los clústeres, sus mecanismos de toma de decisiones, así como las reglas y parámetros que gobiernan su funcionamiento. Identificar las limitaciones y oportunidades de mejora así como los problemas abiertos dentro de esa línea de investigación.

- Obtener cargas de trabajo reales de clústeres de alto rendimiento correspondiente a un entorno multiusuario representativo de escenarios sobre los que se ejecutan una amplia variedad de aplicaciones con características, patrones de uso y tiempos de ejecución diversos. Definir un conjunto de cargas de trabajo sintéticas con diferentes patrones de actividad y que permitan verificar de forma significativa en diferentes escenarios el comportamiento de los sistemas de adaptación dinámica propuestos.

- Desarrollar un simulador de un clúster de computación de alto rendimiento para calcular los resultados objetivos con cada solución en términos de las métricas establecidas para medir la Calidad de Servicio, consumos de energía e impacto para la fiabilidad del clúster.

- Desarrollar y validar experimentalmente mecanismos de toma de decisiones reactivos basados en Sistemas Borrosos Híbridos para la adaptación dinámica de los recursos del clúster.

- Desarrollar y validar experimentalmente un mecanismo de toma de decisiones proactivos basados en modelos de predicción de la carga de trabajo para la adaptación dinámica de los dispositivos de cómputo.

- Diseñar algoritmos de aprendizaje distal supervisado y multiobjetivo para el aprendizaje y ajuste de los mecanismos de toma de decisiones anteriores.

- Desarrollar una solución software completa y distribuible que permita la aplicación práctica de los sistemas de adaptación dinámica propuestos.

- Estudio del arte sobre las técnicas propuestas para mejorar la eficiencia computacional y/o energética de los sistemas de clasificación. Identificar problemas abiertos, nuevos desafíos y oportunidades de mejora.

- Diseñar un clasificador y un algoritmo de aprendizaje multiobjetivo que permita optimizar conjuntamente el rendimiento de clasificación junto con los costes asociados al mismo, realizando simultáneamente el aprendizaje del clasificador y la selección de características. Validación experimental en sistemas embebidos de baja potencia ligados fuentes de energía autónomas.

3

Metodología

Este capítulo resume las herramientas y conjuntos de datos utilizados durante el desarrollo de los trabajos enmarcados en esta disertación. La información detallada sobre la metodología empleada en cada uno de los estudios experimentales se puede encontrar en la documentación respectiva al artículo.

3.1. Herramienta EECluster

EECluster (*Energy-Efficient Cluster*) es una herramienta software desarrollada en el marco de esta disertación para implementar y validar experimentalmente los mecanismos de toma de decisiones, así como para realizar el aprendizaje de los mismos por medio de los algoritmos multiobjetivo. Este software se distribuye bajo la licencia BSD modificada o de tres cláusulas, se puede descargar en [4, 5], y está inscrita en el Registro General de la Propiedad Intelectual con el número de asiento registral *05/2016/172* y bajo el título *EECluster: An Energy-Efficient software tool for managing HPC Clusters.*

En concreto, la herramienta está desarrollada en Java y permite automatizar la adaptación dinámica de los nodos de cómputo de clústeres HPC que empleen gestores de recursos OGE/SGE[1] y PBS/TORQUE[2]. La Figura 3.1 representa gráficamente la arquitectura de la herramienta, la cual se compone de los siguientes módulos:

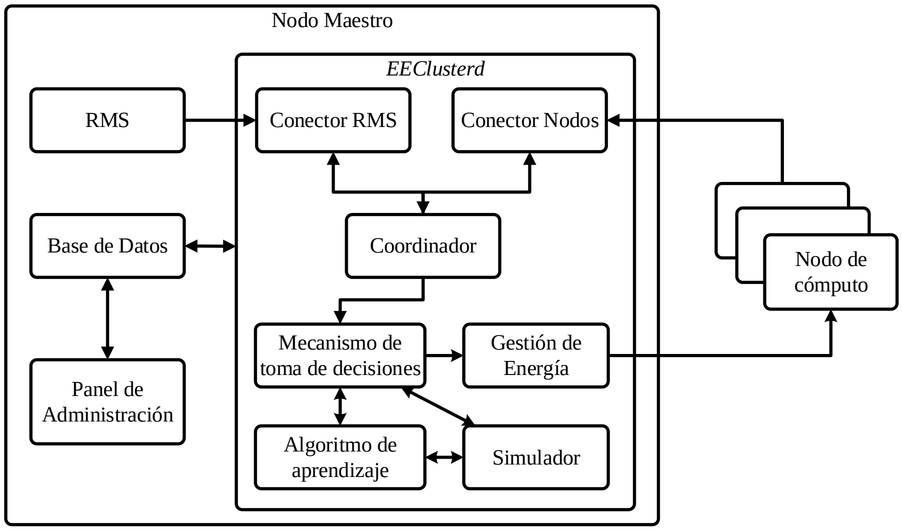


Figura 3.1: Componentes de la herramienta EECluster.

- Servicio *EEClusterd*: servicio ejecutado en el nodo maestro del clúster encargado de la reconfiguración automática del clúster y del aprendizaje del mecanismo de toma de decisiones. Está formado por los siguientes módulos:
  - *Conector RMS*: módulo que extrae información del Sistema de Gestión de Recursos (RMS, *Resource Management System*) del clúster sobre los *slots*, colas, trabajos, históricos, etc.
  - *Conector con los Nodos*: módulo que extrae información sobre hardware de

[1] *Oracle Grid Engine/Open Grid Engine* (OGE), anteriormente conocido como *Sun Grid Engine/Son of Grid Engine* (SGE), es un sistema de gestión de recursos desarrollado inicialmente por Sun Microsystems y más tarde por Oracle. Actualmente cuenta con versiones tanto comerciales como de código abierto [10]. Este sistema ha tenido una gran expansión en la última década gracias a su facilidad de despliegue y uso [31].

[2] *Terascale Open-source Resource and QUEue Manager* (TORQUE) es un sistema de gestión de recursos de código abierto desarrollado por Adaptive Computing Enterprises como una extensión al sistema *Portable Batch System* (PBS) desarrollado inicialmente por la NASA [11, 13]. Actualmente es uno de los sistemas *batch* más ampliamente utilizados en infraestructuras de computación de rendimiento de tamaño pequeño y mediano [31].

los nodos de cómputo del clúster.

- *Coordinador*: módulo que ejecuta periódicamente la rutina de monitorización, toma de decisiones y reconfiguración del clúster para adaptar sus recursos de cómputo a las necesidades de la carga de trabajo.
- *Mecanismo de toma de decisiones*: módulo que implementa el mecanismo reactivo o proactivo que determina el número de elementos de cómputo con los que debe contar el clúster en cada momento.
- *Gestión de Energía*: módulo que envía las órdenes de encendido/apagado a los nodos de cómputo.
- *Algoritmo de aprendizaje*: módulo que implementa el algoritmo de aprendizaje y que produce las correspondientes aproximaciones al frente óptimo de Pareto[3]. Este módulo depende a su vez de la implementación de los algoritmos multiobjetivo disponibles en la biblioteca MOEA Framework (ver Sección 3.2.2).
- *Simulador*: módulo que, dado una carga de trabajo y un mecanismo de toma de decisiones, evalúa el comportamiento del clúster y calcula el valor de las métricas de Calidad de Servicio, ahorros de energía y números de reconfiguraciones. Este módulo es utilizado por parte del algoritmo de aprendizaje para evaluar cada

- *Panel de administración*: aplicación web que permite acceder a la información actual e histórica del clúster, visualizar el estado del clúster y de nodos, y configurar los parámetros de funcionamiento de la herramienta.

[3] En numerosos problemas del mundo real se requiere optimizar múltiples objetivos que están en conflicto entre sí, y para los cuales la mejora del resultado en uno de los objetivos afecta empeorando el de los demás. Estos problemas no tienen una única solución óptima, sino un conjunto de soluciones alternativas denominado soluciones óptimas de Pareto (*Pareto-optimal solutions*) o frente óptimo de Pareto. Estas soluciones son óptimas en el sentido de que no existe ninguna otra en el espacio de búsqueda que sea superior simultáneamente en todos los objetivos. Sin embargo, encontrar el verdadero conjunto de soluciones óptimas de Pareto es generalmente inviable dado que la complejidad subyacente del problema dificulta la aplicación de métodos exactos. Por ello, el frente óptimo de Pareto se suele aproximar por medio algoritmos evolutivos [30, 112, 122, 134].

- *Bases de Datos*: almacena la información utilizada por los componentes de la herramienta EECluster sobre la configuración de la misma y sobre el clúster.

Se puede encontrar información detallada sobre esta herramienta en la Referencias [4, 5, 39].

3.2. Bibliotecas software de terceros

3.2.1. Weka

Weka [65] es la plataforma software de aprendizaje automático y minería de datos cuyas implementaciones de los clasificadores SVMs, C4.5, PART y k-nn se utilizaron en los experimentos. También se utilizó para experimentar con las variantes sensibles al coste de los algoritmos de selección de características *Correlation-based Feature Selection* (CFS) y *minimum Redundancy Maximum Relevance* (mRMR) propuestos en [29].

3.2.2. MOEA Framework

MOEA Framework [7] es una biblioteca de software Java de código abierto para el desarrollo y experimentación con Algoritmos Evolutivos Multiobjetivo (MOEAs, *Multiobjective Evolutionary Algorithms*) y otros algoritmos de optimización mono y multiobjetivo de propósito general. Incluye implementaciones de los algoritmos NSGA-II, NSGA-III, ϵ-MOEA, GDE3, PAES, PESA2, SPEA2, IBEA, SMS-EMOA, SMPSO, OMOPSO, CMA-ES, y MOEA/D. También proporciona herramientas para el diseño, implementación, ejecución y prueba de nuevos algoritmos de optimización. En el marco de esta disertación se utilizaron implementaciones de MOEAs de esta biblioteca para construir el algoritmo de aprendizaje distal de la herramienta EECluster, así como para realizar una selección multiobjetivo de características y clasificadores en el problema de Reconocimiento y Clasificación de Entornos Sonoros.

3.3. Clústeres de Computación de Alto Rendimiento

En el marco de esta disertación se trabajó con tres clústeres HPC de diferentes características y finalidades. A continuación se detalla cada uno de estos clústeres.

3.3.1. Clúster de Modelización Científica

El Clúster de Modelización Científica de la Universidad de Oviedo (CMS) es un clúster profesional y de investigación empleado para la realización de cálculo intensivo paralelo y análisis de datos, con el objetivo de potenciar a nivel universitario el cálculo y la modelización computacional, el procesado y almacenamiento de datos científicos y el uso compartido de aplicaciones informáticas científicas. Fue implantado inicialmente en 2006 dentro del proyecto *Clúster Universitario de Modelización Científica* (referencia FEDER-05-UNOV05-23-009) cofinanciado por el Ministerio de Educación y Ciencia y las ayudas FEDER a la realización de proyectos de infraestructura científica, y ampliado en 2008 y 2012. Está diseñado para el procesamiento de cálculos paralelos en memoria compartida, cálculos paralelos con comunicación entre servidores mediante librerías MPI y red de interconexión Gigabit, cálculos distribuidos entre servidores, etc. Este clúster está formado por tres subclústeres de cómputo independientes con cinco colas transversales que usan PBS como sistema de gestión de recursos y dos cabinas de almacenamiento centralizado (HP MSA1500 y HP EVA 4400). El detalle de los subclústeres y de sus nodos de cómputo se puede observar en la Tabla 3.1. Se puede encontrar más información sobre el CMS en su sitio web [3].

Subclúster	*cmi*	*cmq*	*cmd*
Nodos de cálculo	32	10	50
Modelo de servidor	HP BL465	HP DL585	HP DL140 G2
Número de procesadores	2	4	2
Procesador	AMD Opteron 2356 Quad Core 2.3 GHz	AMD Opteron 875 Dual Core 2.2 GHz	Intel Xeon 3.60 GHz
Memoria RAM	32 GB	16 GB	24 GB
Red de cálculo	InfiniBand	Gigabit Ethernet	Gigabit Ethernet
Colas de acceso	*cola64*, *cola32*, *cola8*	*cola8_cmq*	*cola2*

Tabla 3.1: Detalle de los nodos de cómputo del clúster CMS.

Este clúster se utilizó en el marco de esta disertación para la validación de los mecanismos de adaptación dinámica propuestos sobre un entorno real y de carácter profesional.

3.3.2. Clúster de investigación del grupo IRPCG

El segundo es un clúster de investigación utilizado por el Grupo de Recuperación de Información y Computación Paralela (IRPCG, *Information Retrieval and Parallel*

Computing Group) de la Universidad de Oviedo [6]. Está formado por 5 nodos de cómputo organizados en dos colas, y emplea OGE/SGE como sistema de gestión de recursos. Los nodos incluyen dos servidores Dell PowerEdge 1950 (1x CPU Intel Xeon E5420 @ 2.5 GHz y 16 GB RAM), un servidor Dell PowerEdge 2950 (1x CPU Intel Xeon E5420 @ 2.5 GHz y 16 GB RAM), un servidor ASUS (2x CPU Intel Xeon E5-2650 @ 2.0 GHz, 64 GB RAM, 1x GPU NVIDIA Tesla K40m, 1x GPU NVIDIA Tesla K20m) y un servidor Supermicro (2x CPUs Intel Xeon E5-2603 v3 @ 1.6 GHz, 32 GB RAM, 1x Intel Xeon Phi 5110P, 2x Intel Xeon Phi 31S1P). Este clúster se utilizó para la ejecución de los experimentos en los que se evaluó el comportamiento de los diferentes sistemas y algoritmos propuestos en esta disertación, así como para el uso práctico de la herramienta EECluster.

3.3.3. Clúster académico

El tercer clúster tiene un uso académico y es utilizado para la formación de alumnos de la Universidad de Oviedo en el campo de la computación de alto rendimiento. Este clúster usa OGE/SGE como sistema de gestión de recursos, y está formado por 34 nodos cómputo. Los nodos de cómputo incluyen tanto PCs con CPUs Intel Core i3-2100 @ 3.10 GHz y 4 GB de RAM, como PCs con CPUs Intel Core i7 930 @ 2.80 GHz, 12 GB de RAM y GPUs NVIDIA GeForce GTX 480. Sobre este clúster se desplegó de forma productiva la herramienta EECluster.

3.4. Cargas de trabajo de clústeres HPC

En los experimentos realizados para validar los mecanismos de adaptación dinámica de clústeres HPC se utilizaron un conjunto de cargas de trabajo tanto extraídas de un clúster real, como sintéticas para simular diferentes escenarios. A continuación se detallan las diferentes cargas utilizadas.

3.4.1. Clúster CMS

Las cargas de trabajo reales utilizadas en los experimentos se extrajeron del clúster CMS detallado en la Sección 3.3.1, y están formadas por un total de 2907 trabajos de

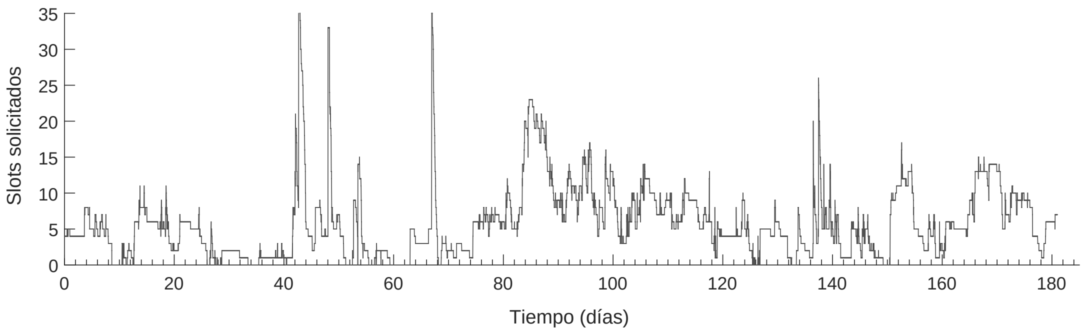


Figura 3.2: Traza de la cargas de trabajo del clúster CMS usadas en los experimentos.

la cola de acceso *cola2* a lo largo de 22 meses. La Figura 3.2 representa gráficamente la demanda de unidades de cómputo (*slots*) a lo largo del tiempo para este conjunto de datos.

3.4.2. Escenarios sintéticos

Para simular diferentes escenarios de un clúster atendiendo a diferentes patrones de las cargas de trabajo, se generaron cuatro escenarios diferentes con creciente grado de fluctuación en términos de tasas de llegadas de trabajos y una duración total de 24 meses. Las llegadas de trabajos en estos escenarios siguen un proceso de Poisson con los valores λ mostrados en la Tabla 3.2. La duración de los trabajos siguen una distribución exponencial con $\lambda = 10^{-5}$ segundos en todos los escenarios. La Figura 3.4 representa gráficamente la demanda de *slots* a lo largo del tiempo para cada escenario. Las Figuras 3.5, 3.6, 3.7 y 3.8 muestran histogramas de frecuencias de la llegada de trabajos según la hora de la semana y la semana del año, así como de sus duraciones, en los diferentes escenarios.

El Escenario 1 muestra una carga de trabajo estable y sostenida en el tiempo. El Escenario 2 añade la distinción entre horas laborales, horas no laborales y horas del fin de semana. El Escenario 3 añade una variación en las tasas de llegada de trabajo en horas laborales, y el Escenario 4 incrementa esta variación.

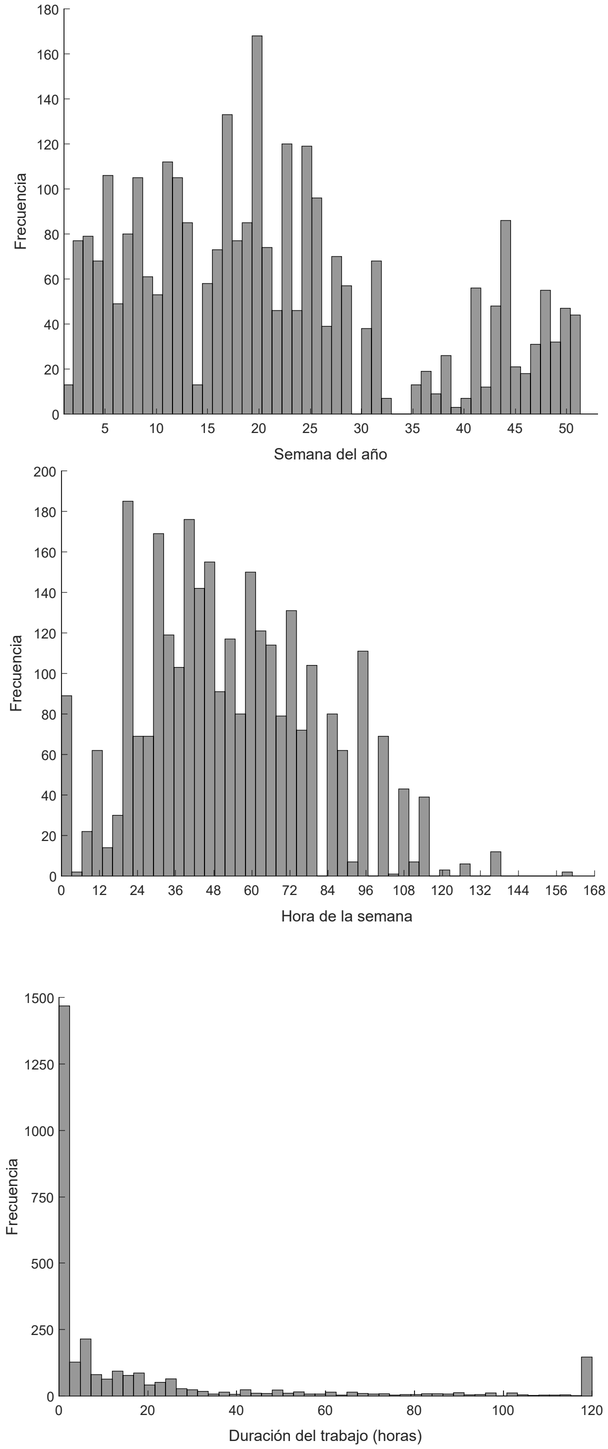


Figura 3.3: Histogramas de llegada de trabajos y duraciones en el clúster CMS.

Escenario	Día de la semana	Horas	Semana del año	λ
1	Todos	Todas	Todas	2×10^{-4} s
2	Lun - Vie	8:00 - 20:00	Todas	2×10^{-4} s
	Sab - Dom	8:00 - 20:00	Todas	2×10^{-5} s
	Lun - Dom	20:00 - 8:00	Todas	10^{-5} s
3	Mon - Fri	8:00 - 20:00	$w \,\%\, 5 = 0$	10^{-4} s
			$w \,\%\, 5 = 1$	2×10^{-4} s
			$w \,\%\, 5 = 2$	5×10^{-4} s
			$w \,\%\, 5 = 3$	5×10^{-4} s
			$w \,\%\, 5 = 4$	2×10^{-4} s
	Lun - Dom	20:00 - 8:00	Todas	2×10^{-5} s
	Lun - Vie	8:00 - 20:00	Todas	10^{-5} s
4	Mon - Fri	8:00 - 20:00	$w \,\%\, 5 = 0$	10^{-4} s
			$w \,\%\, 5 = 1$	10^{-4} s
			$w \,\%\, 5 = 2$	5×10^{-4} s
			$w \,\%\, 5 = 3$	5×10^{-4} s
			$w \,\%\, 5 = 4$	10^{-4} s
	Lun - Dom	20:00 - 8:00	Todas	2×10^{-5} s
	Lun - Vie	8:00 - 20:00	Todas	10^{-5} s

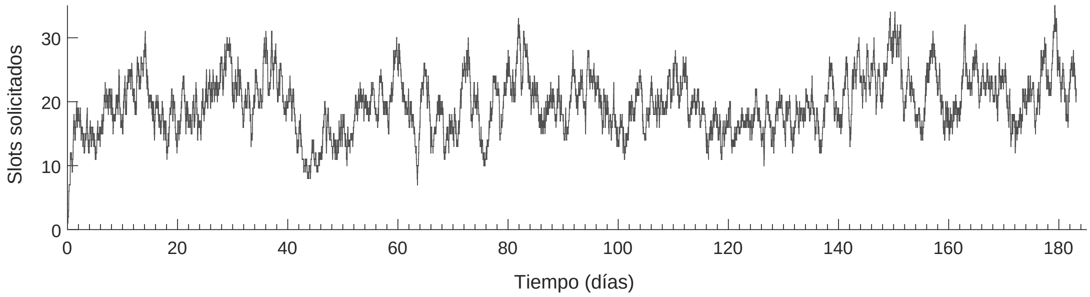
Tabla 3.2: Proceso de Poisson de las llegadas de trabajos en cada escenario.

3.5. Reconocimiento y Clasificación de Entornos Sonoros

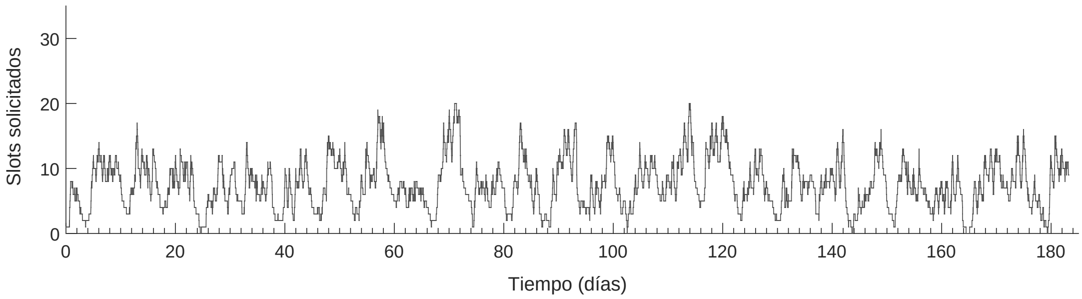
Para validar los clasificadores sensibles al coste propuestos en esta disertación se utilizó un problema real en el que los costes energéticos y computacionales tienen un gran impacto. En concreto, se abordó el Reconocimiento y Clasificación de Entornos Sonoros (SEC) que realizan los audífonos modernos para seleccionar automáticamente los parámetros de ampliación óptimos para cada entorno, mejorando de esta forma el confort del usuario [66], así como apoyar otras funcionalidades como realzar el habla, reducir el ruido ambiente [88, 93], o la detección de voz [95].

Este problema de clasificación consiste en capturar las señales de sonido, calcular los MFCCs[4], extraer las características a partir de estadísticas temporales de los MFCCs,

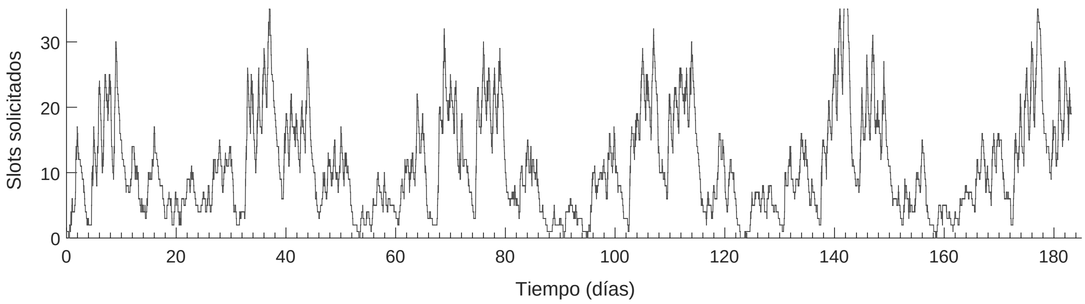
[4] Los *Mel Frequency Cepstral Coefficients* (MFCCs) son unos coeficientes que colectivamente representan el espectro de energía de un sonido basados en una aproximación de la respuesta del sistema auditivo humano [123].



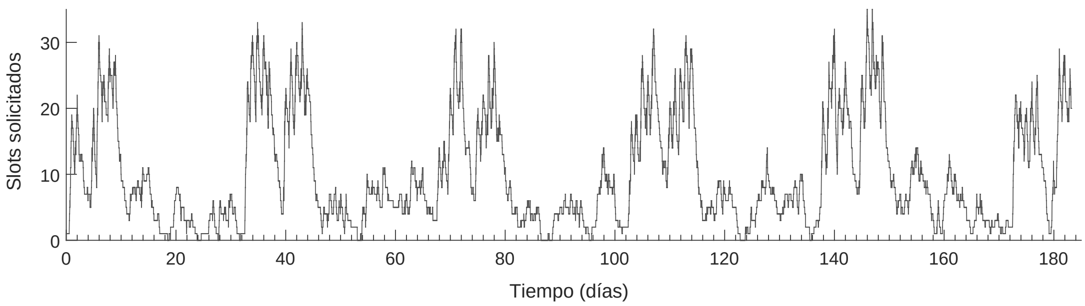
(a) Escenario 1



(b) Escenario 2



(c) Escenario 3



(d) Escenario 4

Figura 3.4: Traza de las cargas de trabajo de los diferentes escenarios.

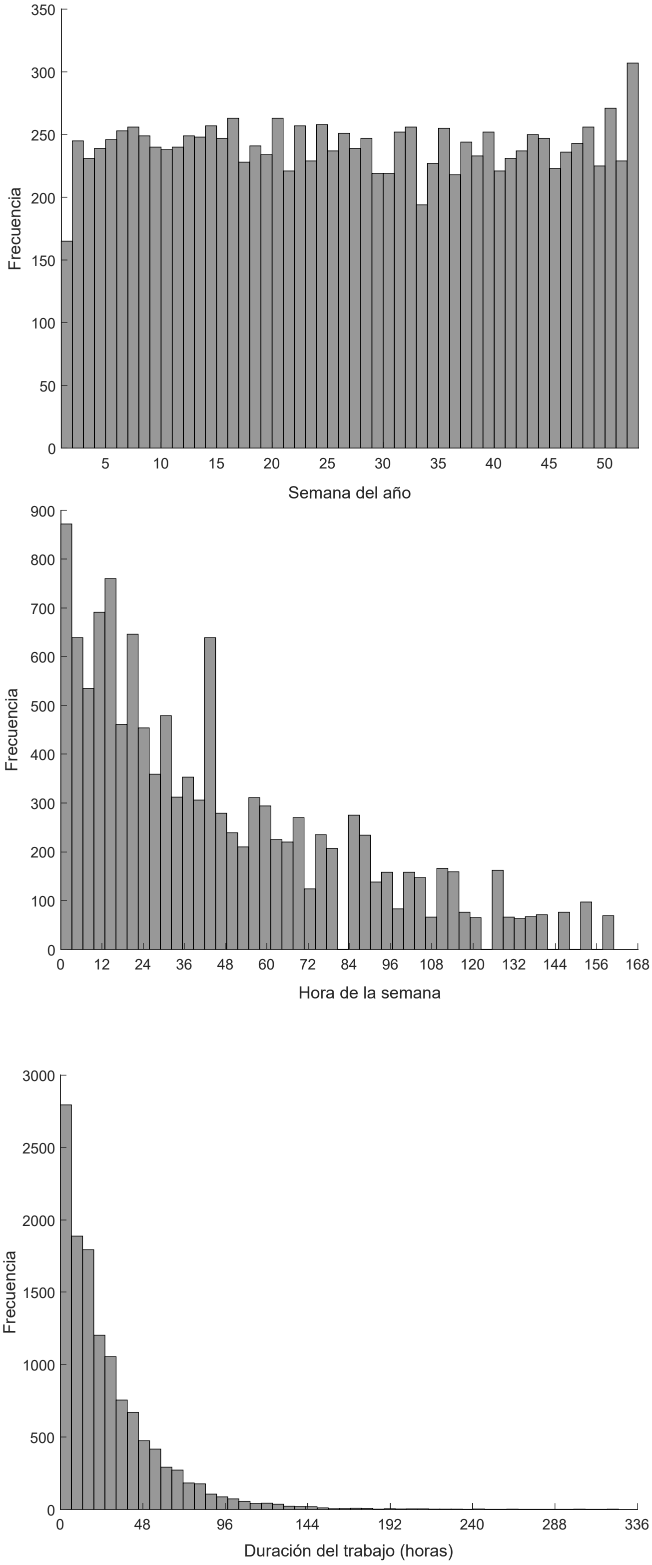


Figura 3.5: Histogramas de llegada de trabajos y duraciones en el Escenario 1.

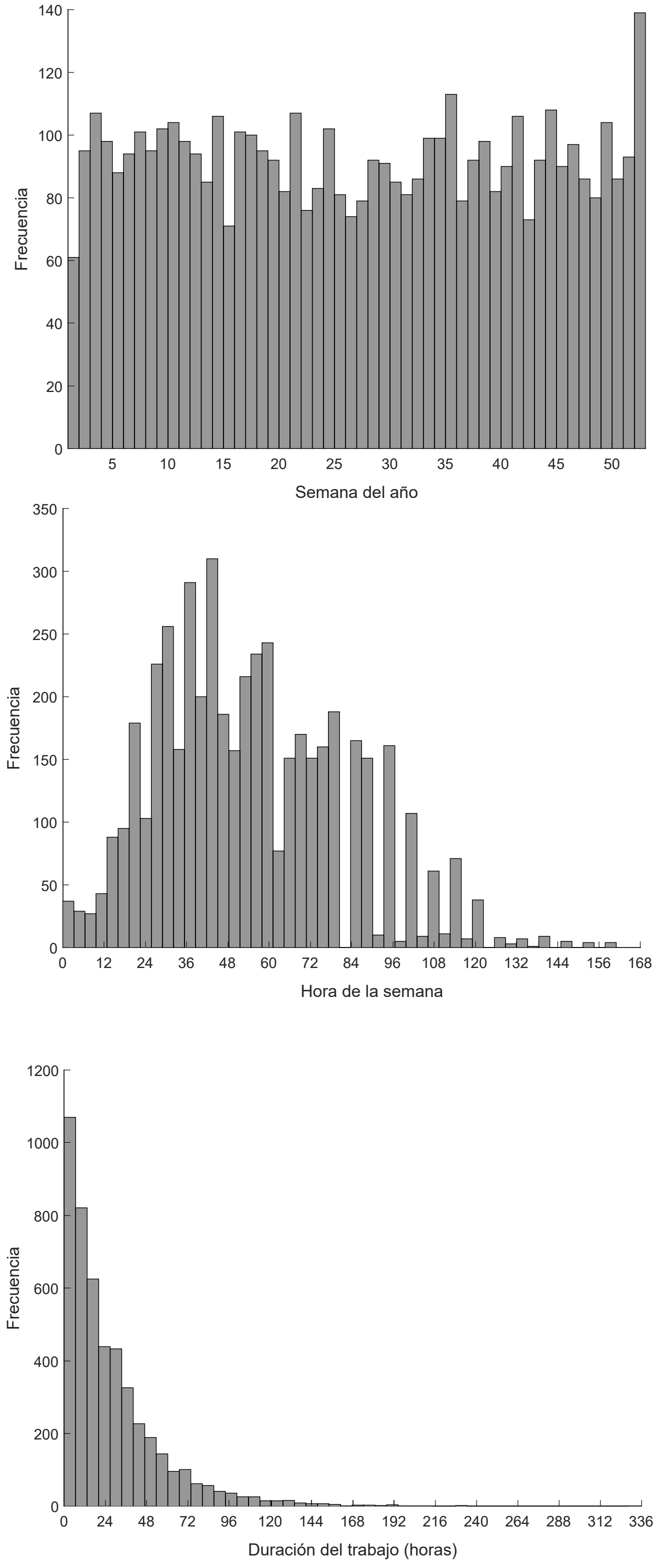


Figura 3.6: Histogramas de llegada de trabajos y duraciones en el Escenario 2.

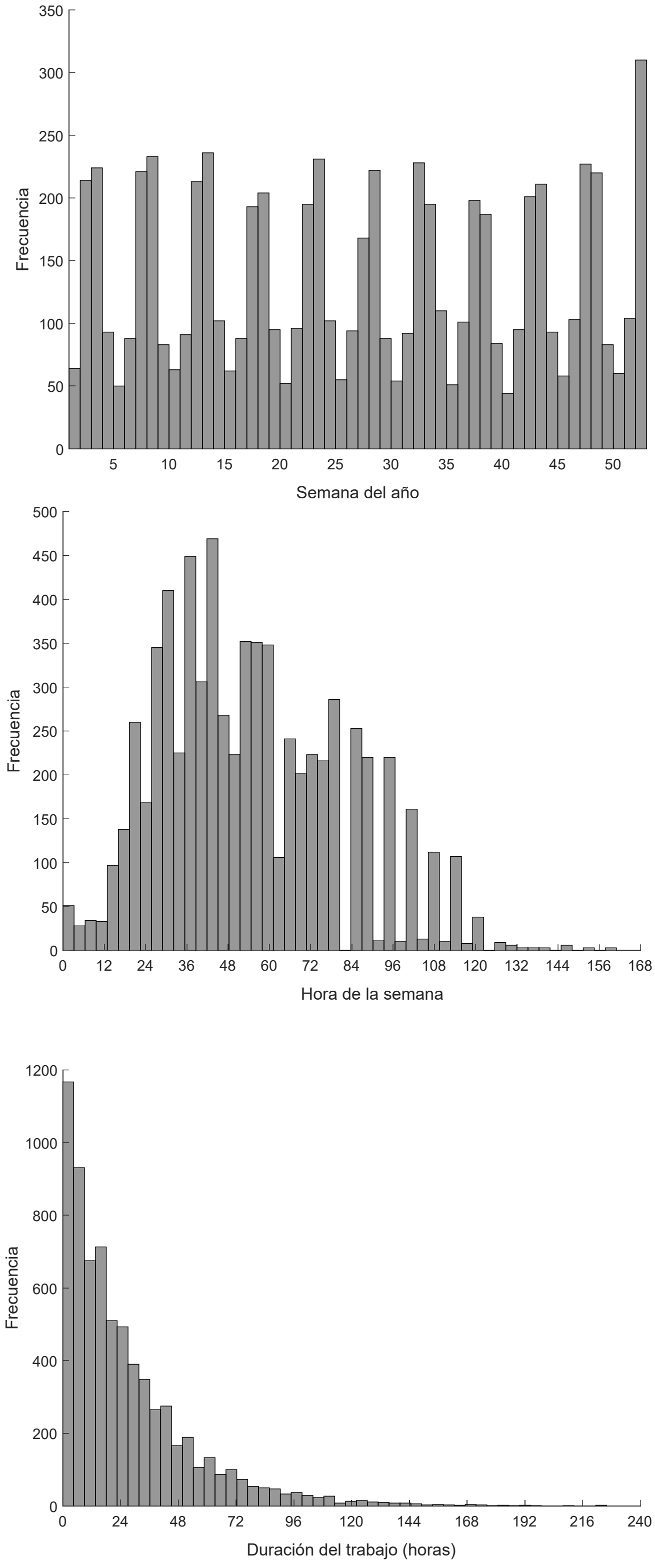


Figura 3.7: Histogramas de llegada de trabajos y duraciones en el Escenario 3.

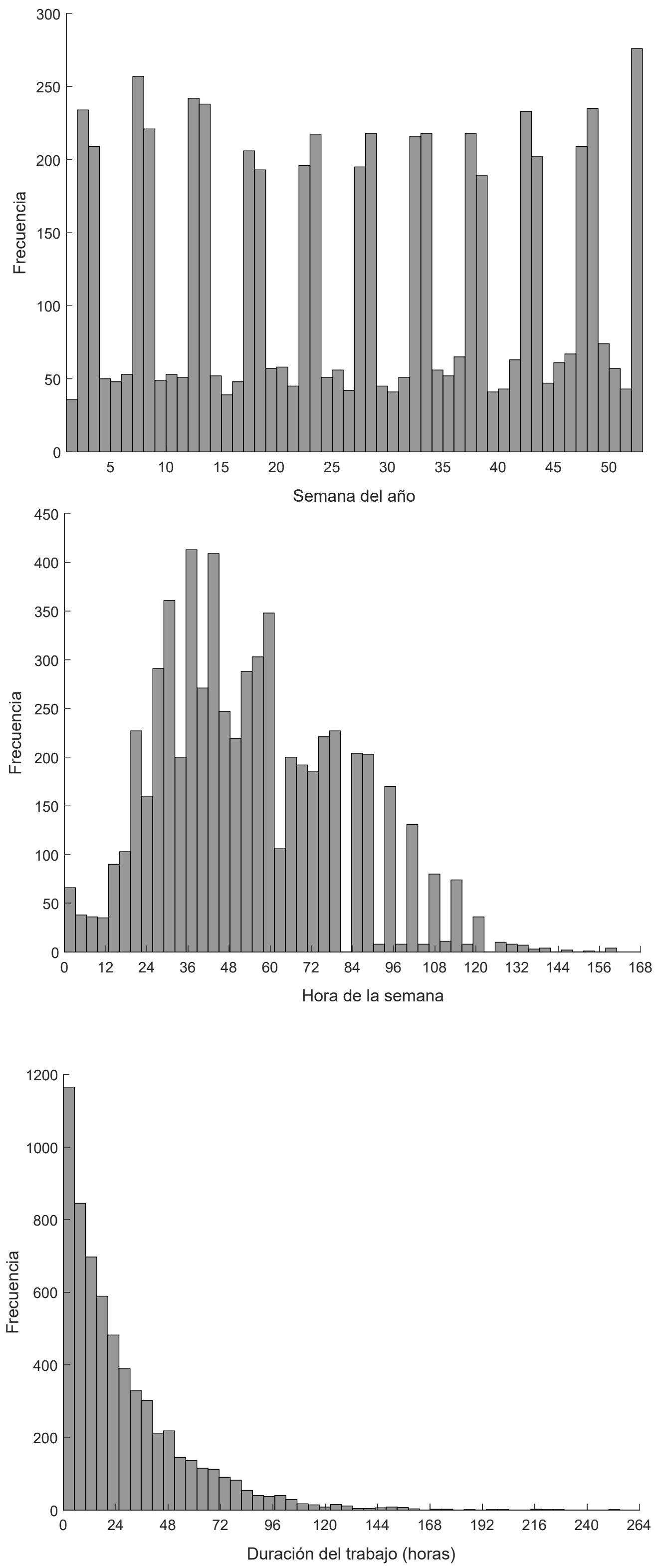


Figura 3.8: Histogramas de llegada de trabajos y duraciones en el Escenario 4.

y clasificar de acuerdo con las clases de entorno predefinidas. Dado que este proceso se debe implementar sobre dispositivos de baja potencia que dependen de baterías, el coste derivado del consumo de energía tiene un gran efecto sobre la reducción de la autonomía, y es imperativo equilibrar este coste con la precisión de clasificación de audio alcanzada por el audífono.

En la experimentación realizada se tomó como referencia un audífono con una batería de 145 miliamperios-hora (mAh) de capacidad y un consumo de corriente por Operación en Coma Flotante (FLOP, *Floating-point Operation*) de 2×10^{-5} miliamperios (mA). Para la clasificación de audio se utilizaron varios conjuntos de datos, incluyendo un conjunto balanceado[5] y varios conjuntos sintéticos para simular escenarios de uso real del audífono. A continuación de detallan estos conjuntos.

3.5.1. Conjunto de datos balanceado

El conjunto de datos balanceado consiste en un total de 2362 segundos de audio con muestras de tres clases de sonido: “voz”, “música” y “ruido”. Los ficheros de audio se muestrearon con una frecuencia de 16 KHz, 16 bits por muestra y una tamaño de ventana de 8 milisegundos. Se calcularon 25 MFCCs y se extrajeron las características a partir de la media aritmética y la desviación típica de los MFCCs para ventanas de 16 milisegundos con un desplazamiento de 4 milisegundos sobre las muestras. En total, cada instancia cuenta con un total 50 atributos. El detalle del número de ficheros de audio y de instancias extraídas se puede ver en la Tabla 3.3.

3.5.2. Escenarios sintéticos

Para validar los clasificadores aprendidos y evaluar la precisión de clasificación así como la duración de la batería, también se emplearon conjuntos imbalanceados que simulan condiciones reales de uso. En concreto, se definieron cuatro escenarios diferentes a partir de los datos de la Sección 3.5.1: trabajo, recreación al aire libre, deporte y concierto.

[5] Los conjuntos de datos *balanceados*, en contraposición con los conjuntos *imbalanceados*, son aquellos en los que no hay diferencias significativas entre las probabilidades a priori de pertenencia de una instancia a las diferentes clases [92].

Clase de sonido	Pista de audio	Núm. ficheros	Núm. instancias
Voz	Voz con/sin ruido y/o música	315	49140
Música	Vocal	96	14976
	Instrumental	219	34164
Ruido	Avión	28	4368
	Autobús	10	1560
	Cafetería	10	1560
	Coche	82	12792
	Guardería	19	2964
	Salón	10	1560
	Naturaleza	28	4368
	Pup	10	1560
	Colegio	10	1560
	Tienda	10	1560
	Deportes	19	2964
	Tráfico	16	2496
	Tren	28	4368
	Estación de tren	10	1560
	Lugar de trabajo	25	3900

Tabla 3.3: Relación de muestras de audio usadas en los experimentos.

Estos escenarios simulados consisten en un muestreo pseudoaleatorio de las instancias de sonido disponibles cuya clase y distribuciones de probabilidad dependen del escenario en cuestión y de la hora del día. Estos escenarios se caracterízan mediante una serie de ventanas temporales delimitadas, cada una con una probabilidad de muestreo para tipo de audio. Los escenarios se detallan las tablas 3.4, 3.5, 3.6 y 3.7. La Tabla 3.8 resume la distribución de clases de sonido para cada escenario.

3.6. Conjuntos de datos heterogéneos para validar los clasificadores

Además del problema Reconocimiento y Clasificación de Entornos Sonoros (SEC, *Sound Environment Classification*) indicado en la Sección anterior, los clasificadores diseñados en el marco de esta disertación se evaluaron y evolucionaron sobre una serie de conjuntos de datos heterogéneos con diversidad en el número de atributos, clases e instancias disponibles. En concreto, se tomó como referencia la relación de datos utilizada en [29] para la validación de los métodos propuestos. Para estos datos el

Escenario 1: Trabajo				Probabilidad de muestreo		
N	Comienzo	Fin	Descripción	Voz	Música	Ruido
1	0:00	6:30	Dispositivo apagado	-	-	-
2	6:30	7:30	Rutina matutina	30 %	-	70 %
3	7:30	9:00	Transporte	10 %	-	90 %
4	9:00	11:30	Trabajo	20 %	-	80 %
5	11:30	12:00	Café	50 %	-	50 %
6	12:00	14:00	Trabajo	20 %	-	80 %
7	14:00	15:00	Comida	50 %	-	50 %
8	15:00	18:00	Trabajo	20 %	-	80 %
9	18:00	19:30	Transporte	10 %	-	90 %
10	19:30	22:00	Tiempo libre en casa	20 %	70 %	10 %
11	22:00	23:00	Cena	50 %	-	50 %
12	23:00	0:00	Dispositivo apagado	-	-	-

Tabla 3.4: Periodos de tiempo y probabilidades de muestreo para cada tipo de sonido en el Escenario 1 (trabajo).

Escenario 2: Recreación al aire libre				Probabilidad de muestreo		
N	Comienzo	Fin	Descripción	Voz	Música	Ruido
1	0:00	6:30	Dispositivo apagado	-	-	-
2	6:30	7:30	Rutina matutina	30 %	-	70 %
3	7:30	9:00	Transporte	10 %	-	90 %
4	9:00	18:00	Tiempo en el exterior	10 %	-	90 %
5	21:00	22:00	Transporte	10 %	-	90 %
6	22:00	23:00	Cena	50 %	-	50 %
7	23:00	0:00	Dispositivo apagado	-	-	-

Tabla 3.5: Periodos de tiempo y probabilidades de muestreo para cada tipo de sonido en el Escenario 2 (recreación al aire libre).

Escenario 3: Deporte				Probabilidad de muestreo		
N	Comienzo	Fin	Descripción	Voz	Música	Ruido
1	0:00	9:00	Dispositivo apagado	-	-	-
2	9:00	10:00	Rutina matutina	30 %	-	70 %
3	10:00	11:00	Transporte	10 %	-	90 %
4	11:00	14:00	Deporte	10 %	-	90 %
5	14:00	15:00	Comida	50 %	-	50 %
6	15:00	20:00	Deporte	10 %	-	90 %
7	20:00	21:00	Transporte	10 %	-	90 %
8	21:00	22:00	Cena	50 %	-	50 %
9	22:00	0:00	Dispositivo apagado	-	-	-

Tabla 3.6: Periodos de tiempo y probabilidades de muestreo para cada tipo de sonido en el Escenario 3 (deporte).

Escenario 4: Concierto				Probabilidad de muestreo		
N	Comienzo	Fin	Descripción	Voz	Música	Ruido
1	0:00	10:00	Dispositivo apagado	-	-	-
2	10:00	16:00	Tiempo en casa	15 %	30 %	55 %
3	16:00	17:00	Transporte	10 %	-	90 %
4	17:00	22:00	Concierto	10 %	80 %	10 %
5	22:00	23:00	Cena en un restaurante	40 %	20 %	40 %
6	23:00	0:00	Transporte	10 %	-	90 %

Tabla 3.7: Periodos de tiempo y probabilidades de muestreo para cada tipo de sonido en el Escenario 4 (concierto).

	Voz	Música	Ruido
Escenario 1: Trabajo	20.53 %	9.89 %	69.58 %
Escenario 2: Recreación al aire libre	13.52 %	0.00 %	86.48 %
Escenario 3: Deporte	15.42 %	0.00 %	84.58 %
Escenario 4: Concierto	14.30 %	42.68 %	43.02 %

Tabla 3.8: Distribución de clases de sonido en cada escenario.

fitness tiene tres componentes: la tasa de error del clasificador, el coste computacional asociado a la clasificación de una instancia (medido en FLOPs), y el coste asociado a los atributos empleados por el clasificador. El detalle de los datos se puede ver en la Tabla 3.9. El primer bloque (*Hepatitis*, *Liver*, *Pima* y *Thyroid*) está formado por problemas de clasificación con costes reales intrínsecos asociados a los diferentes atributos. El resto de bloques no cuenta con unos costes asociados a los atributos, por lo que se generaron aleatoriamente siguiendo una distribución uniforme entre 0 y 1. Por último, los conjuntos de datos empleados están disponibles en los repositorios [1, 2, 14].

Nombre corto	Nombre completo	Atributos	Instancias	Clases
Hepatitis	Hepatitis	19	155	2
Liver	Liver Disorders	6	345	2
Pima	Pima Indians Diabetes	8	768	2
Thyroid	Thyroid Disease	20	3772	3
Letter	Letter Recognition	17	20000	26
Magic04	MAGIC Gamma Telescope	10	19020	2
Optdigits	Optical Recognition of Handwritten Digits	64	5620	10
Pendigits	Pen-Based Recognition of Handwritten Digits	16	7494	10
Sat	Statlog (Landsat Satellite)	36	4435	7
Segmentation	Statlog (Image Segmentation)	19	2310	7
Waveform	Waveform Database Generator (Version 1)	21	5000	3
Yeast	Yeast	8	1033	10
Glioblastoma	Glioblastoma	12625	50	4
CNS	Central Nervous System	7129	60	2
Colon	Colon Tumor	2000	62	2
DLBCL	Diffuse Large B-Cell Lymphoma (Stanford)	4026	47	2
Leukemia	Leukemia (ALL V.S. AML)	7129	72	2

Tabla 3.9: Conjuntos de datos heterogéneos utilizado para el diseño de los clasificadores.

4

Resultados

Este capítulo resume los métodos y soluciones propuestas en esta disertación, así como los resultados alcanzados respectos los objetivos de investigación establecidos anteriormente. Este capítulo se organiza en dos secciones, cada una correspondiente a uno de los aspectos fundamentales en los que se materializan las limitaciones del paradigma tradicional de optimización monoobjetivo del rendimiento.

4.1. Optimización de los centros de computación basada en la eficiencia energética

Mejorar la eficiencia energética de los clústeres de computación de alto rendimiento mediante técnicas de adaptación dinámica de sus recursos no es una tarea trivial. Su complejidad radica fundamentalmente en el modo de empleo de estas instalaciones. Al contrario que los clústeres de balanceo de carga, donde las peticiones son homogéneas desde el punto de vista de consumo de recursos y de tiempo de ejecución, en los clústeres de alto rendimiento cada petición solicita un volumen variable de recursos que puede ir desde un núcleo de procesador (*core*) a un servidor completo, así como un tiempo de ejecución indeterminado a priori, y que puede durar tanto segundos como semanas.

La naturaleza heterogénea de estas peticiones o trabajos (*jobs*) conduce a bruscas variaciones en el uso de recursos del clúster y que deben ser explotadas adecuadamente para poder adaptar los recursos disponibles y así reducir los consumos globales de energía.

Hasta ahora, los métodos propuestos dentro en esta línea de trabajo como [22], [54] o [130], se cimientan sobre el uso de Sistemas Basados en el Conocimiento como mecanismos de toma de decisión para el reajuste de los nodos de cómputo del clúster. En concreto, estos sistemas monitorizan constantemente los recursos utilizados, solicitados y disponibles del clúster, y toman decisiones en base a una serie de reglas predeterminadas que modelan conocimiento experto de forma genérica a cualquier escenario. Sin embargo, los clústeres de alto rendimiento no son agnósticos al escenario específico de uso. Es decir, la arquitectura de sistemas y de comunicaciones, la tipología de los elementos de cómputo y los patrones de actividad no son fácilmente generalizables, ya que son dependientes del objeto del clúster, y condicionan notablemente la distribución de las tasas de llegada de trabajos, sus duraciones, así como la cantidad y tipo de recursos demandados. Por este motivo, en [43] se propuso un Sistema Borroso Genético Híbrido (HGFS, *Hybrid Genetic Fuzzy System*) que combina un conjunto de reglas borrosas con otro conjunto de reglas no borrosas. La parte borrosa se establece con Aprendizaje Automático Basado en Genéticos (GBML, *Genetics-based Machine Learning*) por medio de MOEAss. Esta base de reglas borrosas permite una mejora de los resultados gracias a su diseño a la medida del clúster, tanto en términos de interpretabilidad linguística como de ahorro de energía, comparado con las bases de reglas estáticas. La parte no borrosa está formada por reglas estáticas que modelan conocimiento experto, y tienen por objetivo mejorar la robustez del sistema en caso de que la carga de trabajo del clúster evolucione hacia circunstancias no previstas dentro del aprendizaje de las reglas borrosas. De esta forma, la combinación de una base de reglas borrosas aprendidas a medida con una base de reglas estáticas predeterminada, permite alcanzar una mejora general de la eficiencia energética y de la capacidad de adaptación a las preferencias del administrador del clúster, pero manteniendo un buen comportamiento en escenarios no previstos durante el aprendizaje.

El método propuesto se validó utilizando los datos del Clúster de Modelización Científica de la Universidad de Oviedo (descrito en la Sección 3.4.1 y comparándolo frente a otros tres: un método básico de referencia que dispone tanto recursos como se necesitan en cada momento, el método propuesto en [54] con los parámetros configurados manualmente, y el propuesto en [37]. Los resultados obtenidos muestran como el HGFS logra el máximo ahorro de energía con el menor impacto para la fiabilidad del clúster (menor número de reconfiguraciones de nodos), a la vez que cumple el objetivo de no penalizar la Calidad de Servicio.

En [39] se propone y describe la herramienta software EECluster para transformar los clústeres de computación de alto rendimiento basados en los RMSs OGE/SGE y PBS/TORQUE en clústeres eficientes energéticamente con la capacidad de adaptar sus recursos a las necesidades de la carga, y cumpliendo con las prioridades e intereses de sus administradores. Esta herramienta incluye el aprendizaje automático y multiobjetivo de las reglas y parámetros de configuración, así como el mecanismo de toma de decisión HGFS, para mejorar la flexibilidad y los ahorros energéticos alcanzados con otras herramientas propuestas en [22, 54, 81]. La herramienta EECluster se puede descargar en [4, 5], se distribuye bajo la licencia BSD modificada o de tres cláusulas, y está inscrita en el Registro General de la Propiedad Intelectual con el número de asiento registral *05/2016/172* y bajo el título *EECluster: An Energy-Efficient software tool for managing HPC Clusters*.

En [40] se traslada el foco desde el coste energético hasta el concepto de eficiencia ecológica o *eco-efficiency*. De esta forma, el objetivo no es buscar el equilibrio óptimo entre Calidad de Servicio y consumo energético, sino entre Calidad de Servicio e impacto medioambiental. Con este nuevo enfoque se tienen en cuenta las fuentes de dióxido de carbono asociadas con el ciclo de vida de los servidores del clúster, incluyendo su fabricación, operación y sustitución, así como de los elementos auxiliares de refrigeración. Los experimentos realizados en esta publicación emplean los escenarios de carga descritos en las secciones 3.4.1 y 3.4.2, y muestran los resultados obtenidos con el HGFS bajo una serie de conjuntos de preferencias diferentes con respecto a la Calidad de Servicio, porcentajes y volúmenes de energía ahorrada y reducciones en la huella de carbono. Los resultados obtenidos verifican la capacidad de adaptación del mecanismo HGFS a diferentes escenarios y condiciones, y apoyan implantación de la herramienta EECluster para la reducción del impacto medioambiental de los clústeres de computadores del alto rendimiento.

En [41] se propone un mecanismo de toma decisiones proactivo que, en contraposición a los mecanismos reactivos propuestos hasta ahora en [22,43,54,81], adopta una estrategia predictiva para redimensionar y reconfigurar el clúster. Este nuevo enfoque tiene por objetivo aumentar la capacidad de adaptación a escenarios donde las fluctuaciones de la carga de trabajo son mayores, aprendiendo los patrones y tendencias a partir de los históricos del clúster para predecir la carga futura y actuar en consecuencia para maximizar los ahorros de energía. En concreto, el mecanismo de toma de decisiones proactivo es un controlador predictivo basado en el *framework* propuesto en [16, 17] cuya función de utilidad es un modelo borroso Takagi-Sugeno-Kang (TSK) [75, 110] aprendido por medio de algoritmos evolutivos multiobjetivo siguiendo un enfoque de aprendizaje distal supervisado [78]. En este modelo proactivo la decisión de adaptación del clúster se transforma en un problema de optimización sobre un horizonte temporal, pronosticado éste a partir de un modelo que integra la situación actual del clúster y la carga de trabajo esperada en el futuro, y cuyas posibles decisiones de reconfiguración se valoran por medio de una función de utilidad que implícitamente modela las preferencias del administrador del clúster. Este nuevo mecanismo se ha probado con los escenarios de carga descritos en las secciones 3.4.1 y 3.4.2, y comparado con los métodos propuestos en [54], [22], [37] y [43]. Los resultados obtenidos muestran que en escenarios con una fluctuación significativa en las tasas de llegada de trabajos, situación que es frecuente en los clústeres HPC, el mecanismo proactivo logra un mayor ahorro de energía con un menor impacto para la fiabilidad del equipamiento.

El mecanismo de toma de decisiones proactivo propuesto en [41] también fue objeto de dos contratos de investigación entre la Universidad de Oviedo, la Empresa ASAC Comunicaciones y la Fundación Universidad de Oviedo. El primero, con número de referencia FUO-EM-037-15 y título *Modelos Proactivos de Predicción para Maximizar la Eficiencia Energética en Clústeres de Computadores* tuvo por objetivo la transferencia de conocimiento a la Empresa sobre los mecanismos de toma de decisiones proactivos para incluir en su portfolio soluciones de clústeres HPC eficientes energéticamente. El segundo contrato de investigación, con número de referencia FUO-022-17 y título *Modelos Practivos de Predicción para Maximizar la Eficiencia Energética en Clústeres de Computadores: FASE II*, tuvo por objeto la extensión de los resultados obtenidos con el anterior proyecto a sistemas informáticos diferentes a los clústeres HPC, como el caso de los hipervisores utilizados habitualmente en la provisión de servicios de TI de propósito general.

Las publicaciones obtenidas en esta parte de la disertación son las siguientes:

A. Cocaña-Fernández, J. Ranilla and L. Sánchez, ***Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling***, The Journal of Supercomputing, Vol. 71, Issue 3, pp. 1163-1174, 2015.

A. Cocaña-Fernández, J. Ranilla and L. Sánchez, ***EECluster: An Energy-Efficient Tool for managing HPC Clusters***, Annals of Multicore and GPU Programming, Vol. 2, Issue 1, pp. 15-24, 2015.

A. Cocaña-Fernández, L. Sánchez and J. Ranilla, ***A software tool to efficiently manage the energy consumption of HPC clusters***, Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, pp. 1-8, 2015.

A. Cocaña-Fernández, L. Sánchez and J. Ranilla, ***Leveraging a predictive model of the workload for intelligent slot allocation schemes in energy-efficient HPC clusters***, Engineering Applications of Artificial Intelligence, Vol. 48, pp. 95-105, 2016.

A. Cocaña-Fernández, L. Sánchez and J. Ranilla, ***Improving the Eco-Efficiency of High Performance Computing Clusters Using EECluster***, Energies, Vol. 9, Issue 3, Art. 197, 2016.

Adicionalmente, se han obtenido los siguientes resultados:

A. Cocaña-Fernández, J. Ranilla and L. Sánchez, R. Cortina, ***EECluster: An Energy Efficient software tool for managing HPC Clusters*** (Programa de ordenador), Registro General de la Propiedad Intelectual, Número de Asiento Registral 05/2016/172, 2016. Divulgado en *The Journal of Supercomputing* el 30/10/2014.

Contrato de Investigación FUO-EM-037-15 entre la Universidad de Oviedo, la Empresa ASAC Comunicaciones y la Fundación Universidad de Oviedo, titulado ***Modelos Proactivos de Predicción para Maximizar la Eficiencia Energética en Clústeres de Computadores***, Duración: 9 meses. Firmado el 23 de enero de 2015.

Contrato de Investigación FUO-022-17 entre la Universidad de Oviedo, la Empresa ASAC Comunicaciones y la Fundación Universidad de Oviedo, titulado ***Modelos Proactivos de Predicción para Maximizar la Eficiencia Energética en Clústeres de Computadires: FASE II***, Duración: 8 meses. Firmado el 17 de enero de 2017.

4.2. Optimización de las técnicas de aprendizaje automático basada en la eficiencia computacional

Multitud de problemas de clasificación cuentan con una serie de restricciones inherentes a los mismos, ya sean éstas tiempos máximos de respuesta, consumos de energía, o costes computacionales. Este hecho supone en sí mismo una limitación para la aplicabilidad de los clasificadores tradicionales a problemas sensibles al coste, ya que siguen un enfoque exclusivo de optimización monoobjetivo de la precisión de clasificación. En la Sección 1.2 se resumieron una serie de métodos para la reducción de la complejidad de los clasificadores e, indirectamente, de los costes asociados a la misma, como las técnicas de selección de características, implementaciones eficientes de clasificadores tradicionales, clasificadores multietapa, poda de ensembles, diseño multiobjetivo de clasificadores, etc. No obstante, los métodos propuestos hasta ahora sólo optimizan los costes de forma indirecta, ya que no cuentan con la capacidad de aprender clasificadores explícitamente conscientes de los costes del problema en cuestión, con la consecuente limitación para su aplicabilidad en escenarios sensibles al coste.

Por este motivo, en [45] se propone una primera aproximación para construir clasificadores sensibles al coste basados en Sistemas Borrosos aprendidos por medio de Programación por Recocido Simulado Multiobjetivo (MOSA-P, *Multiobjetive Simulated Annealing Programming*). Este enfoque permite optimizar simultáneamente la precisión del clasificador con los costes asociados a la selección de características y a la clasificación de instancias, encontrando un Frente Eficiente de Pareto con los equilibrios óptimos entre costes computacionales y rendimientos de clasificación. El método propuesto se evaluó utilizando los datos descritos en la Sección 3.5.1, y se comparó con otro enfoque basado en la combinación de un algoritmo multiobjetivo (el NSGA-II [51]) para realizar selección de características, y de los clasificadores SVMs, C4.5, PART y k-NN (con $k = 1$ y $k = 5$). Los resultados obtenidos muestran una mejora general del MOSA-P frente a la alternativa con respecto a los costes energéticos de extracción de características y de clasificación, pero con la limitación de una precisión menor frente a los SVMss de kernel lineal empleados en el experimento.

En [46] se planteó una mejora de los resultados obtenidos en [45] mediante una revisión del genotipo de los individuos del MOSA-P, así como una mejora de la eficiencia computacional del proceso de aprendizaje mediante un filtrado previo basado en la Información Mutua (MI, *Mutual Information*) de los atributos con la clase, eliminando de esta forma aquellos atributos que son irrelevantes.

En [42] se propone un enfoque más avanzado del método anterior mediante un Clasificador Multietapa Basado en Reglas Borrosas (MFRBC, *Multistage Fuzzy Rule-Based Classifier*), aprendido por algoritmos multiobjetivo siguiendo un enfoque Pittsburgh [75] para optimizar conjuntamente la precisión y el conjunto heterogéneo de costes asociado al problema. Este método construye una jerarquía de clasificadores borrosos formado por n etapas en orden creciente de costes, donde cada etapa es un clasificador completo capaz de inferir la clase a partir de cualquier instancia. No obstante, todas las etapas con la excepción de la última tienen opción de rechazo, lo que significa que si una de estas etapas no tiene una certeza suficiente de que la clase correcta es la predicha, trasladarán la responsabilidad de la decisión a la etapa siguiente. Este proceso recursivo sólo finaliza cuando una etapa tiene la suficiente certeza para tomar una decisión, o cuando se alcanza la última etapa. El concepto de certeza se implementa de tal forma que los clasificadores de las etapas intermedias son conscientes del error que están cometiendo potencialmente dada su precisión limitada. Adicionalmente, las clases rechazadas por una etapa se omiten al evaluar las etapas posteriores. La idea general de esta solución es reducir los costes globales de clasificación logrando que la mayoría de las instancias de un problema se clasifiquen empleando clasificadores sencillos y de bajo coste dependientes de un menor número de atributos, y que sólo las instancias más complejas requieran el empleo de clasificadores complejos y de mayor coste. También se amplia la función de *fitness* para incluir la tríada de componentes que valoran la precisión de clasificación propuesta en [117], junto con el conjunto de costes heterogéneos asociados, para mejorar el rendimiento de clasificación de los individuos aprendidos así como su capacidad de generalización evitando el sobreaprendizaje.

Este nuevo método se comparó con las variantes sensibles al coste de los métodos de selección de características CFS y mRMR propuestas en [29]. Se utilizaron los datos descritos en la Sección 3.5.1, empleando validación cruzada con 10 repeticiones y comparando los frentes de Pareto de los diferentes métodos por medio del indicador binario ϵ detallado en [134] para evaluar la dominación de dichos frentes. La significancia de las dominaciones obtenidas para las 10 repeticiones se calculó mediante un test de Wilcoxon [124]. Los resultados muestran una mejora clara del clasificador MFRBC propuesto frente a las alternativas comparadas, tanto respecto a la dominación de los frentes de Pareto, como a la densidad y cobertura del conjunto óptimo de Pareto. Adicionalmente, en el Apéndice A se pueden encontrar los resultados de experimentos complementarios empleando los conjuntos de datos heterogéneos descritos en la Sección 3.6. Con objeto de facilitar la comprensión y visualización de los clasificadores MFRBC, en el Apéndice B se ha incluido la representación gráfica de dos clasificadores aprendidos en el problema SEC y explicados en [42]: el clasificador de menor coste y el clasificador más preciso.

En [36] el clasificador MFRBC se utilizó para encontrar un equilibrio óptimo entre precisión y autonomía en dispositivos embebidos dependientes de baterías. En concreto, se aprende utilizando el conjunto de datos balanceado de la Sección 3.5.1, y se comprueba la capacidad de generalización, rendimiento de clasificación y vida útil de la batería en los diferentes escenarios detallados en la Sección 3.5.2, simulando unas condiciones reales de uso. Comparado con las variantes del CFS y mRMR indicadas en [29], se logra una mejora sustancial de la autonomía de los dispositivos, a la vez que se alcanza el objetivo de precisión fijado para el experimento.

Las publicaciones obtenidas en esta parte de la disertación son las siguientes:

A. Cocaña-Fernández, L. Sánchez, J. Ranilla, R. Gil-Pita and D. Ayllón, ***Energy-Efficient Sound Environment Classifier for Hearing Aids Based on Multi-objective Simulated Annealing Programming***, Proceedings of the 10th International Conference on Soft Computing Models in Industrial and Environmental Applications, Burgos, Spain, pp. 261-270, 2015.

A. Cocaña-Fernández, L. Sánchez, J. Ranilla, R. Gil-Pita and H. Sánchez-Hevia, ***Improving learning efficiency in multi-objective simulated annealing programming for sound environment classification***, Proceedings of the 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio de Janeiro, Brazil, pp. 1-5, 2016.

A. Cocaña-Fernández, J. Ranilla, L. Sánchez and R. Gil-Pita, ***Multicriteria design of energy-conscious fuzzy rule-based classifiers for embedded devices***, Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE), Cádiz, Spain, pp. 372-375, 2016.

A. Cocaña-Fernández, J. Ranilla, R. Gil-Pita and L. Sánchez, ***Multicriteria design of cost-conscious fuzzy rule-based classifiers***, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, *Aceptado*, 2017.

A. Cocaña-Fernández, J. Ranilla, R. Gil-Pita and L. Sánchez, ***Energy-Conscious Fuzzy Rule-based Classifiers for Battery Operated Embedded Devices***, Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Naples, Italy, pp. 1-6, 2017.

5

Conclusiones y líneas futuras de trabajo

En esta disertación se han abordado las principales limitaciones inherentes al paradigma tradicional de optimización monoobjetivo del rendimiento absoluto, sustituyéndolo por un nuevo paradigma multiobjetivo que tiene en cuenta los efectos colaterales y costes incurridos, en pro de construir soluciones software e infraestructuras de cómputo sostenibles. En concreto, se exploraron dos aspectos: la eficiencia energética en las grandes infraestructuras de cómputo y la eficiencia computacional de los algoritmos y técnicas de aprendizaje automático.

En relación al primer aspecto investigado, se desarrollaron diversas soluciones para optimizar conjuntamente el rendimiento computacional y los efectos derivados de los elevados consumos de energía en los clústeres HPC, equilibrando para ello calidad de servicio, costes de operación, impacto medioambiental y fiabilidad del hardware. En primer lugar, se diseñó un mecanismo de adaptación dinámica de los recursos de los clústeres HPC basado en un Sistema Borroso Genético Híbrido, y aprendido por medio de Algoritmos Evolutivos Multiobjetivos [43]. Esta nueva solución se validó experimentalmente sobre un escenario real basado en el Clúster de Modelización

Científica de la Universidad de Oviedo, logrando mejorar notablemente los ahorros de energía alcanzados con los KBS propuestos por otros autores, a la vez que se consigue una alineación precisa con el conjunto de preferencias subjetivas establecido para la operación del clúster en términos de calidad de servicio, ahorros de energía e impacto sobre la fiabilidad. Posteriormente, se implementó este mecanismo de adaptación en una solución software denominada *EECluster*. Esta herramienta, inscrita en el Registro General de la Propiedad Intelectual y distribuida bajo la licencia BDS modificada, permite transformar los clústeres HPC que emplean OGE/SGE y PBS/TORQUE en clústeres eficientes energéticamente capaces de adaptar dinámicamente sus recursos a las necesidades de la carga y cumpliendo con las preferencias y criterios de los administradores del clúster [39, 44]. A continuación, se puso el foco en la eficiencia ecológica, equilibrando para ello el rendimiento e impacto medioambiental del clúster, evaluando este último en función de las emisiones de dióxido de carbono asociadas al ciclo de vida del hardware y de los elementos auxiliares, reduciendo así la huella de carbono de estas instalaciones [40]. Finalmente, se diseñó un mecanismo proactivo más sofisticado para la adaptación dinámica del clúster. Este nuevo mecanismo, basado en un controlador predictivo, tiene en cuenta tanto el escenario presente del clúster como el esperado en un futuro cercano para optimizar la disposición de recursos de acuerdo con una función de utilidad [41]. Tras realizar experimentaciones en diferentes escenarios, se demostró que este nuevo sistema permite lograr mejores resultados que los anteriores en clústeres cuya carga de trabajo fluctúa con ciertas estacionalidades previsibles en base a los registros históricos, lo que es habitual en los clústeres HPC. Así mismo, este mecanismo fue objeto de dos contratos de investigación y de transferencia de conocimiento entre la Universidad de Oviedo, la Empresa ASAC Comunicaciones y la Fundación Universidad de Oviedo.

En el marco del segundo aspecto investigado, se diseñaron algoritmos de aprendizaje automático capaces de optimizar conjuntamente la precisión del clasificación y los costes asociados con la clasificación de las instancias, ya sean éstos computacionales, económicos, energéticos, etc. Como primera aproximación, se desarrolló un clasificador sensible al coste basado en Sistemas Borrosos, aprendido mediante Programación por Recocido Simulado Multiobjetivo, y que permite optimizar simultáneamente los costes asociados a la selección de características y a la clasificación de instancias [45]. Posteriormente, se optimizó la eficiencia del algoritmo de aprendizaje [46]. A continuación se propuso un enfoque más avanzado por medio de Clasificadores Multietapa Basados en Reglas Borrosas aprendidos por algoritmos multiobjetivo, y que mejoró notablemente los resultados anteriores [38, 42]. Este clasificador se puso a prueba en dispositivos ligados a baterías, donde se pudo validar experimentalmente el efecto positivo de su eficiencia computacional sobre la duración de las baterías frente a otras alternativas de clasificadores disponibles en la literatura de aprendizaje automático [36].

En el futuro se abordarán diversas líneas de investigación centradas tanto en la optimización de los recursos de infraestructuras heterogéneas de cómputo, como también de algoritmos para el procesamiento de grandes volúmenes de datos. En primer lugar, se investigará sobre la aplicación de mecanismos proactivos de adaptación dinámica de los recursos en clústeres de balanceo de carga construidos sobre plataformas de virtualización. También se trabajará sobre la mejora de la eficiencia energética en diferentes dispositivos embebidos o portátiles, tanto ligados a fuentes de energía autónomas como regenerativas. En segundo lugar, se continuará investigando sobre la construcción de soluciones software eficientes para el procesamiento inteligente de grandes volúmenes de datos en escenarios sensibles al coste. Con respecto al empleo de algoritmos de optimización, se pondrá el foco sobre la optimización simultánea de más de tres objetivos u optimización *many-objective*, con objeto de diseñar soluciones que aborden de forma más eficiente una mayor cobertura de las preferencias, prioridades y limitaciones del escenario a estudiar.

Bibliografía

[1] *Bioinformatics Laboratory*. http://www.biolab.si/supp/bi-cancer/projections/.

[2] *Cancer Program Legacy Publication Resources*. http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi.

[3] *Clúster de Modelización Científica de los Servicios Científico Técnicos de la Universidad de Oviedo*. http://cms.uniovi.es/.

[4] *EECluster: A tool for energy-efficient resource management in HPC clusters*. https://pirweb.edv.uniovi.es/eecluster.

[5] *EECluster download — SourceForge.net*. https://sourceforge.net/projects/eecluster/.

[6] *Information Retrieval and Parallel Computing Group (IRPCG), University of Oviedo*. https://pirweb.edv.uniovi.es/.

[7] *MOEA Framework, a Java library for multiobjective evolutionary algorithms*. http://moeaframework.org/.

[8] *November 2016 — The Green500*. https://www.top500.org/green500/lists/2016/11/.

[9] *November 2016 — TOP500 Supercomputer Sites*. https://www.top500.org/lists/2016/11/.

[10] *Oracle Grid Engine - Wikipedia, the free encyclopedia.* http://en.wikipedia.org/wiki/Oracle_Grid_Engine.

[11] *Portable Batch System - Wikipedia, the free encyclopedia.* http://en.wikipedia.org/wiki/Portable_Batch_System.

[12] *Server Virtualization & Cloud Infrastructure: VMware vSphere.* http://www.vmware.com/products/vsphere.

[13] *TORQUE Resource Manager - Wikipedia, the free encyclopedia.* http://en.wikipedia.org/wiki/TORQUE_Resource_Manager.

[14] *UCI Machine Learning Repository: Data Sets.* http://archive.ics.uci.edu/ml/datasets.html.

[15] *XenServer - Server Virtualization and Consolidation - Citrix.* http://www.citrix.com/products/xenserver/overview.html.

[16] S. Abdelwahed, J. Bai, R. Su, and N. Kandasamy, *On the application of predictive control techniques for adaptive performance management of computing systems*, IEEE Transactions on Network and Service Management, 6 (2009), pp. 212–225.

[17] S. Abdelwahed, N. Kandasamy, and S. Neema, *A control-based framework for self-managing distributed computing systems*, in Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems - WOSS '04, New York, New York, USA, Oct. 2004, ACM Press, pp. 3–7.

[18] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, *Efficient Machine Learning for Big Data: A Review*, Big Data Research, 2 (2015), pp. 87–93.

[19] R. Alcalá, M. J. Gacto, and F. Herrera, *A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems*, IEEE Transactions on Fuzzy Systems, 19 (2011), pp. 666–681.

[20] R. Alcalá, M. J. Gacto, F. Herrera, and J. Alcalá-Fdez, *A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and*

compact linguistic fuzzy rule-based systems, International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems, 15 (2007), pp. 539–557.

[21] J. Alcalá-Fdez, R. Alcalá, and F. Herrera, *A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning*, IEEE Transactions on Fuzzy Systems, 19 (2011), pp. 857–872.

[22] F. Alvarruiz, C. de Alfonso, M. Caballer, and V. Hernández, *An Energy Manager for High Performance Computer Clusters*, in 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications, IEEE, July 2012, pp. 231–238.

[23] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, *Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7657 LNCS (2012), pp. 216–223.

[24] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, *Energy efficient smartphone-based activity recognition using fixed-point arithmetic*, Journal of Universal Computer Science, 19 (2013), pp. 1295–1314.

[25] C. Bash and G. Forman, *Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center*, USENIX Association, June 2007, p. 29.

[26] L. Bedogni, M. Di Felice, and L. Bononi, *By train or by car? Detecting the user's motion type through smartphone sensors data*, in IFIP Wireless Days, 2012.

[27] J. L. Berral, Í. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, *Towards energy-aware scheduling in data centers using machine learning*, in Proceedings of the 1st International Conference on EnergyEfficient Computing and Networking eEnergy 10, vol. 2, New York, New York, USA, apr 2010, ACM Press, p. 215.

[28] A. Betancourt, M. M. Lopez, C. S. Regazzoni, and M. Rauterberg, *A sequential classifier for hand detection in the framework of egocentric vision*, in

IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 600–605.

[29] V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos, *A framework for cost-based feature selection*, Pattern Recognition, 47 (2014), pp. 2481–2489.

[30] A. Bölte and U. W. Thonemann, *Optimizing simulated annealing schedules with genetic programming*, European Journal of Operational Research, 92 (1996), pp. 402–416.

[31] J. Cacheiro, *Analysis of Batch Systems*, tech. report, CESGA, 2014.

[32] Y. Cheng and Y. Zeng, *Automatic Energy Status Controlling with Dynamic Voltage Scaling in Power-Aware High Performance Computing Cluster*, in 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE, Oct. 2011, pp. 412–416.

[33] G. L. T. Chetsa, L. Lefrvre, J.-M. Pierson, P. Stolf, and G. Da Costa, *A Runtime Framework for Energy Efficient HPC Systems without a Priori Knowledge of Applications*, in 2012 IEEE 18th International Conference on Parallel and Distributed Systems, IEEE, Dec. 2012, pp. 660–667.

[34] Cisco, *The Zettabyte Era Trends and Analysis - Cisco*, 2016.

[35] Citrix, *Citrix XenServer Virtualization, automation, and advanced management tools for the datacenter*, tech. report.

[36] A. Cocaña Fernández, J. Ranilla, R. Gil-Pita, and L. Sánchez, *Energy-Conscious Fuzzy Rule-based Classifiers for Battery Operated Embedded Devices*, in Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–6.

[37] A. Cocaña Fernández, J. Ranilla, and L. Sánchez, *Energy-Efficient Allocation of Computing Node Slots in HPC Clusters through Evolutionary Multi-Criteria Decision Making*, in Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2014, 2014, pp. 318–330.

[38] A. Cocaña Fernández, J. Ranilla, L. Sánchez, and R. Gil-Pita, *Multicriteria design of energy-conscious fuzzy rule-based classifiers for embedded devices*, in Proceedings of the 16th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE), 2016, pp. 372–375.

[39] A. Cocaña Fernández, L. Sánchez, and J. Ranilla, *A software tool to efficiently manage the energy consumption of HPC clusters*, in 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, aug 2015, pp. 1–8.

[40] A. Cocaña Fernández, L. Sánchez, and J. Ranilla, *Improving the Eco-Efficiency of High Performance Computing Clusters Using EECluster*, Energies, 9 (2016), p. 197.

[41] ——, *Leveraging a predictive model of the workload for intelligent slot allocation schemes in energy-efficient HPC clusters*, Engineering Applications of Artificial Intelligence, 48 (2016), pp. 95–105.

[42] A. Cocaña-Fernández, J. Ranilla, R. Gil-Pita, and L. Sánchez, *Multicriteria design of cost-conscious fuzzy rule-based classifiers*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, (2017).

[43] A. Cocaña-Fernández, J. Ranilla, and L. Sánchez, *Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling*, The Journal of Supercomputing, 71 (2014), pp. 1163–1174.

[44] A. Cocaña-Fernández, L. Sánchez, and J. Ranilla, *EECluster: An Energy-Efficient Tool for managing HPC Clusters*, Annals of Multicore and GPU Programming, 2 (2015), pp. 15–24.

[45] A. Cocaña-Fernández, L. Sánchez, J. Ranilla, R. Gil-Pita, and D. Ayllón, *Energy-Efficient Sound Environment Classifier for Hearing Aids Based on Multi-objective Simulated Annealing Programming*, Springer International Publishing, 2015.

[46] A. Cocaña-Fernández, L. Sánchez, J. Ranilla, R. Gil-Pita, and H. Sanchez-Hevia, *Improving learning efficiency in multi-objective simulated annealing programming for sound environment classification*, in 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Río de Janeiro, Brazil, 2016, IEEE, pp. 1–5.

[47] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, *Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier*, Neurocomputing, 150 (2015), pp. 289–303.

[48] B. S. J. Costa, C. G. Bezerra, L. A. Guedes, and P. P. Angelov, *Unsupervised classification of data streams based on Typicality and Eccentricity Data Analytics*, in 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, jul 2016, IEEE, pp. 58–63.

[49] M. P. Cummings and J. C. Huskamp, *Grid Computing*, EDUCAUSE Review, 40 (2005), pp. 116–117.

[50] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine, and H. Chan, *Autonomic multi-agent management of power and performance in data centers*, (2008), pp. 107–114.

[51] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 6 (2002), pp. 182–197.

[52] P. Delforge and J. Whitney, *Issue Paper: Data Center Efficiency Assessment scaling up energy efficiency across the Data Center Industry: evaluating Key Drivers and Barriers*, tech. report, Natural Resources Defense Council (NRDC), 2014.

[53] C. Demir and E. Alpaydin, *Cost-conscious classifier ensembles*, Pattern Recognition Letters, 26 (2005), pp. 2206–2214.

[54] M. F. Dolz, J. C. Fernández, S. Iserte, R. Mayo, E. S. Quintana-Ortí, M. E. Cotallo, and G. Díaz, *EnergySaving Cluster experience in CETA-CIEMAT*, in 5th Iberian GRID Infrastructure conference, Santander, 2011.

[55] M. Ebbers, Mike Archibald, C. F. F. da Fonseca, M. Griffel, V. Para, and M. Searcy, *Smarter Data Centers: Achieving Greater Efficiency*, tech. report, IBM Redpaper, 2011.

[56] E. N. Elnozahy, M. Kistler, and R. Rajamony, *Energy-efficient server clusters*, (2002), pp. 179–197.

[57] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, *A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions*, IEEE Transactions on Fuzzy Systems, 21 (2013), pp. 45–65.

[58] W. Forrest, J. M. Kaplan, and N. Kindler, *Data centers: How to cut carbon emissions and costs*, tech. report, McKinsey & Company, 2008.

[59] P. J. Fortier and H. E. Michel, *Computer systems performance evaluation and prediction*, Digital Press, 2003.

[60] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. L. Rountree, and M. E. Femal, *Analyzing the Energy-Time Trade-Off in High-Performance Computing Applications*, IEEE Transactions on Parallel and Distributed Systems, 18 (2007), pp. 835–848.

[61] Gartner, *Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions*. http://www.gartner.com/newsroom/id/503867, 2007.

[62] R. Ge, X. Feng, W.-c. Feng, and K. W. Cameron, *CPU MISER: A Performance-Directed, Run-Time System for Power-Aware Clusters*, in 2007 International Conference on Parallel Processing (ICPP 2007), IEEE, Sept. 2007, pp. 18–18.

[63] H. Ghasemzadeh, P. Panuccio, S. Trovato, G. Fortino, and R. Jafari, *Power-aware activity monitoring using distributed wearable sensors*, IEEE Transactions on Human-Machine Systems, 44 (2014), pp. 537–544.

[64] O. Giustolisi, *Using a multi-objective genetic algorithm for SVM construction*, Journal of Hydroinformatics, 8 (2006), pp. 125–139.

[65] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, *The WEKA data mining software*, ACM SIGKDD Explorations Newsletter, 11 (2009), p. 10.

[66] V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass, *Signal Processing in High-End Hearing Aids: State of the Art, Challenges, and Future Trends*, EURASIP Journal on Advances in Signal Processing, 2005 (2005), pp. 2915–2929.

[67] R. Haring, M. Ohmacht, T. Fox, M. Gschwind, D. Satterfield, K. Sugavanam, P. Coteus, P. Heidelberger, M. Blumrich, R. Wisniewski, alan Gara, G. Chiu, P. Boyle, N. Chist, and C. Kim, *The IBM Blue Gene/Q Compute Chip*, IEEE Micro, 32 (2012), pp. 48–60.

[68] C.-H. Hsu and W.-c. Feng, *A Power-Aware Run-Time System for High-Performance Computing*, in ACM/IEEE SC 2005 Conference (SC'05), IEEE, 2005, pp. 1–1.

[69] C.-H. Hsu and U. Kremer, *The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction*, ACM SIGPLAN Notices, 38 (2003), p. 38.

[70] S. Huang and W. Feng, *Energy-Efficient Cluster Computing via Accurate Workload Characterization*, in 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE, 2009, pp. 68–75.

[71] B. Hui, Y. Yang, and G. I. Webb, *Anytime classification for a pool of instances*, Machine Learning, 77 (2009), pp. 61–102.

[72] IBM Systems and Technology Group, *IBM System Blue Gene/Q*, tech. report, IBM, Somers, NY, 2011.

[73] C. Igel, *Multi-objective model selection for support vector machines*, in Lecture Notes in Computer Science, vol. 3410, 2005, pp. 534–546.

[74] H. Ishibuchi, *Multiobjective Genetic Fuzzy Systems: Review and Future Research Directions*, in 2007 IEEE International Fuzzy Systems Conference, IEEE, jun 2007, pp. 1–6.

[75] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing)*, (2004).

[76] H. Ishibuchi and Y. Nojima, *Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning*, International Journal of Approximate Reasoning, 44 (2007), pp. 4–31.

[77] A. K. Jain, R. P. W. Duin, and J. Mao, *Statistical pattern recognition: A review*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (2000), pp. 4–37.

[78] M. Jordan and D. E. Rumelhart, *Forward models: Supervised learning with a distal teacher*, Cognitive Science, 16 (1992), pp. 307–354.

[79] D. Kangin, P. Angelov, and J. A. Iglesias, *Autonomously evolving classifier TEDAClass*, Information Sciences, 366 (2016), pp. 1–11.

[80] S. S. Keerthi, O. Chapelle, and D. DeCoste, *Building support vector machines with reduced classifier complexity*, Journal of Machine Learning Research, 7 (2006), pp. 1493–1515.

[81] S. Kiertscher, J. Zinke, S. Gasterstädt, and B. Schnor, *Cherub: Power consumption aware cluster resource management*, in Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom), Dec 2010, pp. 325–331.

[82] R. Kohavi and G. H. John, *Wrappers for feature subset selection*, Artificial Intelligence, 97 (1997), pp. 273–324.

[83] V. Kononen, J. Mantyjarvi, H. Simila, J. Parkka, and M. Ermes, *A computationally light classification method for mobile wellness platforms*, 2008, pp. 1167–1170. cited By 0.

[84] W. Lang, J. M. Patel, and J. F. Naughton, *On energy management, load balancing and replication*, ACM SIGMOD Record, 38 (2010), p. 35.

[85] Ó. D. Lara and M. A. Labrador, *A survey on human activity recognition using wearable sensors*, IEEE Communications Surveys and Tutorials, 15 (2013), pp. 1192–1209.

[86] Y.-J. Lee and O. L. Mangasarian, *RSVM: Reduced Support Vector Machines.*, in SDM, vol. 1, 2001, pp. 325–361.

[87] L. Li, U. Topkara, B. Coskun, and N. Memon, *CoCoST: A computational cost sensitive classifier*, in Proceedings - IEEE International Conference on Data Mining, ICDM, 2009, pp. 268–277.

[88] M. Li, H. McAllister, N. Black, and T. De Perez, *Perceptual time-frequency subtraction algorithm for noise reduction in hearing aids*, IEEE Transactions on Biomedical Engineering, 48 (2001), pp. 979–988.

[89] D. J. Lilja, *Measuring computer performance : a practitioner's guide*, Cambridge University Press, 2000.

[90] M. Lim, V. Freeh, and D. Lowenthal, *Adaptive, Transparent Frequency and Voltage Scaling of Communication Phases in MPI Programs*, in ACM/IEEE SC 2006 Conference (SC'06), IEEE, Nov. 2006, pp. 14–14.

[91] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, *LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy*, Neurocomputing, 123 (2014), pp. 424–435.

[92] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, *An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*, Information Sciences, 250 (2013), pp. 113 – 141.

[93] J.-B. Maj, L. Royackers, M. Moonen, and J. Wouters, *SVD-based optimal filtering for noise reduction in dual microphone hearing aids: a real time implementation and perceptual evaluation.*, IEEE transactions on bio-medical engineering, 52 (2005), pp. 1563–73.

[94] G. Martínez-Muñoz and A. Suárez, *Using boosting to prune bagging ensembles*, Pattern Recognition Letters, 28 (2007), pp. 156–165.

[95] M. Marzinzik, *Noise reduction schemes for digital hearing aids and their use for hearing impaired*, PhD thesis, Carl von Ossietzky University Oldenburg, 2000.

[96] National Science Foundation, *Advisory Committee for Cyberinfrastructure Task Force on Grand Challenges*, tech. report, 2011.

[97] A. M. Nia, M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, *Energy-Efficient Long-term Continuous Personal Health Monitoring*, IEEE Transactions on Multi-Scale Computing Systems, 1 (2015), pp. 85–98.

[98] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song, *E-Gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices*, in Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11, New York, New York, USA, 2011, ACM Press, p. 260.

[99] J. Pärkkä, L. Cluitmans, and M. Ermes, *Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree*, IEEE Transactions on Information Technology in Biomedicine, 14 (2010), pp. 1211–1215.

[100] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, *Monitoring motor fluctuations in patients with parkinsons disease using wearable sensors*, IEEE Transactions on Information Technology in Biomedicine, 13 (2009), pp. 864–873.

[101] P. Pierleoni, L. Pernini, A. Belli, and L. Palma, *An android-based heart monitoring system for the elderly and for patients with heart disease*, International Journal of Telemedicine and Applications, 2014 (2014).

[102] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, *Load balancing and unbalancing for power and performance in cluster-based systems*, in Workshop on compilers and operating systems for low power, vol. 180, Barcelona, Spain, 2001, pp. 182–195.

[103] M. Pratama, S. G. Anavatti, M. Joo, and E. D. Lughofer, *pClass: An Effective Classifier for Streaming Examples*, IEEE Transactions on Fuzzy Systems, 23 (2015), pp. 369–386.

[104] J. Pujara, H. Daumé, III, and L. Getoor, *Using classifier cascades for scalable e-mail classification*, in ACM International Conference Proceeding Series, 2011, pp. 55–63.

[105] T. Rault, A. Bouabdallah, and Y. Challal, *Energy efficiency in wireless sensor networks: A top-down survey*, Computer Networks, 67 (2014), pp. 104–122.

[106] M. Ring and B. M. Eskofier, *An approximation of the Gaussian RBF kernel for efficient classification with SVMs*, Pattern Recognition Letters, 84 (2016), pp. 1339–1351.

[107] T. Senator, *Multi-Stage Classification*, in Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE, 2005, pp. 386–393.

[108] Standard Performance Evaluation Corporation, *Readme 1st CPU2006*.

[109] F.-T. Sun, C. Kuo, and M. Griss, *PEAR: Power efficiency through activity recognition (for ECG-based sensing)*, in 2011 5th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2011, 2011, pp. 115–122.

[110] T. Takagi and M. Sugeno, *Fuzzy identification of systems and its applications to modeling and control*, IEEE Transactions on Systems, Man, and Cybernetics, SMC-15 (1985), pp. 116–132.

[111] G. Tang, Q. and Gupta, S. K S and Varsamopoulos, *Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach*, IEEE Transactions on Parallel and Distributed Systems, 19 (2008), pp. 1458–1472.

[112] O. Tekinalp and G. Karsli, *A new multiobjective simulated annealing algorithm*, Journal of Global Optimization, 39 (2007), pp. 49–77.

[113] The ACPI Promoters, *ACPI - Advanced Configuration and Power Interface*.

[114] The Economist Intelligence Unit, *IT and the environment A new item on the CIO's agenda?*, tech. report, The Economist, 2007.

[115] K. Trapeznikov, V. Saligrama, and D. Castañón, *Multi-Stage Classifier Design*, in JMLR W&CP, vol. 25, 2012, pp. 459–474.

[116] K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez, *Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?*, Knowledge-Based Systems, 54 (2013), pp. 3–21.

[117] K. Trawinski, O. Cordon, L. Sanchez, and A. Quirin, *A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers*, IEEE Transactions on Fuzzy Systems, 21 (2013), pp. 950–965.

[118] U.S. Environmental Protection Agency, *Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431*, tech. report, ENERGY STAR Program, 2007.

[119] G. L. Valentini, W. Lassonde, S. U. Khan, N. Min-Allah, S. A. Madani, J. Li, L. Zhang, L. Wang, N. Ghani, J. Kolodziej, H. Li, A. Y. Zomaya, C.-Z. Xu, P. Balaji, A. Vishnu, F. Pinel, J. E. Pecero, D. Kliazovich, and P. Bouvry, *An overview of energy efficiency techniques in cluster computing systems*, Cluster Computing, 16 (2011), pp. 3–15.

[120] S. Venkataramani, A. Raghunathan, J. Liu, and M. Shoaib, *Scalable-effort classifiers for energy-efficient machine learning*, in Proceedings of the 52nd Annual Design Automation Conference on - DAC '15, New York, New York, USA, jun 2015, ACM Press, pp. 1–6.

[121] VMware, *VMware Distributed Power Management: Concepts and Usage.* 2013.

[122] S. Voss, *Meta-heuristics: The State of the Art*, in Local Search for Planning and Scheduling, A. Nareyek, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 1–23.

[123] Wikipedia, *Mel-frequency cepstrum.*

[124] F. Wilcoxon, *Individual Comparisons by Ranking Methods*, Biometrics Bulletin, 1 (1945), pp. 80–83.

[125] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Elsevier, 2011.

[126] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, *Data mining with big data*, IEEE Transactions on Knowledge and Data Engineering, 26 (2014), pp. 97–107.

[127] Z. Xu, M. J. Kusner, K. Q. Weinberger, and M. Chen, *Cost-sensitive tree of classifiers*, in 30th International Conference on Machine Learning, ICML 2013, no. PART 1, 2013, pp. 133–141.

[128] Z. E. Xu, M. J. Kusner, K. Q. Weinberger, M. Chen, and O. Chapelle, *Classifier cascades and trees for minimizing feature evaluation cost*, Journal of Machine Learning Research, 15 (2014), pp. 2113–2144.

[129] B. Xue, M. Zhang, and W. N. Browne, *Particle swarm optimization for feature selection in classification: a multi-objective approach.*, IEEE transactions on cybernetics, 43 (2013), pp. 1656–71.

[130] Z. Xue, X. Dong, S. Ma, S. Fan, and Y. Mei, *An Energy-Efficient Management Mechanism for Large-Scale Server Clusters*, in The 2nd IEEE Asia-Pacific Service Computing Conference (APSCC 2007), IEEE, Dec. 2007, pp. 509–516.

[131] P. D. Yoo, J. W. Ng, and A. Y. Zomaya, *An Energy-Efficient Kernel Framework for Large-Scale Data Modeling and Classification*, in 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum, IEEE, may 2011, pp. 404–408.

[132] L. Zhang, J. Liu, H. Jiang, and Y. Guan, *SensTrack: Energy-efficient location tracking with smartphone sensors*, IEEE Sensors Journal, 13 (2013), pp. 3775–3784.

[133] Y. Zhang, S. Burer, and W. N. Street, *Ensemble pruning via semi-definite programming*, Journal of Machine Learning Research, 7 (2006), pp. 1315–1338.

[134] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, *Performance assessment of multiobjective optimizers: an analysis and review*, IEEE Transactions on Evolutionary Computation, 7 (2003), pp. 117–132.

[135] Z. Zong, M. Nijim, A. Manzanares, and X. Qin, *Energy efficient scheduling for parallel applications on mobile clusters*, Cluster Computing, 11 (2007), pp. 91–113.

[136] Z. Zong, X. Ruan, A. Manzanares, K. Bellam, and X. Qin, *Improving Energy-Efficiency of Computational Grids via Scheduling*, in Handbook of Research on P2P and Grid Systems for Service-Oriented Computing, N. Antonopoulos, G. Exarchakos, M. Li, and A. Liotta, eds., IGI Global, Jan. 2010, ch. 22.

Parte II: Publicaciones

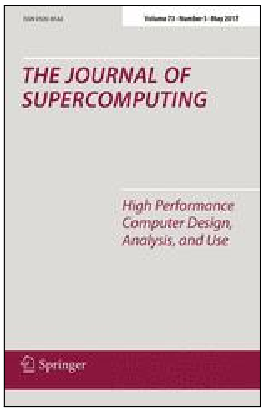
Título

Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling

Autores

Alberto Cocaña-Fernández, José Ranilla and Luciano Sánchez

Journal



The Journal of Supercomputing, *Volumen 71, Issue 3, Páginas 1163–1174, 2015*

DOI: 10.1007/s11227-014-1320-9

Ranking

Factor de impacto (JCR 2015): 1,088

Áreas:

Computer Science, Theory & Methods: 47/105 (Cuartil Q2)

Computer Science, Hardware & Architecture: 23/51 (Cuartil Q2)

Engineering, Electrical & Electronic: 147/257 (Cuartil Q3)

J Supercomput (2015) 71:1163–1174
DOI 10.1007/s11227-014-1320-9

Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling

Alberto Cocaña-Fernández · Jose Ranilla · Luciano Sánchez

Published online: 30 October 2014
© Springer Science+Business Media New York 2014

Abstract Decision-making mechanisms for online allocation of computer node slots in HPC clusters are commonly based on simple knowledge-based systems comprised of individual sets of if–then rules. In contrast with previous works where these rules were designed using expert knowledge, two different types of evolutionary learning algorithms are compared in this paper. In the first case, some of the numerical parameters defining a human-designed knowledge base are tuned. In the second case, a genetic fuzzy system evolves a partial rule set that, after being combined with some expert rules, conforms the most appropriate knowledge base for a given load scenario. In both cases, the proposed approaches optimize the quality of service and the number of node reconfigurations along with the energy consumption. An experimental study has been made using actual workloads from the Scientific Modeling Cluster at Oviedo University, and statistical evidence was found supporting the adoption of the new learning system.

Keywords Energy-efficient cluster computing · Multi-criteria decision making · Evolutionary algorithms

1 Introduction

High-performance computing clusters have become a very important element in both scientific and industrial communities because they are an excellent platform for solving

A. Cocaña-Fernández · J. Ranilla (✉) · L. Sánchez
Departamento de Informática, Universidad de Oviedo, Gijón, Spain
e-mail: ranilla@uniovi.es

A. Cocaña-Fernández
e-mail: cocanaalberto@gmail.com

L. Sánchez
e-mail: luciano@uniovi.es

a wide range of problems through parallel and distributed applications [5]. Nowadays, HPC clusters are, in fact, the main architecture for supercomputers (as shown in Top500 architecture distribution[1]) due to the high performance of commodity microprocessors and networks, to the standard tools for high-performance distributed computing, and to the lower price/performance ratio [41].

Nevertheless, this high performance comes at the price of consuming large amounts of energy. According to the U.S. Environmental Protection Agency [37], the consumption of data centers in USA was estimated at 61 billion kilowatt-hours (kWh) in 2006 for a total electricity cost of about $4.5 billion.

Large energy consumptions combined with notably increasing electricity prices in both EU [16] and USA [13] also have an important economical impact for IT companies, driving up power and cooling costs and forcing them to reduce operation costs [12,36].

The environmental impact of the high-energy consumption is also very significant. The EPA 2011 projected CO_2 emissions were 67.9 million metric tons [37]. Gartner estimates that the ICT industry accounts for 2 % of global CO_2 emissions, a figure equivalent to aviation [20].

This environmental and economical impact is the main bottleneck constraining the expansion of supercomputing and data centers and, therefore, a powerful motivation to maximize the efficiency of clusters. Moreover, a side effect of reducing the energy consumption of clusters is the reduction in heat dissipation, what can increase reliability. Also, it produces a cascade effect reducing the consumption of auxiliary devices such as Power Supply Units, power distribution, cooling, lighting and building switchgear, what further encourages to look for energy efficiency in cluster computing [15].

Many methods have been proposed within the field of energy-efficient cluster computing following both static and dynamic approaches. An example of static approach is the development of low-power CPUs such as the IBM PowerPC A2 of IBM Blue Gene/Q [22,26], or the use of GPUs and Intel Xeon Phi coprocessors. On the other hand, dynamic approaches adapt the cluster to its resource requirements at every given moment, thus saving energy when not needed [38]. An example is the Dynamic Voltage and Frequency Scaling (DVFS) technique, which reduces CPU voltage and frequency when the CPU is idle or under-used. This technique was used in [6,7,18,21,23–25,30]. Other examples are the software frameworks to develop energy-efficient applications, such as [1,17,29,33,39], energy-efficient job schedulers [42,43] and thermal-aware methods [3,35].

However, the most relevant technique for this paper is the adaptive resource cluster, which consists mainly in switching on and off cluster compute nodes, adapting to the requested resources at every moment and, therefore, saving energy. This technique was first introduced in [32] for Load-Balancing clusters, and was also used in [4,9,14,19,28,31] and in VMware vSphere[2] and Citrix XenServer hypervisors.[3]

[1] June 2014 | TOP500 Supercomputer Sites, http://www.top500.org/lists/2014/06/.

[2] VMware Distributed Power Management Concepts and Use. http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.

[3] Citrix XenServer—Efficient Server Virtualization Software. http://www.citrix.com/products/xenserver/overview.html.

Recently, it has also been applied to HPC clusters in [2,11] or [40]. In these works, the decision-making mechanism for determining the adequate resources (e.g., number of compute node slots) at every moment is based on a simple knowledge-based system (KBS) comprised of an individual set of if–then rules. The KBS constantly monitors requested, idle and available resources. The rule base governing this system is made to depend on certain configuration parameters such as the time of inactivity to shutdown nodes. These parameters are tuned by hand, according to the experience of the administrator.

According to our own experience, these systems are not location-agnostic. To obtain the best energy saving, both the set of rules defining the system and the parameters on which the rules depend must be optimized for the actual load scenario.

Otherwise, the results either would interfere with the desired operation of the cluster or would not save as much energy as it could be possible. Because of this, we proposed in [8] a cluster management system that works with both OGE/SGE and PBS/TORQUE resource management systems (RMS), whose decision-making mechanism shares the same rule set proposed in [11], but whose numerical parameters were obtained by means of a multiobjective evolutionary algorithm in a machine learning approach. The results of the KBS implemented in [8] are suitable for many practical situations and capable of yielding good results in terms of compliance with administrator preferences in all QoS, energy saved and node reconfigurations.

Since the publication of this last reference, the question has been raised whether the mentioned rule base was optimal or, on the contrary, there exist an alternate definition of the rule base for which the behavior of the system could be further improved. To answer this question, in this paper, a system is developed that learns the linguistic definition of a part of the aforementioned knowledge base, making it depend on the cluster behavior. The learned part will be combined with expert rules to produce a new system whose results compare favorably to that of reference [8]. In particular, we present here a Hybrid Genetic Fuzzy System (HGFS) that combines both a fuzzy and a non-fuzzy set of rules. The fuzzy part is learned by means of a genetic-based machine leaning (GBML) multiobjective evolutionary algorithm (MOEA). The purpose of using a HGFS is to achieve better results in both linguistic interpretability and efficiency compared to the static KBS implemented in [8]

The remainder of the paper is as follows. Section 2 explains the architecture of the solution proposed. Section 3 explains the learning algorithm used. Section 4 shows the experimental results. Section 5 concludes the paper and discusses the future work.

2 Architecture

As mentioned in reference [8], the solution proposed consists on a service and an administration dashboard, coupled with a Database Management System, and deployed over an HPC cluster running a Resource Management System such as OGE/SGE or PBS/TORQUE.

Figure 1 provides a high-level overview of the system components. The mission of the EEClusterd service is to periodically synchronize with the system status using various components and applications, and then use the knowledge-based system (KBS)

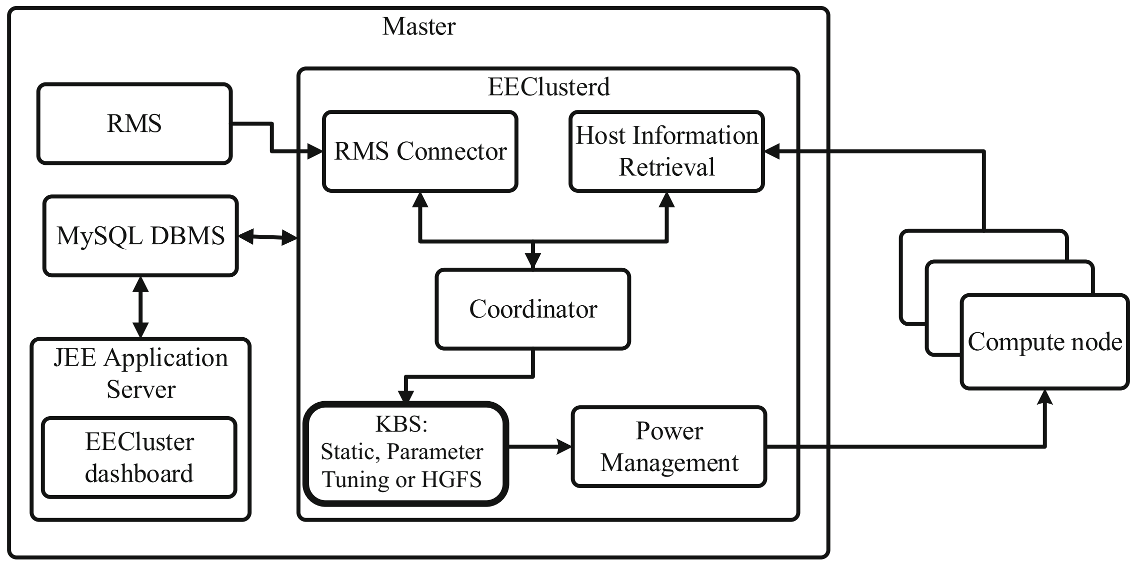


Fig. 1 System components overview

to make decisions on whether any reconfiguration of the compute nodes must be performed. The synchronization task of the service collects and keeps updated records of the RMS and of every compute node. RMS data include the cluster parallel environments (OGE/SGE), queues, hosts, users, and completed, queued and running jobs. The Power Management module is responsible for switching on/off the nodes appointed by the KBS with Ethernet cards or Intelligent Platform Management Interface cards (IPMI).

The KBS that is at the center of the decision-making system for Power Management is usually static (hand written by a human) [11] and comprises a rule-based decision system for powering on nodes and also an algorithmic procedure for switching off hosts. In this paper, it is proposed that either the parameters of the KBS or both their parameters and linguistic structure can be adapted to fit a particular cluster behavior by means of multiobjective genetic algorithms.

3 Static, parametric KBS and hybrid genetic fuzzy systems

Combining the mentioned static rule-based decision system for powering on nodes with a simple procedure for switching off machines is known to produce good results for a wide range of scenarios. Notwithstanding this, in this section, it will be shown that there is still room for improvement. On the one hand, the thresholds triggering the switching actions in the KBS can be tailored to the statistical properties of the load. On the other hand, different procedures for powering-off nodes can be used. It will also be shown that a Fuzzy Rule-Based Systems (FRBS) can be learned from data and used to replace the expert-defined knowledge base with a hybrid KBS combining some of the expert rules and the fuzzy rules that were learnt.

3.1 Tuning of KBS parameters

As mentioned, the KBS is the basis of the decision-making system for Power Management. Taking as a basis, the set of rules proposed in [11], a set of configuration parameters can be configured that can be adapted to most of the desired cluster behaviors. These rules are as follows:

- **if** $s_{\text{running}} + s_{\text{starting}} < s_{\min}$ **then** power on $(s_{\min} - (s_{\text{running}} + s_{\text{starting}}))$ slots
- **if** $t_{\text{avg}} > t_{\max}$ or $n_{\text{queued}} > n_{\max}$ **then** power on 1 slot
- **if** $t_{\text{avg}} < t_{\min}$ or $n_{\text{queued}} < n_{\min}$ **then** power off 1 slot
- **for each** h in $hosts$ **do**
  if $\text{idle}_h > \text{idle}_{\max}$ **then** power off host h

s_{running} and s_{starting} are the number of slots currently running and starting. $s_{\min}$ is the minimum number of slots required to run each of the queued jobs, i.e., the maximum requested slots of an individual job among the queued ones. t_{avg} is the average waiting time for the queued jobs, and $t_{\max}$ and $t_{\min}$ are, respectively, the maximum and minimum average waiting time for the queued jobs. n_{queued} is the number of queued jobs, and $n_{\max}$ and $n_{\min}$ are the maximum and minimum number of queued jobs before an action is performed. Finally, idle_h is the time that the host h has been at idle state and $\text{idle}_{\max}$ is the maximum time that a host can be at idle state. Observe that a particular instance of the knowledge-based system can, therefore, be expressed as a combination of five parameters: $(t_{\min}, t_{\max}, n_{\min}, n_{\max}, \text{idle}_{\max})$.

It is remarked that the KBS described before controls how many slots are powered on/off, but the specific nodes that will be reconfigured are not being identified; additional procedures are used to determine this. The selection of the precise nodes to be acted upon involves two values: the node efficiency and the timestamp of the last timed out node. Hosts are splitted first by whether they succeeded or failed to comply with the last order. Those that succeeded are sorted according to their efficiency so that powered-on nodes are the most efficient and powered-off nodes are the least efficient ones. Conversely, those that failed are sorted according to the timestamps of their failures; those with the earliest values are always chosen. This mechanism allows the system to continuously iterate through the potentially malfunctioning nodes, thus increasing the possibility of finding a repaired one.

3.1.1 Genetic learning of the parameters

As mentioned before, the advantage of the knowledge-based system detailed earlier is the ability to adapt to any desired working mode for the cluster due to the many configuration parameters that rule its operation. However, this ability to adapt comes with the problem of actually finding the right set of values to match the desired working mode. In reference [8], multiobjective evolutionary algorithms (MOEAs) were used to find the parameters defining the KBS, by optimizing a fitness function consisting in three conflicting criteria: the quality of service, the energy saved and the number of node reconfigurations. Specifically, the chosen MOEA is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10].

For a given set of n jobs, where the j-th job ($j = 1 \ldots n$) is scheduled to start at time tsch_j, but effectively starts at time ton_j and stops at time toff_j, the quality of service in a HPC cluster reflects the amount of time that each job has to wait before is assigned its requested resources. Once the job starts its execution, it will not be halted, thus we focus only on its waiting time. Because jobs do not last the same amount of time, their waiting in the queue is better expressed as a ratio considering their execution time. Finally, due to the potential existence of outlier values, the 90 percentile is used instead of average:

$$\text{QoS} = \min\left\{p : \left|\left|\left\{j \in 1\dots n : \frac{\text{ton}_j - \text{tsch}_j}{\text{toff}_j - \text{ton}_j} \le p\right\}\right|\right| > 0.9\,n\right\} \tag{1}$$

where $||A||$ is the cardinality of the set A.

The energy saved is measured as the sum of the amount of seconds that each node has been powered off. Let c be the number of nodes, let state(i, t) be 1 if the i-th node $(i = 1 \dots c)$ is powered at time t and 0 otherwise. Lastly, let the time scale be the lapse between tini=$\min_j\{\text{tsch}_j\}$ and tend= $\max_j\{\text{toff}_j\}$. Then,

$$\text{Energy saved} = c \cdot (\text{tend} - \text{tini}) - \sum_{i=1}^{c} \int_{\text{tini}}^{\text{tend}} \text{state}(i, t)\mathrm{d}t. \tag{2}$$

The node reconfigurations is the number of times that a node has been powered on or off. Let nd(i) the number of discontinuities of the function state(i, t) in the time interval $t \in (\text{tini}, \text{tend})$:

$$\text{Reconfigured nodes} = \sum_{i=1}^{c} \text{nd}(i) \tag{3}$$

3.2 Hybrid GFS

In addition to the parameter tuning mechanism defined in the preceding section, an alternate definition of the powering-off mechanism is proposed. The key component of this new definition is a Hybrid Genetic Fuzzy System (HGFS) implementing the decision-making mechanism that determines how many of the cluster resources must be on at every moment. This HGFS combines both non-fuzzy rules taken from reference [11] and fuzzy rules in the form of a zero-order Takagi–Sugeno–Kang (TSK) fuzzy model [27,34]. The structure of this hybrid system can be expressed as follows:

if $s_{\text{running}} + s_{\text{starting}} < s_{\min}$ **then** power on $(s_{\min} - (s_{\text{running}} + s_{\text{starting}}))$ slots
if $t_{\text{avg}} > t_{\max}$ or $n_{\text{queued}} > n_{\max}$ **then** power on 1 slot
if $t_{\text{avg}} < t_{\min}$ or $n_{\text{queued}} < n_{\min}$ **then** power off 1 slot
for each h in *hosts* **do**
  if idle$_h$ is $\widetilde{T}_1$ **then** off $= w_1$
  if idle$_h$ is $\widetilde{T}_2$ **then** off $= w_2$
  if $\cdots$ **then** $\cdots$
  if idle$_h$ is $\widetilde{T}_N$ **then** off $= w_N$

Apart from the variables already defined in Sect. 3.1, $\widetilde{T}_1, \ldots, \widetilde{T}_N$ are triangular fuzzy sets forming a uniform fuzzy partition [27] of the domain of the variables idle_h, which measure the time that the h-th host has been at idle state. For instance, $\widetilde{T}_1$ may be "VERY SHORT", "SHORT", "MEDIUM", "LONG" or "VERY LONG". The values $w_1, \ldots, w_N$ are between 0 and 1 and can be understood as the degree of truth of the assert "the h-th node must be switched off". The idea under this expression is that the number of nodes that must actually be switched off is not decided on a purely host-by-host basis, but each host contributes with a certain weight to the number of actually off hosts. For instance, if the idle time on node 1 is very high but the idle time in nodes 2 and 3 is medium, the weight of first node will be one, and the weights of nodes 2 and 3 will be 0.5, thus the total number of powered-off nodes will be $1 + 0.5 + 0.5 = 2$. This is contrast with the behavior of the rule base in the preceding section, where the number of nodes could be $1+0+0 = 1$ if the idle times of nodes 2 and 3 are slightly lower than $\text{idle}_{\max}$ and jump to $1 + 1 + 1 = 3$ if the idle times of nodes 2 and 3 grow to a small degree but exceed $\text{idle}_{\max}$.

Also, let the intermediate function $\text{defuzz}(t)$ (which is the output of the zero-order TSK fuzzy model formed by the N fuzzy rules included in the KBS) be defined as follows:

$$\text{defuzz}(t) = \frac{\sum_{r=1}^{N} \widetilde{T}_r(t) \cdot w_r}{\sum_{r=1}^{N} \widetilde{T}_r(t)}. \tag{4}$$

Then, the number of nodes that are powered off is given by the following expression:

$$\text{Powered off nodes} = \left\lfloor \sum_{h=1}^{c} \text{defuzz}(\text{idle}_h) \right\rfloor. \tag{5}$$

Observe that a particular instance of the hybrid GFS can, therefore, be expressed as a combination of: $(t_{\min}, t_{\max}, n_{\min}, n_{\max}, \widetilde{T}_1, \ldots, \widetilde{T}_N, w_1, \ldots, w_N)$. The same evolutionary algorithm used for tuning the parameters, and guided by the same fitness function, can still be used to learn the extended set of parameters. It is remarked that in the current version of the learning algorithm, the membership functions $\widetilde{T}_1, \ldots, \widetilde{T}_N$ are not adjusted and a uniform partition is defined, but this is not a fundamental limitation; in our experimentations, any change in the membership function of these sets could be compensated by the corresponding modification in the weights w_i.

4 Experimental results

The experimental setup is based on actual workloads from the Scientific Modeling Cluster of the University of Oviedo (CMS) spanning 22 months, with a total of 2,907 jobs. CMS consists of three independent computing clusters and five transversal queues using PBS as a Resource Management Systems (RMS). A full description of CMS is given in its web site (http://cms.uniovi.es). For both training and testing, a cluster simulator has been developed so that every model can be evaluated in the three criteria described in the previous section.

Table 1 Experiment results for the training set

	Training set		
	QoS	Energy saved (s)	Reconfigurations
Single rule	157.80	1.26E+09	2,755
Rules (0, 60, 0, 5, 3600)	112.00	1.29E+09	2,047
Rules (0, 300, 0, 10, 3600)	184.10	1.29E+09	2,023
Rules (0, 60, 0, 5, 7200)	103.30	1.29E+09	1,945
Rules (0, 60, 0, 0, 14400)	93.66	1.28E+09	1,845
Rules NSGA-II	0.00	8.54E+08	81
Hybrid GFS NSGA-II	0.00	8.84E+08	75

Table 2 Experiment results for the test set

	Test set		
	QoS	Energy saved (s)	Reconfigurations
Single rule	80.16	4.22E+08	2,504
Rules (0, 60, 0, 5, 3600)	48.62	4.25E+08	1,538
Rules (0, 300, 0, 10, 3600)	77.43	4.26E+08	1,512
Rules (0, 60, 0, 5, 7200)	22.34	4.23E+08	1,386
Rules (0, 60, 0, 0, 14400)	2.92	4.19E+08	1,216
Rules NSGA-II	0.00	1.88E+08	47
Hybrid GFS NSGA-II	0.00	2.41E+08	42

Four solutions have been tested using this simulator and the workloads: a) a basic model, b) the rule model proposed in [11], with its parameters manually configured by the administrator, c) the learning mechanism proposed in [8], and d) the hybrid GFS proposed in this paper. The holdout method was used for validation, with a 70–30 % split in training and test.

The administrator preferences for the experiment are based upon a lexicographic ordering of the three criteria: the administrator always seeks the best QoS and the amount of energy saved is used only to break ties in QoS. In turn, the number of reconfigurations also serves to break ties in QoS and energy saving.

First, the basic model (labeled "Single rule" in Tables 1 and 2) consists on the allocation of as many compute node slots as are required to run all queued jobs, shutting down every idle node whenever the decision mechanism is triggered. Second, five different manual configurations were tested for the model in [11], intended to give different weights to QoS, energy and reconfigurations. Finally, the machine learning approach we proposed earlier in [8] (labeled "Rules NSGA-II") and the hybrid GFS proposed in this paper (labeled "Hybrid GFS NSGA-II") were applied to the same data.

As shown in the aforementioned Tables 1 and 2, none of the manual configurations neither the basic model was competitive with the machine learning approaches. The experimentation shows that finding manually a suitable configuration for the multi-rule

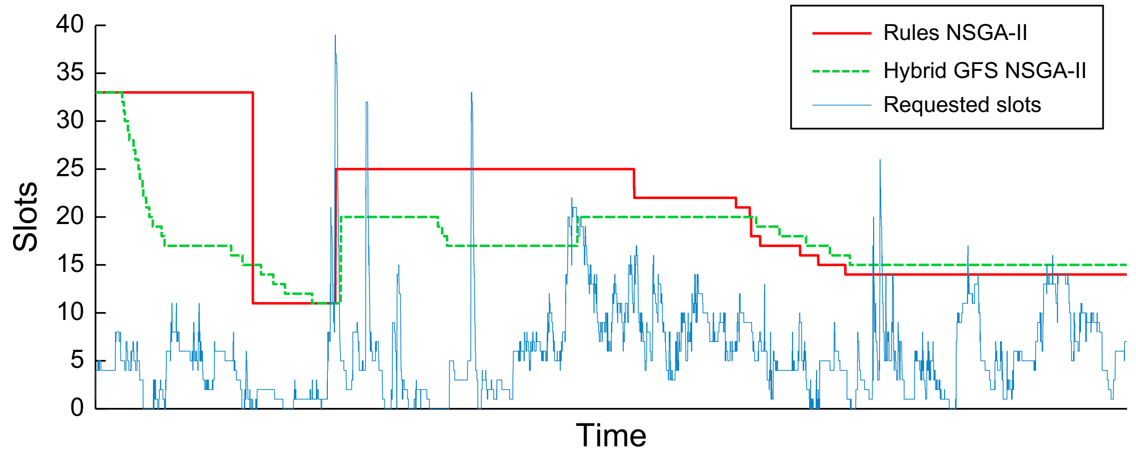


Fig. 2 Cluster simulation trace obtained in the experiment for the test set

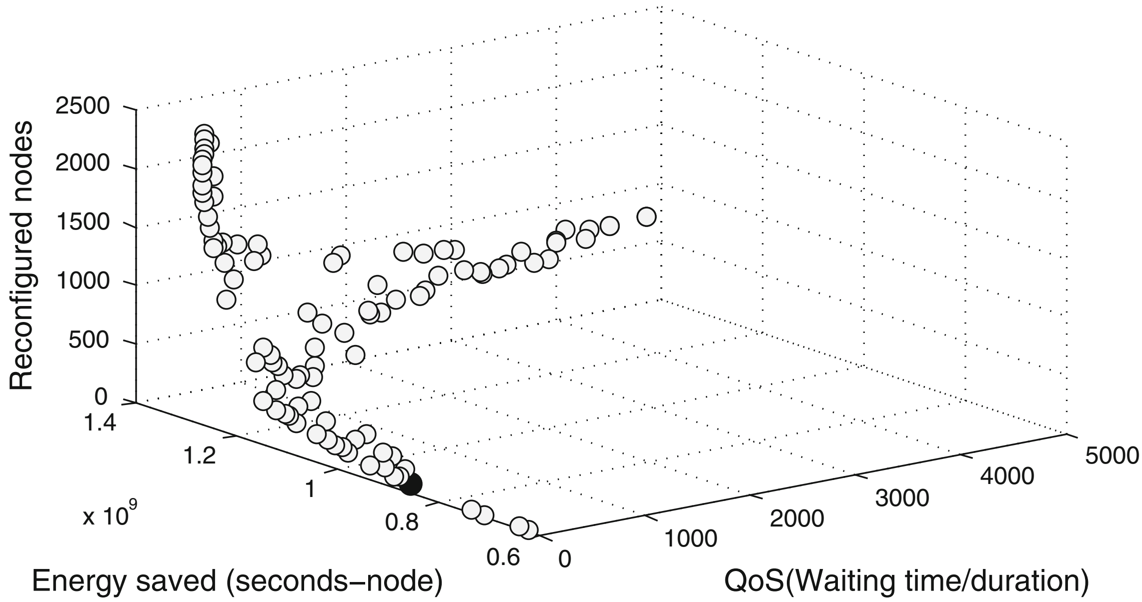


Fig. 3 Pareto Efficient Frontier obtained in the experiment

model is an infeasible task due to the large number of combinations. Regarding the machine learning approaches, the hybrid GFS achieves better results than the static KB used in our previous solution in both energy saved and node reconfigurations, as shown in Tables 1 and 2. The decisions made by each system are represented graphically in Fig. 2 through the cluster simulation trace of each experiment. This figure shows the evolution over time of the aggregated requested slots by the jobs and the slots powered on by each system. As can be seen, the hybrid GFS keeps, on average, a lower number of slots powered on, thus saving more energy, but without any significant impact on the QoS.

Lastly, the Pareto Efficient Frontier obtained in this experiment is represented in Figure 3. The chosen configuration, marked with a black dot in the figure, achieves optimal QoS and also saves energy while keeping acceptable node reconfigurations, thus complying with the previously declared preferences. Observe that many different balances between energy consumption, QoS and reconfigured nodes can be obtained from this set of solutions, and also that none of the manually found sets of parameters is part of the set of Pareto-optimal configurations.

5 Concluding remarks and future work

An evolutionary learning algorithm has been designed that is able to optimize the parameters defining the rules in the KBS that drives the Power Management module of a HPC cluster. The new procedure has been tested with actual workloads captured at the Scientific Modeling Cluster at Oviedo University. It has been found that expert knowledge is not enough for fine-tuning this system. Therefore, in a first stage, a learning system was tested and found to be able to produce a combination of parameters that improved the initial solution in the three criteria, at the same time: QoS, energy saving and node reconfiguration. In a second stage, a Hybrid Genetic Fuzzy System (HGFS) was defined that implements an alternate decision-making mechanism for determining how many of the cluster resources must be on at every moment. The HGFS further improved the performance of the first learning system.

One might wonder whether these approaches are general enough for being applied to different scenarios, and what the expected gain would be in those cases. Further work is needed to provide a sound answer to this question. On the one hand, it is clear that the KBS is highly dependent on the expected profile of the workload. On the other hand, for those cases where the load does not follow a regular pattern, the improvement over the simpler schemes might not be relevant enough.

Acknowledgments This work has been partially supported by "Ministerio de Economía y Competitividad" from Spain/FEDER under grants TEC2012-38142-C04-04 and TIN2011-24302.

References

1. Alonso P, Badia RM, Labarta J, Barreda M, Dolz MF, Mayo R, Quintana-Orti ES, Reyes R (2012) Tools for power-energy modelling and analysis of parallel scientific applications. In: 2012 41st international conference on parallel processing. IEEE, New Jersey, pp 420–429
2. Alvarruiz F, de Alfonso C, Caballer M, Hernández V (2012) An energy manager for high performance computer clusters. In: 2012 IEEE 10th international symposium on parallel and distributed processing with applications. IEEE, New Jersey, pp 231–238
3. Bash C, Forman G (2007) Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center. USENIX Association, Berkeley, p 29
4. Berral JL, Goiri Í, Nou R, Julià F, Guitart J, Gavaldà R, Torres J (2010) Towards energy-aware scheduling in data centers using machine learning. In: Proceedings of the 1st international conference on energy-efficient computing and networking—e-energy '10. ACM Press, New York, p 215
5. Buyya R, Jin H, Cortes R (2002) Cluster computing. Future Gener Comput Syst 18(3):v–viii
6. Cheng Y, Zeng Y (2011) Automatic energy status controlling with dynamic voltage scaling in power-aware high performance computing cluster. In: 2011 12th international conference on parallel and distributed computing, applications and technologies. IEEE, New York, pp 412–416
7. Chetsa GLT, Lefrvre L, Pierson JM, Stolf P, Da Costa G (2012) A runtime framework for energy efficient HPC systems without a priori knowledge of applications. In: 2012 IEEE 18th international conference on parallel and distributed systems. IEEE, New York, pp 660–667
8. Cocaña Fernández A, Ranilla J, Sánchez L (2014) Energy-efficient allocation of computing node slots in hpc clusters through evolutionary multi-criteria decision making. In: Proceedings of the 14th

international conference on computational and mathematical methods in science and engineering, CMMSE 2014, pp 318–330
9. Das R, Kephart JO, Lefurgy C, Tesauro G, Levine DW, Chan H (2008) Autonomic multi-agent management of power and performance in data centers, pp 107–114
10. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
11. Dolz MF, Fernández JC, Iserte S, Mayo R, Quintana-Ortí ES, Cotallo ME, Díaz G (2011) Energy saving cluster experience in CETA-CIEMAT. In: 5th Iberian GRID infrastructure conference, Santander
12. Matthew E, Mike A, Felipe FC, da Fonseca M, Para GV, Michael S (2011) Smarter data centers: achieving greater efficiency. Technical report, IBM Redpaper
13. EIA. Electric Power Monthly—Energy Information Administration
14. Elnozahy EN, Kistler M, Rajamony R (2002) Energy-efficient server clusters, pp 179–197
15. Emerson Network Power (2009) Energy logic: reducing data center energy consumption by creating savings that cascade across systems. Technical report
16. Eurostat (2013) Electricity and natural gas price statistics—statistics explained
17. Freeh VW, Lowenthal DK (2005) Using multiple energy gears in MPI programs on a power-scalable cluster. In: Proceedings of the tenth ACM SIGPLAN symposium on principles and practice of parallel programming—PPoPP '05. ACM Press, New York, p 164
18. Freeh VW, Lowenthal DK, Pan F, Kappiah N, Springer R, Rountree BL, Femal ME (2007) Analyzing the energy–time trade-off in high-performance computing applications. IEEE Trans Parallel Distrib Syst 18(6):835–848
19. Garcia DF, Entrialgo J, Garcia J, Garcia M (2010) A self-managing strategy for balancing response time and power consumption in heterogeneous server clusters. In: 2010 international conference on electronics and information engineering, vol 1. IEEE, New York, pp V1–537-V1-541
20. Gartner (2007) Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions
21. Ge R, Feng X, Feng W, Cameron KW (2007) CPU MISER: a performance-directed, run-time system for power-aware clusters. In: 2007 international conference on parallel processing (ICPP 2007). IEEE, New York, pp 18–18
22. Ruud H (2011) The blue gene/Q compute chip. Technical report, IBM Corporation
23. Hsu CH, Feng W (2005) A power-aware run-time system for high-performance computing. In: ACM/IEEE SC 2005 conference (SC'05). IEEE, New York, pp 1–1
24. Hsu CH, Kremer U (2003) The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction. ACM SIGPLAN Not 38(5):38
25. Huang S, Feng W (2009) Energy-efficient cluster computing via accurate workload characterization. In: 2009 9th IEEE/ACM international symposium on cluster computing and the grid. IEEE, New York, pp 68–75
26. IBM Systems and Technology Group (2011) IBM system blue gene/Q—DCD12345USEN.pdf. Technical report, IBM, Somers, NY
27. Ishibuchi H, Nakashima T, Nii M (2004) Classification and modeling with linguistic information granules: advanced approaches to linguistic data mining. Adv Inf Process
28. Lang W, Patel JM, Naughton JF (2010) On energy management, load balancing and replication. ACM SIGMOD Record 38(4):35
29. Li D, Nikolopoulos DS, Cameron K, de Supinski BR, Schulz M (2010) Power-aware MPI task aggregation prediction for high-end computing systems. In: 2010 IEEE international symposium on parallel & distributed processing (IPDPS). IEEE, New York, pp 1–12
30. Lim M, Freeh V, Lowenthal D (2006) Adaptive, transparent frequency and voltage scaling of communication phases in MPI programs. In: ACM/IEEE SC 2006 conference (SC'06). IEEE, New York, p 14
31. Llamas RM, Garcia DF, Entrialgo J (2012) A technique for self-optimizing scalable and dependable server clusters under QoS constraints. In: 2012 IEEE 11th international symposium on network computing and applications. IEEE, New York, pp 61–66
32. Pinheiro E, Bianchini R, Carrera EV, Heath T (2001) Load balancing and unbalancing for power and performance in cluster-based systems. In: Workshop on compilers and operating systems for low power, vol 180. Barcelona, Spain, pp 182–195
33. Schubert S, Kostic D, Zwaenepoel W, Shin KG (2012) Profiling software for energy consumption. In: 2012 IEEE international conference on green computing and communications. IEEE, New York, pp 515–522

34. Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern, SMC-15(1):116–132
35. Tang G, Gupta Q, Varsamopoulos SKS (2008) Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Trans Parallel Distrib Syst 19(11):1458–1472
36. Unit The Economist Intelligence (2007) IT and the environment A new item on the CIOGs agenda? Technical report, The Economist
37. U.S. Environmental Protection Agency (2007) Report to congress on server and data center energy efficiency public law. Technical report, ENERGY STAR Program, pp 109–431
38. Lassonde W, Khan SU, Min-Allah N, Madani SA, Li J, Zhang L, Wang L, Ghani N, Kolodziej J, Li H, Zomaya AY, Xu CZ, Balaji P, Vishnu A, Pinel F, Pecero JE, Kliazovich D, Bouvry P (2011) An overview of energy efficiency techniques in cluster computing systems. Cluster Comput 16(1):3–15
39. Xian C, Lu YH, Li Z (2007) A programming environment with runtime energy characterization for energy-aware applications. In: Proceedings of the 2007 international symposium on low power electronics and design—ISLPED '07. ACM Press, New York, pp 141–146
40. Xue Z, Dong X, Ma S, Fan S, Mei Y (2007) An energy-efficient management mechanism for large-scale server clusters. In: The 2nd IEEE Asia-Pacific service computing conference (APSCC 2007). IEEE, New York, pp 509–516
41. Yeo CS, Buyya R, Pourreza H, Rasit Eskicioglu M, Graham P, Pourreza P, Sommers F (2006) Cluster computing: high-performance, high-availability, and high-throughput processing on a network of computers. In: Zomaya AY (ed) Handbook of nature-inspired and innovative computing. Springer, Berlin, pp 521–551
42. Zong Z, Nijim M, Manzanares A, Qin X (2007) Energy efficient scheduling for parallel applications on mobile clusters. Cluster Comput 11(1):91–113
43. Zong Z, Ruan X, Manzanares A, Bellam K, Qin X (2010) Improving energy-efficiency of computational grids via scheduling. In: Antonopoulos N, Exarchakos G, Li M, Liotta A (eds) Handbook of research on P2P and grid systems for service-oriented computing, chap. 22. IGI Global, Hershey

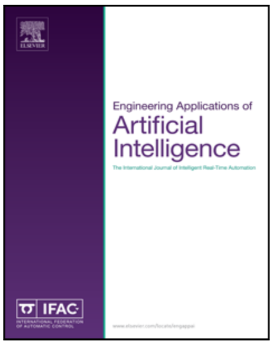
Título

Leveraging a predictive model of the workload for intelligent slot allocation schemes in energy-efficient HPC clusters

Autores

Alberto Cocaña-Fernández, Luciano Sánchez and José Ranilla

Journal



***Engineering Applications of Artificial Intelligence**, Volumen 48, Páginas 95–105, 2016*

DOI: 10.1016/j.engappai.2015.10.003

Ranking

Factor de impacto (JCR 2015): 2,368

Áreas:

- *Engineering, Electrical & Electronic*, 46/257 (Cuartil Q1)
- *Engineering, Multidisciplinary*, 10/85 (Cuartil Q1)
- *Computer Science, Artificial Intelligence*, 32/130 (Cuartil Q1)

Engineering Applications of Artificial Intelligence 48 (2016) 95–105

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Leveraging a predictive model of the workload for intelligent slot allocation schemes in energy-efficient HPC clusters

Alberto Cocaña-Fernández*, Luciano Sánchez, José Ranilla

Departamento de Informática, Universidad de Oviedo, Gijón, Spain

ARTICLE INFO

Article history:
Received 13 January 2015
Received in revised form
30 August 2015
Accepted 12 October 2015
Available online 19 November 2015

Keywords:
Energy-efficient cluster computing
Multi-criteria decision making
Evolutionary algorithms
Distal learning

ABSTRACT

A proactive mechanism to learn an efficient strategy for adaptive resource clusters is proposed. In contrast to reactive techniques, that rescale the cluster to fit the past load, a predictive strategy is adopted. The cluster incoming workload is forecasted and an optimization problem is defined whose solution is the optimal action according to a utility function. Genetic-based machine learning techniques are used, including multi-objective evolutionary algorithms under the distal supervised learning setup. Experimental evaluations show that the proactive system presented in this work improves either the energetic efficiency or the number of reconfigurations of previous approaches without a loss in the quality of service. Depending on the predictability of the workload, in real world cluster scenarios additional energy savings of up to approximately 40% were measured over the best previous approach, with a $2\times$ factor increment in the number of reconfigurations.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

According to Delforge and Whitney (2014), U.S. data centres consumed 91 billion kilowatt-hours of electricity in 2013. This consumption is projected to increase to roughly 140 billion kilowatt-hours by 2020, costing $13 billion annually. The carbon footprint is also very significant and accounts for nearly 100 million metric tons of carbon pollution per year, a figure equivalent to aviation (Gartner, 2007). In particular, energy efficient operation in High Performance Computing (HPC) Clusters is a challenging issue. HPC Clusters are the main architecture for supercomputers[1] due to the high performance of commodity microprocessors and networks, to the standard tools for high performance distributed computing, and to the lower price/performance ratio (Yeo et al., 2006). Because of the ubiquity of HPC clusters, there are powerful economical and environmental incentives for developing new techniques to reduce their electrical consumption. Furthermore, improvements in consumption result in lower heat dissipation. As a side effect, the reliability of the cluster is improved and the consumption of auxiliary devices, such as power supply units, power distribution, cooling, lighting and building switchgear, is lessened as well (Emerson Network Power, 2009).

Static approaches to the problem are focused on the development of new hardware with a lower consumption, for instance low-power CPUs such as the IBM PowerPC A2 of IBM Blue Gene/Q (Haring et al., 2012; IBM Systems and Technology Group, 2011), GPUs or Intel Xeon Phi coprocessors. Dynamic approaches, on the other hand, reshape clusters to match the existing load, downspeeding or shutting down unneeded resources (Valentini et al., 2011). For example, the Dynamic Voltage and Frequency Scaling (DVFS) technique reduces CPU voltage and frequency when the CPU is idle or under-used, as shown in Hsu and Kremer (2003), Hsu and Feng (2005), Freeh et al. (2007), Lim et al. (2006), Cheng and Zeng (2011), Ge et al. (2007), Huang and Feng (2009), and Chetsa et al. (2012). In this respect, there are software frameworks that can take advantage of different cluster energy-saving features when developing energy-efficient applications, such as Alonso et al. (2012), Schubert et al. (2012), Freeh and Lowenthal (2005), Li et al. (2010), and Xian et al. (2007). Lastly, at an intermediate level between low-power hardware and energy-conscious software, there are job schedulers (Zong et al., 2007, 2010) and thermal-aware methods (Bash et al., 2007; Tang et al., 2008) that can be used in combination with software that was not designed with energy savings in mind. This intermediate-level software can scale down and up the cluster in response to changing conditions. In particular in the so-called adaptive resource clusters, the compute nodes are switched on and off on the basis of requested, idle and available resources. The ultimate purpose of this technique is to switch off all idle nodes, however the decision algorithm is not

* Corresponding author.
E-mail addresses: cocanaalberto@gmail.com (A. Cocaña-Fernández), luciano@uniovi.es (L. Sánchez), ranilla@uniovi.es (J. Ranilla).

[1] June 2014|TOP500 Supercomputer Sites, http://www.top500.org/lists/2014/06/

http://dx.doi.org/10.1016/j.engappai.2015.10.003
0952-1976/© 2015 Elsevier Ltd. All rights reserved.

trivial and has competing restrictions. For instance, a high number of reconfigurations are not wanted as it might hamper the reliability of the cluster, thus a node should not be shut down if it is going to be needed shortly after.

Many different variants of the adaptive resource cluster technique exist. It was introduced in Pinheiro et al. (2001) for Load-Balancing clusters, and it was also used in Das et al. (2008), Elnozahy et al. (2002), Berral et al. (2010), Lang et al. (2010), Garcia et al. (2010), and Llamas et al. (2012) and in VMware vSphere[2] and Citrix XenServer hypervisors.[3] It has also been applied to HPC clusters in Alvarruiz et al. (2012), Dolz et al. (2011) and Xue et al. (2007). In these last works, a Knowledge based System (KBS) is used to determine the resources (e.g. the number of compute node slots) that are needed at every moment.

In all cases, the KBSs depend on certain heuristic configuration parameters such as the time of inactivity to shutdown nodes. The heuristic component of these systems is of a high importance as it governs the balance between energy savings and the number of reconfigurations. However, these parameters cannot be preset: the KBS and the parameters on which it depends must be hand tuned for the expected workload. In previous works (Cocaña Fernández et al., 2014a) we proposed to learn the heuristic parameters defined in Dolz et al. (2011) by means of a multiobjective evolutionary algorithm in a machine learning approach. The resulting system works with both OGE/SGE and PBS/TORQUE Resource Management Systems (RMS), has a good compliance with administrator preferences in all QoS, and yields reasonable energy savings and node reconfigurations.

After this, we further worked on improving the results of the KBS as the decision making mechanism of the cluster management system. In particular, we presented in Cocaña Fernández et al. (2014b) a Hybrid Genetic Fuzzy System (HGFS) that seeks the optimal rule base for the KBS by eliciting the linguistic definition of part of the aforementioned knowledge base from data, making it depend on the cluster behaviour, and having it combined with expert rules to produce a new system whose results proofed better in both linguistic interpretability and efficiency than those achieved previously in Cocaña Fernández et al. (2014a).

However, one might still wonder whether a purely reactive strategy such as the one used in both of our previous works is the best for all scenarios or, on the contrary, a significant improvement could be obtained from a predictive management approach profiting from a higher degree of adaptability to changing workload conditions. To answer this question, we present a new decision-making mechanism for the cluster management system which consists in a predictive controller based on the framework proposed in Abdelwahed et al. (2004, 2009) whose utility function is a fuzzy model learned by means of a genetic-based machine leaning (GBML) multiobjective evolutionary algorithm (MOEA) in a distal supervised learning approach (Jordan and Rumelhart, 1992). This new mechanism, along with the KBS parameter learning algorithm proposed in Cocaña Fernández et al. (2014a) and the HGFS in Cocaña Fernández et al. (2014b), is then tested in different scenarios to give a sound answer on which strategy proofs better in each case.

The remainder of the paper is as follows. Section 2 explains succinctly the architecture of the cluster management system. Section 3 summarizes the aforementioned reactive strategy. Section 4 explains the predictive controller. Section 5 shows the experimental results. Section 6 concludes the paper and discusses the future work.

[2] VMware Distributed Power Management Concepts and Use, http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf

[3] Citrix XenServer – Efficient Server Virtualization Software, http://www.citrix.com/products/xenserver/overview.html

2. Architecture

As mentioned in Cocaña Fernández et al. (2014a,b, 2015), the solution proposed consists in a service and an administration dashboard, coupled with a Database Management System, and deployed over an HPC cluster running a Resource Management System such as OGE/SGE or PBS/TORQUE.

An overview of the system is depicted in Fig. 1. The system status is monitored by the EEClusterd service, which uses a knowledge-based Decision System to perform node reconfigurations through the Power Management module. This last module switches on/off the nodes appointed by the KBS with Ethernet cards or IPMI cards (Intelligent Platform Management Interface). The EEClusterd service collects and keeps records of the RMS and of every compute node. RMS data comprises the cluster parallel environments (OGE/SGE), queues, hosts, users, and completed, queued and running jobs. Further information on the EECluster tool architecture, deployment and use can be found in Cocaña Fernández et al. (2015).

As can be seen, the decision-making mechanism is the key component of the system, since it is the one responsible for rescaling the compute nodes to match the cluster workload. Find the optimal amount of nodes that must be on at every moment given a set of preferences is no trivial problem. Because of this, both reactive and predictive strategies have been experimented with in order to achieve the best results in each scenario.

The reactive strategy consists in the reconfiguration of nodes to match the cluster status whenever the decision-making mechanism is triggered, having this status measured in terms of queued jobs, average waiting times and node idle times. Following this strategy are the KBS and HGFS explained in Section 3.

The predictive strategy consists in forecasting the cluster incoming workload and then solve an optimization problem to choose the optimal action among the permissible ones given the cluster current status and according to a machine-learned utility function. The mechanism implementing this strategy is explained in Section 4.

Observe that all of the decision-making algorithms referenced up to this point control how many slots are powered or switched off, but none of them identify the precise nodes that must be reconfigured. These nodes are selected according to its past efficiency and how long has passed since its state was changed. The rationale is to keep the most efficient nodes running for an extended time. In addition to this, nodes that failed to comply with the last order are marked, and those with earlier timestamps are preferred. The system iterates over the potentially malfunctioning nodes thus the probability of finding a repaired node is increased.

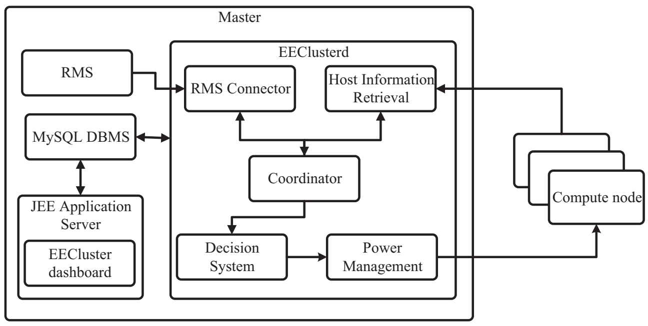


Fig. 1. System components overview.

3. Reactive decision-making through parametric KBS and hybrid genetic fuzzy systems

3.1. KBS parameter learning

The first implementation of the reactive strategy is a KBS comprised exclusively of a set of expert hand-written rules that were proposed in Dolz et al. (2011), which rely on a set of configuration parameters whose values rule the KBS operation.

These rules are as follows:

- **if** $s_{running}+s_{starting}<s_{min}$ **then** power on $(s_{min}-(s_{running}+s_{starting}))$ slots
- **if** $t_{avg}>t_{max}$ or $n_{queued}>n_{max}$ **then** power on 1 slot
- **If** $t_{avg}<t_{min}$ or $n_{queued}<n_{min}$ **then** power off 1 slot
- **for each** h in *hosts* **do**
  if $\text{idle}_h>\text{idle}_{max}$ **then** power off host h

The number of slots running and starting at a given time is respectively named $s_{running}$ and $s_{starting}$. The minimum number of slots required to run each of the queued jobs is s_{min}. The number of total slots (running and powered off) in the cluster is s_{total}. The average waiting time for the queued jobs is t_{avg}. The maximum and minimum average waiting time for the queued jobs are t_{max} and t_{min} respectively. The number of queued jobs is n_{queued}. The maximum and minimum number of queued jobs before an action is performed are n_{max} and n_{min} respectively. Finally, the time that the host h has been at idle state is called idle_h and the maximum time a host can be at idle state is idle_{max}.

If the linguistic structure of the Knowledge-based System mentioned before is not altered, each decision system can be described by the following five parameters:

$$(t_{min},t_{max},n_{min},n_{max},\text{idle}_{max}) \tag{1}$$

This Knowledge-based System can potentially adapt to any desired working mode for the cluster. However, this ability to adapt comes with the problem of actually finding the right set of values to match the desired working mode. Because of this, in Cocaña Fernández et al. (2014a), multiobjective evolutionary algorithms (MOEAs) were used to find the parameters defining the KBS, by optimizing a fitness function consisting in three conflicting criteria: the quality of service, the energy saved and the number of node reconfigurations.

3.2. Hybrid GFS

To further improve the results achieved using the previous KBS, in Cocaña Fernández et al. (2014b), was introduced a Hybrid Genetic Fuzzy System (HGFS) that is learned from data and replaces the expert-defined knowledge base with a hybrid knowledge base combining some of the expert non-fuzzy rules taken from Dolz et al. (2011) and the fuzzy rules in the form of a zero-order Takagi–Sugeno–Kang (TSK) fuzzy model (Ishibuchi et al., 2004; Takagi and Sugeno, 1985) that were learnt.

The structure of this hybrid system and can be expressed as follows:

if $s_{running}+s_{starting}<s_{min}$ **then** power on $(s_{min}-(s_{running}+s_{starting}))$ slots
if $t_{avg}>t_{max}$ or $n_{queued}>n_{max}$ **then** power on 1 slot
if $t_{avg}<t_{min}$ or $n_{queued}<n_{min}$ **then** power off 1 slot

for each h in *hosts* **do**
if idle_h is $\tilde{T}_1$ **then** off $=w_1$
if idle_h is $\tilde{T}_2$ **then** off $=w_2$
if ⋯ **then** ⋯
if idle_h is $\tilde{T}_N$ **then** off $=w_N$

All the variables defined in Section 3.1 are needed plus a few additional ones specific to the definition of the fuzzy linguistic terms. These are:

- N triangular fuzzy subsets $\tilde{T}_1,\dots,\tilde{T}_N$ of the domain of the variables idle_h. Each triangular fuzzy number $\tilde{T}_r$ for $r=1\dots,N$ depends on three parameters

  $(\text{left}_r,\text{center}_r,\text{right}_r)$.

  In this particular case the fuzzy sets are arranged to form a fuzzy partition (Ishibuchi et al., 2004), thus each fuzzy membership shares two parameters with the preceding element: $\text{left}_r=\text{center}_{r-1}$ and $\text{center}_r=\text{right}_{r-1}$. Therefore, the whole fuzzy partition depends on $N+2$ parameters:

  $(\text{left}_1,\text{center}_1,\text{right}_1,\text{right}_2,\dots,\text{right}_N)$.

  Recall that idle_h is the time that the h-th host has been at idle state.
- The terms w_r, with $w_r\in[0,1]$, for $r=1\dots,N$ that can be thought of as the degrees of truth of the asserts "if the idle time of the h-th node is $\tilde{T}_r$ then the node must be switched off".

The total number of nodes that are powered off is given by the following expression:

$$\text{Powered off nodes}=\left\lfloor\sum_{h=1}^{c}\text{TSKoutput}(\text{idle}_h)\right\rfloor \tag{2}$$

where the intermediate function $\text{TSKoutput}(t)$ is the output of the zero-order TSK fuzzy model formed by the N fuzzy rules included in the KBS, and is defined as follows:

$$\text{TSKoutput}(t)=\frac{\sum_{r=1}^{N}\tilde{T}_r(t)\cdot w_r}{\sum_{r=1}^{N}\tilde{T}_r(t)}. \tag{3}$$

Observe that a particular instance of the hybrid GFS can, therefore, be expressed as a tuple

$$(t_{min},t_{max},n_{min},n_{max},\text{left}_1,\text{center}_1,\text{right}_1,\text{right}_2,\dots,\text{right}_N,w_1,\dots,w_N).$$

Apart from the extra $2N+2$ parameters in the definition of an individual, the same evolutionary algorithm used in Cocaña Fernández et al. (2014a) is valid for solving the hybrid problem.

4. Proactive model

The Knowledge-based Systems introduced before both share the inner limitation of the reactive strategy: decisions of node reconfigurations are just based upon a series of values such as the number of queued jobs or the average waiting time. These values only reflect a static snapshot of the cluster status, thus disallowing these mechanisms from assessing the future consequences of each decision that make, what limits their capabilities of adaptation to volatile, although potentially stationary, scenarios with notably changing patterns of cluster activity, something rather expectable in many HPC clusters. In other words, these KBS have their operation based entirely on a set of parameters whose values were learnt upon a given cluster workload, and are not based on the actual situation of the cluster and the incoming workloads whenever the decision is made.

To improve the system results in volatile cluster scenarios, a proactive model based on the application of predictive techniques for computing systems introduced in Abdelwahed et al. (2004, 2009) is used as an alternative decision-making mechanism. It is

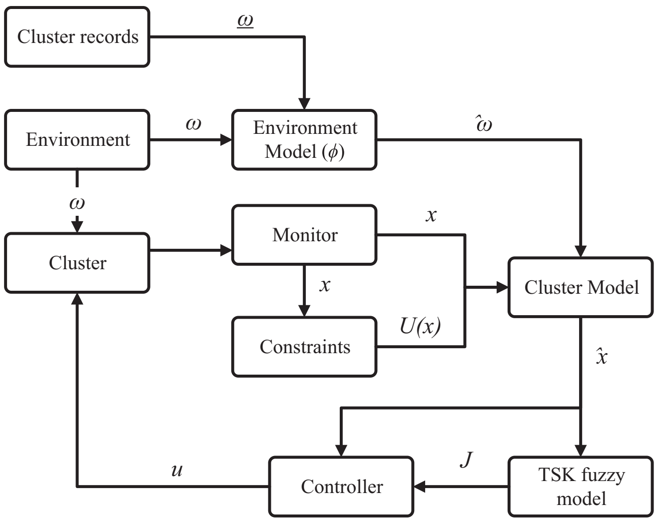


Fig. 2. Predictive controller components overview.

noteworthy that, although examples of the application of these predictive techniques can be found (see for instance Roy et al., 2011 and Bhat et al., 2006), these works have not yet been applied to the High Performance Computing (HPC) clusters addressed in this paper, which are fundamentally different from Load Balancing clusters or Grid environments in their architectures, number of concurrent requests, nature and number of resources requested and request running times.

This model transforms the cluster reconfiguration into an optimization problem of the cluster behaviour over a future temporal horizon, having this forecast by a model of both the cluster and the environment. To do so, time is split at regular intervals of $t_{interval}$ units of time, and at the beginning of each interval the control algorithm is executed resolving the optimization problem.

Fig. 2 represents the components of the controller, and the predictive control algorithm is showed in Algorithm 1.

Algorithm 1. Predictive control for an interval beginning at time k.

input: $x(k), \underline{\omega}(k-1, r)$
1: $\hat{\omega}(k) \leftarrow \phi_{k+1}(\underline{\omega}(k-1), r)$
2: **for each** $u \in U(x)$ **do**
3: $\hat{x}(k+1) \leftarrow f(x(k), u, \hat{\omega}(k))$
4: Compute utility of u based on $J(\hat{x}(k+1), u)$
5: **end for each**
output: $\underset{u}{\operatorname{argmax}}\{J(\hat{x}(k+1), u)\}$

Let k be the beginning of the current control interval, let $k+1$ be the end of the current interval and the beginning of the next one, let $x(k)$ be the cluster state and time k, let $u \in U(x)$ be a control action within the permissible actions at state x, let $w(k)$ be the actual environment and $\hat{\omega}(k)$ the forecast environment at time k, let ϕ_k be the environment forecasting model built at time k, let J be the utility function. Lastly, let f be the cluster model that runs a simulation to measure the expected outcome of taking an action u in a state $x(k)$ and with an incoming workload $\hat{\omega}(k)$ over the temporal horizon. Then, the chosen control action u at time k is the one of the allowed ones $U(x)$ for the current state $x(k)$ that gets the highest value in the utility function $J(\hat{x}(k+1), u)$, having the expected outcome $\hat{x}(k+1)$ of control action u computed by means of a simulation in the cluster model f (line 3 in Algorithm 1).

A given control action u represents the number of compute node slots that will be running after the cluster reconfiguration is done at time k. Due to obvious operating constraints, a control action cannot power on more nodes than the ones physically existing and available in the cluster. Also, since a running job cannot be halted, nodes that are currently executing jobs must never be shutdown. Because of these constraints, the set of control actions for a given state x, denoted by $U(x)$, represents the numbers of nodes that can be powered on as a result of a control action.

The rest of the section describes the environment forecasting, the utility function and the controller learning algorithm.

4.1. Environment forecasting

The cluster environment represents every external input to the system from the users in the form of jobs, and that cannot be controlled. This environment is estimated by generating synthetic workloads through the Monte Carlo simulation method using a forecasting model composed of a set of adjusted probability distributions.

Let $\hat{\omega}(k)$ be the estimated incoming workload during the control interval beginning at time k, let $\underline{\omega}(k-1, r)$ be the recorded workload of the r previous control intervals $\{\omega(k-1), \ldots, \omega(k-r-1)\}$, and let ϕ_k be the forecasting model adjusted at time k. Then, the forecast incoming workload at time k is

$$\hat{\omega}(k) = \phi_k(\underline{\omega}(k-1, r)) \tag{4}$$

If the actual environment $\omega(k)$ of the time interval that begins at time k represents the workload submitted by the cluster users in the form of n jobs, where the j-th job $(j = 1 \ldots n)$ arrives $tarr_j$ seconds after the previous job, requests s_j slots and has a runtime of r_j seconds, then, the estimated environment $\hat{\omega}(k)$ represents $\hat{n}$ jobs where the j-th job $(j = 1 \ldots \hat{n})$ arrives $\hat{tarr}_j$ seconds after the previous job, requests $\hat{s}_j$ slots and has a runtime of $\hat{r}_j$ seconds. The forecasting model ϕ for this environment is formed by three models ϕ^{tarr}, ϕ^{s} and ϕ^{r}, which generate the estimated values for the arrival times, requested slots and run times of the $\hat{n}$ jobs, having this number of jobs depending on the size of the temporal horizon and the arrival time of the last job.

The values of each of these forecasting models are generated following an adjusted probability distribution. For example, if the run times of the jobs are supposed to be exponentially distributed with a rate parameter λ, then the model ϕ^r adjusted at time k generates the values:

$$\phi^r(k) = \left\{ -\frac{1}{\lambda_k}\log(U_1), -\frac{1}{\lambda_k}\log(U_2) \cdots -\frac{1}{\lambda_k}\log(U_{\hat{n}}) \right\} \tag{5}$$

where $U_1 \ldots U_{\hat{n}}$ are nonzero uniform deviates.

4.2. Fuzzy utility function

As mentioned before, the control action chosen every time that the optimization problem is solved is ultimately determined by a utility function that establishes how good or bad each control action is. In other words, the utility function returns a real value which measures the expected degree of "utility" that a given control action u has for the overall cluster system. The higher the value, the more useful the action u is.

Before describing the way the utility function works, a few metrics must be defined. Since the input to this function is the expected future $x(k+1)$ resulting in executing a given action u at time k and with an expected workload $\hat{\omega}(k)$, which is done by running a simulation with the cluster model, the way the expected future state is assessed numerically so it can be used as an input to the utility function must be established first. To do so, three values are used: the number of nodes powered on at time $k+1$ as a result of the control action, the average relation between the waiting time of the jobs and their running time u_{wdr}, and the number of reconfigured nodes u_{rn}. These last two values are computed as follows.

The average waiting time/running time is used as a quality service metric, computed as

$$u_{wdr} = \ln\left(1 + \sum_{i=1}^{\hat{n}} \frac{\text{ton}_j - \text{tsch}_j}{\text{toff}_j - \text{ton}_j}\right) \tag{6}$$

where the $\hat{n}$ jobs are the forecast workload, with the j-th job ($j = 1\ldots\hat{n}$) arriving at time tsch_j, but starting its execution at time ton_j and stopping at time toff_j. Note that the logarithm is used as "squashing" function. As for the reconfigured nodes u_{rn}, it measures the degradation caused by the control action due to node thrashing by adding the number of nodes that are powered on and the number of nodes that are powered off:

$$u_{rn} = |\hat{x}(k+1)_{nodes} - x(k)_{nodes}| \tag{7}$$

The utility function as such is implemented using a zero-order Takagi–Sugeno–Kang (TSK) fuzzy model (Ishibuchi et al., 2004; Takagi and Sugeno, 1985), which uses u_{wdr}, $\hat{x}_{nodes}$ and u_{rn} as input values, is composed of Q rules, and whose structure can be expressed as follows:

if u_{wdr} is $\tilde{W}_1$ and $\hat{x}_{nodes}$ is $\tilde{N}_1$ and u_{rn} is $\tilde{R}_1$ **then** value $= w_1$
if u_{wdr} is $\tilde{W}_1$ and $\hat{x}_{nodes}$ is $\tilde{N}_1$ and u_{rn} is $\tilde{R}_2$ **then** value $= w_2$
if ⋯ **then** ⋯
if u_{wdr} is $\tilde{W}_1$ and $\hat{x}_{nodes}$ is $\tilde{N}_1$ and u_{rn} is $\tilde{R}_{N_3}$ **then** value $= w_{N_3}$
if u_{wdr} is $\tilde{W}_1$ and $\hat{x}_{nodes}$ is $\tilde{N}_2$ and u_{rn} is $\tilde{R}_1$ **then** value $= w_{N_3+1}$
if ⋯ **then** ⋯
If u_{wdr} is $\tilde{W}_1$ and $\hat{x}_{nodes}$ is $\tilde{N}_{N_2}$ and u_{rn} is $\tilde{R}_{N_3}$ **then** value $= w_{N_2 \times N_3}$
if u_{wdr} is $\tilde{W}_2$ and $\hat{x}_{nodes}$ is $\tilde{N}_1$ and u_{rn} is $\tilde{R}_1$ **then** value $= w_{N_2 \times N_3 + 1}$
if ⋯ **then** ⋯
If u_{wdr} is $\tilde{W}_{N_1}$ and $\hat{x}_{nodes}$ is $\tilde{N}_{N_2}$ and u_{rn} is $\tilde{R}_{N_3}$ **then** value $= w_Q$

where $\tilde{W}_1, \ldots, \tilde{W}_{N_1}$, $\tilde{N}_1, \ldots, \tilde{N}_{N_2}$ and $\tilde{R}_1, \ldots, \tilde{R}_{N_3}$ are triangular fuzzy sets forming a fuzzy partition (Ishibuchi et al., 2004) of the domain of the variables u_{wdr}, $\hat{x}_{nodes}$ and u_{rn} respectively. The partition $\tilde{W}$ has N_1 linguistic terms, $\tilde{N}$ has N_2 and $\tilde{R}$ has N_3. For instance, with $N_1 = 5$, $\tilde{T}_1$ may be "VERY LOW, "LOW", "MEDIUM", "HIGH", "VERY HIGH". The values $w_1, \ldots, w_Q$ are between 0.0 and 1.0, and represent the utility of the control action where 1.0 is the highest utility and 0.0 is the lowest. The intermediate output function is defined as follows:

$$\text{TSKOutput}(u_{wdr}, \hat{x}_{nodes}, u_{rn}) = \frac{\sum_{R_q \in S} \tilde{W}_q(u_{wdr}) \cdot \tilde{N}_q(\hat{x}_{nodes}) \cdot \tilde{R}_q(u_{rn}) \cdot w_q}{\sum_{R_q \in S} \tilde{W}_q(u_{wdr}) \cdot \tilde{N}_q(\hat{x}_{nodes}) \cdot \tilde{R}_q(u_{rn})} \tag{8}$$

Lastly, the output of the utility function can be expressed as:

$$J(\hat{x}, u) = \text{TSKoutput}(u_{wdr}, \hat{x}_{nodes}, u_{rn}) \tag{9}$$

4.3. *Learning algorithm*

As shown in the preceding sections, the predictive controller relies in a set of configuration parameters to determine its behaviour:

$$(t_{interval}, r, \tilde{W}_1, \ldots, \tilde{W}_{N_1}, \tilde{N}_1, \ldots, \tilde{N}_{N_2}, \tilde{R}_1, \ldots, \tilde{R}_{N_3}, w_1, \ldots, w_Q) \tag{10}$$

However, finding the right set of values to match the desired working mode for the cluster is not trivial. Leaving aside the fact that an exhaustive search is infeasible due to the large number of combinations, there is not either an optimal solution, since there are multiple conflicting objectives: the quality of service, the energy saved and the number of node reconfigurations. It is proposed that multiobjective evolutionary algorithms (MOEAs) are used to find the parameters of the predictive controller by optimizing a fitness function consisting in the three conflicting criteria. Specifically, the chosen MOEA is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002). For a given set of n jobs, where the j-th job ($j = 1\ldots n$) is scheduled to start at time tsch_j, but effectively starts at time ton_j and stops at time toff_j, the quality of service in a HPC cluster reflects the amount of time that each job has to wait before is assigned its requested resources. Once the job starts its execution, it will not be halted, thus we focus only on its waiting time. Because jobs do not last the same amount of time, their waiting in the queue is better expressed as a ratio considering their runtime. Finally, due to the potential existence of outlier values, the 90 percentile is used instead of average:

$$\text{QoS} = \min\left\{p : \left\| \left\{ j \in 1\ldots n : \frac{\text{ton}_j - \text{tsch}_j}{\text{toff}_j - \text{ton}_j} \le p \right\} \right\| > 0.9\, n \right\} \tag{11}$$

where $\|A\|$ is the cardinality of the set A.

The energy saved is measured as the sum of the amount of seconds that each node has been powered off. Let c be the number of nodes, let $\text{state}(i, t)$ be 1 if the i-th node ($i = 1\ldots c$) is powered at time t and 0 otherwise. Lastly, let the time scale be the lapse between $\text{tini} = \min_j\{\text{sch}_j\}$ and $\text{tend} = \max_j\{\text{toff}_j\}$. Then,

$$\text{Energy saved} = c \cdot (\text{tend} - \text{tini}) - \sum_{i=1}^{c} \int_{\text{tini}}^{\text{tend}} \text{state}(i, t)\, dt. \tag{12}$$

The node reconfigurations are the number of times that a node has been powered on or off. Let $\text{nd}(i)$ the number of discontinuities of the function $\text{state}(i, t)$ in the time interval $t \in (\text{tini}, \text{tend})$:

$$\text{Reconfigured nodes} = \sum_{i=1}^{c} \text{nd}(i) \tag{13}$$

Although MOEAs can address the problem of finding Pareto-optimal solutions, the learning problem has an additional complication: each solution found by the NSGA-II algorithm must be tested in a cluster environment to measure the results. This is known as the distal supervised learning problem (Jordan and Rumelhart, 1992), where the learning algorithm must control indirectly the cluster outcome (distal variables) through the instances of the predictive controller (proximal variables) having the outcome fitness as a feedback to guide the learning process. This can be seen in Fig. 3, where the NSGA-II learning algorithm produces predictive controller instances, which are then tested in a cluster simulator with a given workload to compute their fitness value. It is also remarked that in the provided results the membership functions $\tilde{W}_1, \ldots, \tilde{W}_{N_1}, \tilde{N}_1, \ldots, \tilde{N}_{N_2}, \tilde{R}_1, \ldots, \tilde{R}_{N_3}$ will not be

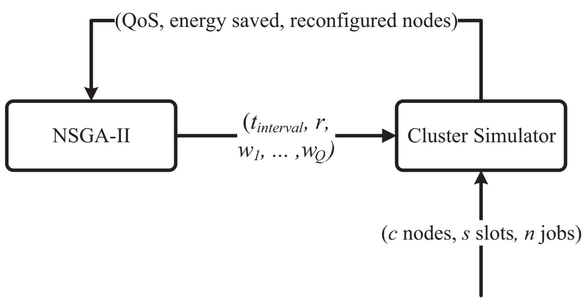


Fig. 3. Distal supervised learning of the predictive controller.

adjusted but a uniform partition is defined instead. This is not a fundamental limitation, since any change in the membership function of these sets could be compensated by the corresponding modification in the weights w_q.

5. Experimental results

Since HPC clusters may exhibit very different patterns of activity depending on their applications, a sound answer about which of the described decision-making mechanism works better involves thorough testing of different scenarios building together a significant representation of HPC clusters. To do so, four cluster scenarios have been defined in terms of synthetic cluster workloads characterized by the probability distributions of both job arrival rates and run times. The run times are distributed exponentially with rate $\lambda = 10^{-5}$ s in all scenarios, and the arrivals follow a Poisson process with the rate values shown in Table 1.

Essentially, scenario 1 depicts a cluster environment where jobs are submitted following an extremely regular arrival pattern where each hour of the year shares the same arrival rate. Scenario 2 also maintains regularity on a weekly basis, but distinguishes clearly between working hours, non-working hours and weekends, adjusting the arrival rate accordingly to each hour range. Scenario 3 adds a significant variation in arrival rates between consecutive weeks, and scenario 4 increases the degree of variation between weeks.

In addition, experiments with actual workloads from the Scientific Modelling Cluster of the University of Oviedo (CMS) spanning 22 months, with a total of 2907 jobs, were performed. This real world cluster, consisting of three independent computing clusters and five transversal queues using PBS as Resource Management System (RMS), can accurately show a very common activity pattern in most HPC clusters.

Similar to the experimentation in our previous work, a cluster simulator has been developed for both training and testing, so that every model can be evaluated in the three criteria of the fitness function.

The three decision-making mechanisms described in Section 3 and 4, along with mechanisms proposed by other authors, have been tested using this simulator in combination with the five workloads described before:

1. A basic model (labelled "Single rule") consisting in the allocation of as many compute node slots as are required to run all queued jobs, shutting down every idle node whenever the decision mechanism is triggered.
2. The rule model proposed in Dolz et al. (2011) (labelled as "EnergySaving ($t_{min}, t_{max}, n_{min}, n_{max}, \text{idle}_{max}$)") with multiple configurations and its parameters hand tuned by the administrator.
3. The rule model proposed in Alvarruiz et al. (2012), labelled as "CLUES ($\text{extra}_{slots}, \text{idle}_{max}$)", where extra_{slots} represents the number of extra slots that are powered on whenever additional slots are required to serve the current workload. Similar to the previous mechanism, multiple configurations were tested with their parameters hand tuned by the administrator.
4. The rule model proposed in Dolz et al. (2011) (labelled as "EnergySaving KBS NSGA-II") with the learning mechanism proposed in Cocaña Fernández et al. (2014a)
5. The hybrid GFS proposed in Cocaña Fernández et al., 2014b (labelled as "Hybrid GFS NSGA-II")
6. The predictive controller proposed in this paper, labelled as "Predictive controller (N_1, N_2, N_3)", with a different number of linguistic terms in each partition.

The holdout method was used for validation, with a 50–25–25% split in training, validation and test.

The administrator preferences for the experiment are based upon a lexicographic ordering of the three criteria: the administrator always seeks the best QoS and the amount of energy saved is used only to break ties in QoS. In turn, the number of reconfigurations also serves to break ties in QoS and energy saving.

The experiment results are shown in the following tables in terms of QoS, energy saved and node reconfigurations, and in the following charts in terms of the cluster simulation traces, which reveal the evolution over time of the aggregated requested slots by the jobs and the slots powered on by each mechanism. In particular, results for the scenario 1 are displayed in Table 2 and in Fig. 4, scenario 2 in Table 3 and in Fig. 5, scenario 3 in Table 4 and in Fig. 6, and scenario 4 in Table 5 and in Fig. 7. Lastly, results obtained for the CMS cluster recorded workloads are displayed in Table 6 and in Fig. 8.

It is remarked that the "QoS" column displays the 90 percentile (see Eq. (11)) thus a value of zero means that more than 90% of tasks were not delayed.

These results show that excessively simple mechanisms, such as the one labelled as "Single rule" do not perform well, negatively impacting cluster service quality, and neither can be tuned to suit administrator preferences due to the lack of configuration parameters. Similarly, mechanisms that despite relying on parameters to rule their function require these to be configured manually often perform poorly, impacting QoS and causing node thrashing. This is ultimately due to the complex task of finding the right set of values to comply with administrator preferences because of the large number of combinations.

Results also show that when the cluster workload exhibits a pattern of high arrival regularity, such as in scenario 1 and relatively in scenario 2, the decision-making mechanisms following a reactive strategy tend to achieve significantly better results in energy saving than the one following a predictive strategy. In this type of scenarios the distance between local workload peaks is very short, and so is the distance between peaks and valleys. The best approach is possibly to do minor cluster adjustments over short periods of time to save energy during these short-duration valleys. The reactive strategy is well suited to this situation. This also explains why the predictive controller barely reconfigures the

Table 1
Poisson process of job arrivals in each scenario.

Scenario	Day of week	Hour range	Week of year	λ value (s)
1	All	All	All	2×10^{-4}
	Mon–Fri	8:00–20:00	All	2×10^{-4}
2	Sat–Sun	8:00–20:00	All	2×10^{-5}
	Mon–Sun	20:00–8:00	All	10^{-5}
3	Mon– Fri	8:00–20:00	$w\,\%5=0$	10^{-4}
			$w\,\%\,5=1$	2×10^{-4}
			$w\,\%\,5=2$	5×10^{-4}
			$w\,\%\,5=3$	5×10^{-4}
			$w\,\%\,5=4$	2×10^{-4}
	Mon–Sun	20:00–8:00	All	2×10^{-5}
	Mon–Fri	8:00–20:00	All	10^{-5}
4	Mon– Fri	8:00–20:00	$w\,\%\,5=0$	10^{-4}
			$w\,\%\,5=1$	10^{-4}
			$w\,\%\,5=2$	5×10^{-4}
			$w\,\%\,5=3$	5×10^{-4}
			$w\,\%\,5=4$	10^{-4}
	Mon–Sun	20:00–8:00	All	2×10^{-5}
	Mon–Fri	8:00–20:00	All	10^{-5}

Table 2
Experiment results for the test set of the scenario 1.

Decision-making mechanism	Scenario 1 test set		
	QoS	Energy saved(s)	Reconfigurations
Single rule	4.49E−02	2.09E+08	5931
EnergySaving (0, 60, 0, 5, 3600)	2.74E−02	2.02E+08	3639
EnergySaving (0, 300, 0, 10, 3600)	4.24E−02	2.02E+08	3523
EnergySaving (0, 60, 0, 5, 7200)	2.14E−02	1.96E+08	2757
EnergySaving (0, 60, 0, 0, 14 400)	1.31E−02	1.88E+08	1968
CLUES (0, 3600)	2.72E−02	2.02E+08	3721
CLUES (0, 7200)	1.99E−02	1.96E+08	2801
CLUES (0, 14 400)	1.31E−02	1.88E+08	1962
CLUES (2, 3600)	1.77E−02	1.93E+08	7157
CLUES (2, 7200)	1.05E−02	1.87E+08	4609
CLUES (2, 14 400)	5.81E−03	1.79E+08	2827
EnergySaving KBS NSGA-II	0.00E+00	1.34E+08	473
Hybrid GFS NSGA-II	0.00E+00	1.49E+08	487
Predictive controller (3, 2, 2)	0.00E+00	9.32E+07	6
Predictive controller (3, 3, 3)	0.00E+00	9.31E+07	6
Predictive controller (4, 3, 3)	0.00E+00	9.32E+07	6
Predictive controller (5, 3, 3)	0.00E+00	9.24E+07	6
Predictive controller (5, 4, 4)	0.00E+00	9.24E+07	6

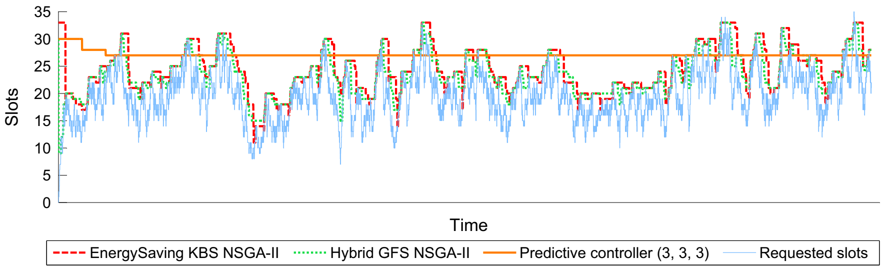


Fig. 4. Cluster simulation trace for the test set of the scenario 1.

Table 3
Experiment results for the test set of the scenario 2.

Decision-making mechanism	Scenario 2 test set		
	QoS	Energy saved(s)	Reconfigurations
Single rule	4.18E−02	4.03E+08	2390
EnergySaving (0, 60, 0, 5, 3600)	3.61E−02	3.99E+08	1962
EnergySaving (0, 300, 0, 10, 3600)	5.32E−02	3.99E+08	1944
EnergySaving (0, 60, 0, 5, 7200)	3.26E−02	3.96E+08	1754
EnergySaving (0, 60, 0, 0, 14 400)	2.48E−02	3.91E+08	1502
CLUES (0, 3600)	3.15E−02	3.99E+08	1972
CLUES (0, 7200)	2.93E−02	3.96E+08	1758
CLUES (0, 14 400)	2.51E−02	3.91E+08	1504
CLUES (2, 3600)	2.36E−02	3.95E+08	3900
CLUES (2, 7200)	1.60E−02	3.91E+08	2876
CLUES (2, 14 400)	1.20E−02	3.84E+08	2164
EnergySaving KBS NSGA-II	0.00E+00	2.92E+08	236
Hybrid GFS NSGA-II	0.00E+00	3.14E+08	218
Predictive controller (3, 2, 2)	0.00E+00	2.78E+08	18
Predictive controller (3, 3, 3)	0.00E+00	2.94E+08	20
Predictive controller (4, 3, 3)	0.00E+00	2.83E+08	18
Predictive controller (5, 3, 3)	0.00E+00	2.83E+08	18
Predictive controller (5, 4, 4)	0.00E+00	2.85E+08	27

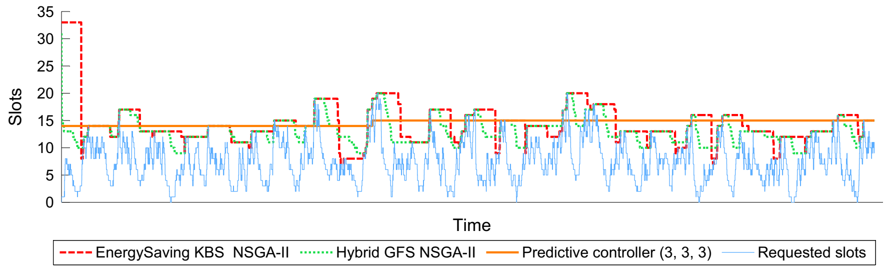


Fig. 5. Cluster simulation trace for the test set of the scenario 2.

Table 4
Experiment results for the test set of the scenario 3.

Decision-making mechanism	Scenario 3 test set		
	QoS	Energy saved(s)	Reconfigurations
Single rule	4.53E−02	3.49E+08	3394
EnergySaving (0, 60, 0, 5, 3600)	3.89E−02	3.44E+08	2586
EnergySaving (0, 300, 0, 10, 3600)	5.95E−02	3.44E+08	2552
EnergySaving (0, 60, 0, 5, 7200)	3.51E−02	3.40E+08	2325
EnergySaving (0, 60, 0, 0, 14 400)	2.88E−02	3.34E+08	2077
CLUES (0, 3600)	3.99E−02	3.44E+08	2626
CLUES (0, 7200)	3.51E−02	3.40E+08	2343
CLUES (0, 14 400)	3.16E−02	3.33E+08	2081
CLUES (2, 3600)	2.52E−02	3.39E+08	4442
CLUES (2, 7200)	1.91E−02	3.35E+08	3327
CLUES (2, 14 400)	1.47E−02	3.27E+08	2595
EnergySaving KBS NSGA-II	0.00E+00	1.19E+08	202
Hybrid GFS NSGA-II	0.00E+00	1.46E+08	194
Predictive controller (3, 2, 2)	0.00E+00	1.29E+08	175
Predictive controller (3, 3, 3)	0.00E+00	1.47E+08	337
Predictive controller (4, 3, 3)	0.00E+00	1.76E+08	338
Predictive controller (5, 3, 3)	0.00E+00	1.20E+08	479
Predictive controller (5, 4, 4)	0.00E+00	1.14E+08	155

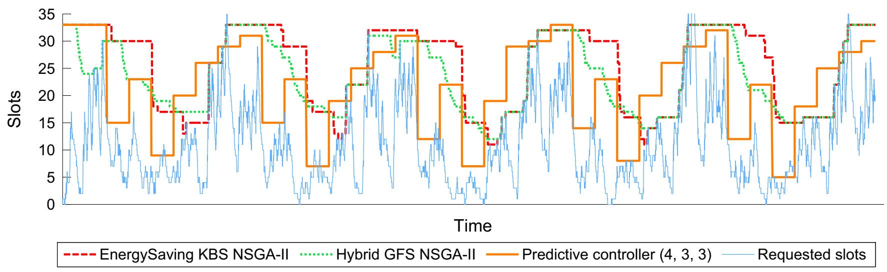


Fig. 6. Cluster simulation trace for the test set of the scenario 3.

cluster: the short duration of the valleys makes it very difficult for the controller to reconfigure the cluster over the rather long temporal horizons and save energy with no impact on QoS.

On the other hand, in cluster scenarios that exhibit a high degree of variation in the job arrival rates, such as scenarios 3 and 4, the predictive controller achieves better results that the reactive ones. Similar to the previous case, the key is both the duration of the workload valleys and the size of the local peaks, and how these values in scenarios 3 and 4 fit well the longer temporal horizons of the predictive controller. In particular, a comparison between results in scenarios 3 and 4 shows that the longer the valleys and the bigger the distance between peaks and valleys are, the better results are obtained by the predictive controller compared to the reactive ones. The reason for the bad performance of the reactive controllers in these scenarios, specially the expert-defined KBS, is that these mechanisms rely heavily on the time that the hosts have been at idle state to save energy. This is a problem here because, in order to assure good service quality whenever the load swiftly increases, the idle values are set too high to allow good energy saving or, otherwise, a very negative impact on QoS would occur as the workload grows. Regarding the results obtained in a real world cluster such as the CMS, these are very similar to the ones in scenarios 3 and 4, as could be expected due to the resemblance of the CMS activity pattern to these scenarios (see Fig. 8).

Lastly, it should also be noted that the Hybrid GFS always obtains better energy savings that the expert-defined KBS, what is achieved thanks to the higher flexibility of the fuzzy rule base regarding the host idle times.

Table 5
Experiment results for the test set of the scenario 4.

Decision-making mechanism	Scenario 4 test set		
	QoS	Energy saved(s)	Reconfigurations
Single rule	4.24E−02	3.66E+08	3031
EnergySaving (0, 60, 0, 5, 3600)	3.42E−02	3.62E+08	2275
EnergySaving (0, 300, 0, 10, 3600)	5.19E−02	3.62E+08	2255
EnergySaving (0, 60, 0, 5, 7200)	3.12E−02	3.58E+08	2022
EnergySaving (0, 60, 0, 0, 14 400)	2.61E−02	3.52E+08	1820
CLUES (0, 3600)	3.29E−02	3.61E+08	2289
CLUES (0, 7200)	2.92E−02	3.58E+08	2028
CLUES (0, 14 400)	2.81E−02	3.52E+08	1818
CLUES (2, 3600)	2.19E−02	3.57E+08	3923
CLUES (2, 7200)	1.65E−02	3.53E+08	3048
CLUES (2, 14 400)	1.24E−02	3.45E+08	2395
EnergySaving KBS NSGA-II	0.00E+00	3.61E+07	8
Hybrid GFS NSGA-II	0.00E+00	1.31E+08	173
Predictive controller (3, 2, 2)	0.00E+00	1.69E+08	414
Predictive controller (3, 3, 3)	0.00E+00	1.78E+08	393
Predictive controller (4, 3, 3)	0.00E+00	1.62E+08	252
Predictive controller (5, 3, 3)	0.00E+00	1.33E+08	273
Predictive controller (5, 4, 4)	0.00E+00	1.01E+08	52

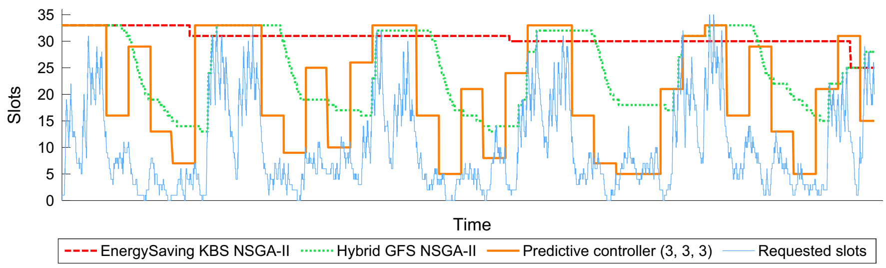


Fig. 7. Cluster simulation trace for the test set of the scenario 4.

Table 6
Experiment results for the test set of the CMS cluster workload records.

Decision-making mechanism	CMS cluster test set		
	QoS	Energy saved (s)	Reconfigurations
Single rule	8.02E+01	4.22E+08	2504
EnergySaving (0, 60, 0, 5, 3600)	4.86E+01	4.25E+08	1538
EnergySaving (0, 300, 0, 10, 3600)	7.74E+01	4.26E+08	1512
EnergySaving (0, 60, 0, 5, 7200)	2.23E+01	4.23E+08	1386
EnergySaving (0, 60, 0, 0, 14 400)	2.92E+00	4.19E+08	1216
CLUES (0, 3600)	3.43E+01	4.20E+08	2724
CLUES (0, 7200)	1.28E+01	4.16E+08	2308
CLUES (0, 14 400)	4.24E+00	4.11E+08	1828
CLUES (2, 3600)	2.60E+01	4.18E+08	3810
CLUES (2, 7200)	5.95E+00	4.14E+08	2942
CLUES (2, 14 400)	2.66E+00	4.07E+08	2366
EnergySaving KBS NSGA-II	0.00E+00	1.88E+08	47
Hybrid GFS NSGA-II	0.00E+00	2.41E+08	42
Predictive controller (3, 2, 2)	0.00E+00	2.01E+08	77
Predictive controller (3, 3, 3)	0.00E+00	3.00E+08	96
Predictive controller (4, 3, 3)	0.00E+00	2.78E+08	89
Predictive controller (5, 3, 3)	0.00E+00	2.72E+08	70
Predictive controller (5, 4, 4)	0.00E+00	2.03E+08	95

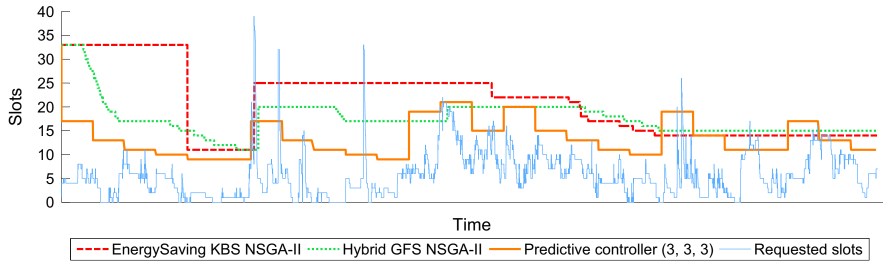


Fig. 8. Cluster simulation trace for the test set of the CMS cluster workload records.

6. Concluding remarks

Adaptive resource clusters are an efficient method for reducing electrical consumption, however this technique depends on a decision algorithm that has conflicting restrictions. Since a high number of reconfigurations are not desired, a node should not be shut down if it is going to be needed shortly after. Reactive techniques fulfill this objective by means of heuristics such as imposing delays before a node is stopped after a valley in the workload, or enforcing a minimum uptime for each functioning node. However, the best balance between consumption and reconfigurations is achieved with the proactive model described in this paper. The proposed strategy consists in forecasting the cluster incoming workload and then solving an optimization problem to choose the optimal action according to a fuzzy utility function. Specific genetic-based machine learning techniques were deployed that consist of mutiobjective evolutionary algorithms under the distal supervised learning setup. Empirical results prove that reactive systems tend to consume less energy in scenarios with a constant job arrival rate; nonetheless, the proactive system presented in this work achieves the highest energetic efficiency when the workload is not quite regular, as happens in most of practical scenarios.

Acknowledgements

This work has been partially supported by Ministerio de Economía y Competitividad from Spain/FEDER under Grants TEC2012-38142-C04-04, TEC2015-67387-C4-3-R and TIN2014-56967-R and by the Regional Ministry of the Principality of Asturias under Grant FC-15-GRUPIN14-073.

References

Abdelwahed, S., Bai, J., Su, R., Kandasamy, N., 2009. On the application of predictive control techniques for adaptive performance management of computing systems. IEEE Trans. Netw. Serv. Manag. 6 (December (4)), 212–225, URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5374030⟩.

Abdelwahed, S., Kandasamy, N., Neema, S., 2004. A control-based framework for self-managing distributed computing systems. In: Proceedings of the First ACM SIGSOFT Workshop on Self-Managed Systems—WOSS '04. ACM Press, New York, NY, USA, pp. 3–7, October. URL ⟨http://dl.acm.org/citation.cfm?id=1075405.1075406⟩.

Alonso, P., Badia, R.M., Labarta, J., Barreda, M., Dolz, M.F., Mayo, R., Quintana-Orti, E. S., Reyes, R., 2012. Tools for power-energy modelling and analysis of parallel scientific applications. In: 2012 41st International Conference on Parallel Processing. IEEE, Pittsburgh, PA, pp. 420–429, September. URL⟨http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6337603⟩.

Alvarruiz, F., de Alfonso, C., Caballer, M., Hernández, V., 2012. An energy manager for high performance computer clusters. In: 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications. IEEE, Leganés, Madrid, Spain, pp. 231–238, July. URL ⟨http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6280297⟩.

Bash, C., Forman, G., 2007. Cool Job Allocation: Measuring the Power Savings of Placing Jobs at Cooling-Efficient Locations in the Data Center. USENIX Association, pp. 29:1–29:6 June. URL ⟨http://dl.acm.org/citation.cfm?id=1364385.1364414⟩.

Berral, J.L., Goiri, I.N., Nou, R., Julià, F., Guitart, J., Gavaldà, R., Torres, J., 2010. Towards energy-aware scheduling in data centers using machine learning. In: Proceedings of the First International Conference on Energy-Efficient Computing and Networking—e-Energy '10. ACM Press, New York, NY, USA, p. 215, April. URL ⟨http://dl.acm.org/citation.cfm?id=1791314.1791349⟩.

Bhat, V., Parashar, M., Khandekar, M., Kandasamy, N., Klasky, S., 2006. A self-managing wide-area data streaming service using model-based online control. In: Proceedings of the Seventh IEEE/ACM International Conference on Grid Computing. GRID '06. IEEE Computer Society, Washington, DC, USA, pp. 176–183. http://dx.doi.org/10.1109/ICGRID.2006.311013.

Cheng, Y., Zeng, Y., 2011. Automatic energy status controlling with dynamic voltage scaling in power-aware high performance computing cluster. In: 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies. IEEE, Gwangju, Korea, pp. 412–416, October. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6118547⟩.

Chetsa, G.L.T., Lefrvre, L., Pierson, J.-M., Stolf, P., Da Costa, G., 2012. A runtime framework for energy efficient HPC systems without a priori knowledge of applications. In: 2012 IEEE 18th International Conference on Parallel and Distributed Systems. IEEE, Singapore, pp. 660–667, December. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6413638⟩.

Cocaña Fernández, A., Ranilla, J., Sánchez, L., 2014a. Energy-efficient allocation of computing node slots in HPC clusters through evolutionary multi-criteria decision making. In: Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2014, pp. 318–330.

Cocaña Fernández, A., Ranilla, J., Sánchez, L., 2014b. Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling. J. Supercomput. (October). URL ⟨http://link.springer.com/10.1007/s11227-014-1320-9⟩.

Cocaña Fernández, A., Sánchez, L., Ranilla, J., 2015. A software tool to efficiently manage the energy consumption of HPC clusters. In: 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2015).

Das, R., Kephart, J.O., Lefurgy, C., Tesauro, G., Levine, D.W., Chan, H., 2008. Autonomic Multi-Agent Management of Power and Performance in Data Centers, pp. 107–114, May. URL ⟨http://dl.acm.org/citation.cfm?id=1402795.1402816⟩.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (April(2)), 182–197. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=996017⟩.

Delforge, P., Whitney, J., 2014. Issue Paper: Data Center Efficiency Assessment Scaling up Energy Efficiency Across the Data Center Industry: Evaluating Key Drivers and Barriers. Technical Report, Natural Resources Defense Council (NRDC). URL ⟨http://www.nrdc.org/energy/files/data-center-efficiency-assessment-IP.pdf⟩.

Dolz, M.F., Fernández, J.C., Iserte, S., Mayo, R., Quintana-Ortí, E.S., Cotallo, M. E., Díaz, G., 2011. EnergySaving cluster experience in CETA-CIEMAT. In: The Fifth Iberian GRID Infrastructure Conference, Santander.

Elnozahy, E.N., Kistler, M., Rajamony, R., 2002. Energy-Efficient Server Clusters, pp. 179–197, February. URL ⟨http://dl.acm.org/citation.cfm?id=1766991.1767007⟩.

Emerson Network Power, 2009. Energy Logic: Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems. Technical Report. URL ⟨https://www.cisco.com/web/partners/downloads/765/other/Energy_Logic_Reducing_Data_Center_Energy_Consumption.pdf⟩.

Freeh, V.W., Lowenthal, D.K., 2005. Using multiple energy gears in MPI programs on a power-scalable cluster. In: Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming—PPoPP '05. ACM Press, New York, NY, USA, pp. 164–173, June. URL ⟨http://dl.acm.org/citation.cfm?id=1065944.1065967⟩.

Freeh, V.W., Lowenthal, D.K., Pan, F., Kappiah, N., Springer, R., Rountree, B.L., Femal, M.E., 2007. Analyzing the energy-time trade-off in high-performance computing applications. IEEE Trans. Parallel Distrib. Syst. 18 (June (6)), 835–848, URL ⟨http://dl.acm.org/citation.cfm?id=1263127.1263246⟩.

Garcia, D.F., Entrialgo, J., Garcia, J., Garcia, M., 2010. A self-managing strategy for balancing response time and power consumption in heterogeneous server clusters. In: 2010 International Conference on Electronics and Information Engineering, vol. 1. IEEE, pp. V1-537–V1-541, August. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5559691⟩.

Gartner, 2007. Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions. URL ⟨http://www.gartner.com/newsroom/id/503867⟩.

Ge, R., Feng, X., Feng, W.-c., Cameron, K.W., 2007. CPU MISER: a performance-directed, run-time system for power-aware clusters. In: 2007 International Conference on Parallel Processing (ICPP 2007). IEEE, p. 18, September. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4343825⟩.

Haring, R., Ohmacht, M., Fox, T., Gschwind, M., Satterfield, D., Sugavanam, K., Coteus, P., Heidelberger, P., Blumrich, M., Wisniewski, R., alan Gara, Chiu, G., Boyle, P., Chist, N., Kim, C., Mar. 2012. The IBM blue gene/Q compute chip. IEEE Micro 32 (2), 48–60. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6109225⟩.

Hsu, C.-H., Feng, W.-c., 2005. A power-aware run-time system for high-performance computing. In: ACM/IEEE SC 2005 Conference (SC'05). IEEE, Washington, DC, USA, p. 1. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1559953⟩.

Hsu, C.-H., Kremer, U., 2003. The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction. ACM SIGPLAN Not. 38 (May (5)), 38, URL ⟨http://dl.acm.org/citation.cfm?id=780822.781137⟩.

Huang, S., Feng, W., 2009. Energy-efficient cluster computing via accurate workload characterization. In: 2009 Ninth IEEE/ACM International Symposium on Cluster Computing and the Grid. IEEE, Shanghai, China, pp. 68–75. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5071856⟩.

IBM Systems and Technology Group, 2011. IBM System Blue Gene/Q. Technical Report, IBM, Somers, NY. URL ⟨http://www.fz-juelich.de/SharedDocs/Downloads/IAS/JSC/EN/JUQUEEN/BGQIBMDataSheet.pdf?__blob=publicationFile⟩.

Ishibuchi, H., Nakashima, T., Nii, M., 2004. Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing), November. URL ⟨http://dl.acm.org/citation.cfm?id=1044904⟩.

Jordan, M., Rumelhart, D.E., 1992. Forward models: supervised learning with a distal teacher. Cogn. Sci. 16 (September (3)), 307–354 URL ⟨http://www.sciencedirect.com/science/article/pii/036402139290036T⟩.

Lang, W., Patel, J.M., Naughton, J.F., 2010. On energy management, load balancing and replication. ACM SIGMOD Rec. 38 (June (4)), 35–42 URL ⟨http://dl.acm.org/citation.cfm?id=1815948.1815956⟩.

Li, D., Nikolopoulos, D.S., Cameron, K., de Supinski, B.R., Schulz, M., 2010. Power-aware MPI task aggregation prediction for high-end computing systems. In: 2010 IEEE International Symposium on Parallel& Distributed Processing (IPDPS). IEEE, Atlanta, GA, pp. 1–12. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5470464⟩.

Lim, M., Freeh, V., Lowenthal, D., 2006. Adaptive, transparent frequency and voltage scaling of communication phases in mpi programs. In: ACM/IEEE SC 2006 Conference (SC'06). IEEE, Tampa, FL, p. 14, November. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4090188⟩.

Llamas, R.M., Garcia, D.F., Entrialgo, J., 2012. A technique for self-optimizing scalable and dependable server clusters under qos constraints. In: 2012 IEEE 11th International Symposium on Network Computing and Applications. IEEE, pp. 61–66, August. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6299127⟩.

Pinheiro, E., Bianchini, R., Carrera, E. V., Heath, T., 2001. Load balancing and unbalancing for power and performance in cluster-based systems. In: Workshop on Compilers and Operating Systems for Low Power, vol. 180, Barcelona, Spain, pp. 182–195.

Roy, N., Dubey, A., Gokhale, A., 2011. Efficient autoscaling in the cloud using predictive models for workload forecasting. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 500–507, July.

Schubert, S., Kostic, D., Zwaenepoel, W., Shin, K.G., 2012. Profiling software for energy consumption. In: 2012 IEEE International Conference on Green Computing and Communications. IEEE, Besancon, France, pp. 515–522, November. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6468359⟩.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. 15 (January (1)), 116–132, URL ⟨http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6313399⟩.

Tang, Q., Gupta, S.K.S, Varsamopoulos, G., 2008. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Trans. Parallel Distrib. Syst. 19 (11), 1458–1472, November. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4553706⟩.

Valentini, G.L., Lassonde, W., Khan, S.U., Min-Allah, N., Madani, S.A., Li, J., Zhang, L., Wang, L., Ghani, N., Kolodziej, J., Li, H., Zomaya, A.Y., Xu, C.-Z., Balaji, P., Vishnu, A., Pinel, F., Pecero, J.E., Kliazovich, D., Bouvry, P., 2011. An overview of energy efficiency techniques in cluster computing systems. Clust. Comput. 16 (September (1)), 3–15 URL ⟨http://link.springer.com/10.1007/s10586-011-0171-x⟩.

Xian, C., Lu, Y.-H., Li, Z., 2007. A programming environment with runtime energy characterization for energy-aware applications. In: Proceedings of the 2007 international symposium on Low power electronics and design—ISLPED '07. ACM Press, New York, New York, USA, pp. 141–146, August. URL ⟨http://dl.acm.org/citation.cfm?id=1283780.1283811⟩.

Xue, Z., Dong, X., Ma, S., Fan, S., Mei, Y., 2007. An Energy-efficient management mechanism for large-scale server clusters. In: The Second IEEE Asia-Pacific Service Computing Conference (APSCC 2007). IEEE, Tsukuba Science City, Japan, pp. 509–516, December. URL ⟨http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4414502⟩.

Yeo, CheeShin, Buyya, Rajkumar, Pourreza, Hossein, Eskicioglu, Rasit, Graham, Peter, Sommers, F., 2006. Cluster computing: high-performance, high-availability, and high-throughput processing on a network of computers. In: Zomaya, A. (Ed.), Handbook of Nature-Inspired and Innovative Computing. Springer, USA, pp. 521–551. http://dx.doi.org/10.1007/0-387-27705-6_16.

Zong, Z., Nijim, M., Manzanares, A., Qin, X., 2007. Energy efficient scheduling for parallel applications on mobile clusters. Clust. Comput. 11 (November (1)), 91–113, URL ⟨http://dl.acm.org/citation.cfm?id=1349620.1349631⟩.

Zong, Z., Ruan, X., Manzanares, A., Bellam, K., Qin, X., 2010. Improving energy-efficiency of computational grids via scheduling. In: Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A. (Eds.), Handbook of Research on P2P and Grid Systems for Service-Oriented Computing. IGI Global (Chapter 22), January. URL ⟨http://www.igi-global.com/chapter/improving-energy-efficiency-computational-grids/40816/⟩.

Título

Improving the Eco-Efficiency of High Performance Computing Clusters Using EECluster

Autores

Alberto Cocaña-Fernández, Luciano Sánchez and José Ranilla

Journal

***Energies**, Volumen 9, Issue 3, Artículo 197, 2016*

DOI: 10.3390/en9030197

Ranking

Factor de impacto (JCR 2015): 2,077

Categorías:

Energy & Fuels, 43/88 (Cuartil Q2)

energies

Article

Improving the Eco-Efficiency of High Performance Computing Clusters Using EECluster

Alberto Cocaña-Fernández *, Luciano Sánchez † and José Ranilla †

Departamento de Informática, Universidad de Oviedo, 33204 Gijón, Spain; luciano@uniovi.es (L.S.); ranilla@uniovi.es (J.R.)
* Correspondence: cocanaalberto@gmail.com; Tel.: +34-985-182-121 or +34-985-182-130; Fax: +34-985-181-986
† These authors contributed equally to this work.

Academic Editor: Enrico Pontelli
Received: 31 December 2015; Accepted: 7 March 2016; Published: 14 March 2016

Abstract: As data and supercomputing centres increase their performance to improve service quality and target more ambitious challenges every day, their carbon footprint also continues to grow, and has already reached the magnitude of the aviation industry. Also, high power consumptions are building up to a remarkable bottleneck for the expansion of these infrastructures in economic terms due to the unavailability of sufficient energy sources. A substantial part of the problem is caused by current energy consumptions of High Performance Computing (HPC) clusters. To alleviate this situation, we present in this work EECluster, a tool that integrates with multiple open-source Resource Management Systems to significantly reduce the carbon footprint of clusters by improving their energy efficiency. EECluster implements a dynamic power management mechanism based on Computational Intelligence techniques by learning a set of rules through multi-criteria evolutionary algorithms. This approach enables cluster operators to find the optimal balance between a reduction in the cluster energy consumptions, service quality, and number of reconfigurations. Experimental studies using both synthetic and actual workloads from a real world cluster support the adoption of this tool to reduce the carbon footprint of HPC clusters.

Keywords: energy-efficient cluster computing; multi-criteria decision making; evolutionary algorithms

1. Introduction

Data and supercomputing centres are an essential element in modern society, as the vast majority of IT services are supported by them, profiting from the consolidation and centralization of high performance processors and networks. Targeting both academic and industrial communities, they provide the key infrastructure for web and application servers, e-commerce platforms, corporate databases, network storage, data mining, or the high performance computing resources required to address fundamental problems in science and engineering, to name a few examples.

The versatility of these computing facilities, coupled with the ever-increasing demand for IT services and their substantial power consumption, makes data and supercomputing centres one of the fastest-growing users of electricity in developed countries [1]. According to [1,2], electricity consumptions is the U.S. alone escalated from 61 billion kilowatt-hours (kWh) in 2006 to 91 billion kWh in 2013, and is projected to increase to 140 billion kWh in 2020. However, it should be noted that these large energy demands not only produce a significant economical impact for IT services providers [3,4], but also a carbon footprint equivalent to the aviation industry [5], which is expected to reach 340 million metric tons of CO_2 by 2020 worldwide [6].

Because of this, there is an unyielding need to improve the energy efficiency of data and supercomputing centres to reduce their environmental impact, operation costs and to improve the reliability of their components.

Abundant research has been conducted over the last years on the improvement of cluster computing efficiency, following multiple approaches that could be taxonomically classified in two categories: static and dynamic power management [7]. Static approaches focus on the development of low power CPUs seeking maximum efficiency, such as the IBM PowerPC A2 processors [8,9], as well as using GPUs or Intel Xeon Phi coprocessors as the main computing resources, given that this type of hardware is designed for an optimal FLOPS/watt relation instead of just raw performance. Dynamic techniques focus on the reconfiguration of the compute resources to best suit current workloads, saving energy when the cluster is underused. Among these techniques is the Dynamic Voltage and Frequency Scaling (DVFS) [10–17], which adjusts CPU frequency and voltage to match current demand, energy-efficient job schedulers that implement algorithms capable of reducing intercommunication-related power consumptions [18,19], thermal-aware methods which take into account the cooling efficiency of each area of the cluster [20,21], or software frameworks to assist in the development of energy-efficient applications [22–26]. Lastly, the adaptive resource cluster technique consists of the automatic reconfiguration of the cluster resources to fit the workload at every moment by switching on or off its compute nodes, thus saving energy whenever these are idle. This technique has been applied to Load-Balancing clusters in [27–32] and in VMware vSphere [33] and Citrix XenServer [34] hypervisors. Recently, various software tools implementing this technique in HPC clusters have also been developed [35–37].

However, previous adaptive resource solutions for High Performance Computing (HPC) clusters have limited practical applications for two fundamental reasons. Firstly, as shown in [38,39], closed sets of expert-defined rules are not optimal to every scenario, leading to a lack of flexibility when it comes to complying with the preferences and tolerances of real-world cluster administrators in terms of impact in service quality and node reliability. Secondly, these solutions require its expert system to be tuned by hand, what is a complex task and is likely to conduce to incorrectly-configured systems that can cause substantial interferences with the cluster operation, such as node thrashing or reduction of its productivity, as demonstrated in [40].

Because of this, the tool EECluster is presented that overcomes these limitations. EECluster can improve the energy efficiency of HPC clusters by dynamically adapting their resources to the changing workloads. This is done using a Hybrid Genetic Fuzzy System as the decision-making mechanism and is tuned by means of multi-objective evolutionary algorithms in a machine learning approach to achieve good compliance with the administrator preferences.

The remainder of the paper is as follows. Section 2 explains the concept of eco-efficiency and details the modelling assumptions for the carbon footprint of an HPC. Section 3 explains the architecture of the EECluster tool. Section 4 explains the decision making-mechanism. Section 5 explains the learning algorithm used. Section 6 shows multiple experimental results in both synthetic and actual scenarios. Section 7 concludes the paper and discusses the future work.

2. Eco-Efficiency

The concept of eco-efficiency brings together economic and environmental factors for a more efficient use of resources and lower emissions [41]. Eco-efficiency is represented by the quotient between the service value and its environmental influence. In the particular case of HPCs, the service value is related to the Quality of Service (QoS), and the environmental influence affects both energy consumption and greenhouse gas emissions.

As mentioned in the introduction, the dependence between the energy consumption and the Quality of Service has been studied, and different strategies were proposed to improve their balance [35–40]. In this work, these studies are updated by including other sources of carbon dioxide emissions that are originated in the life cycle of a compute node. These additional sources are of secondary importance but nonetheless represent a significant part of the emissions. According to [42], manufacturing a computer requires more than 1700 kWh of primary energy, and more than 380 kg of CO_2 are emitted in the process, accounting for a significant fraction of the greenhouse emissions during the whole life of the equipment. It must be noted that a standard factor of 220 kg CO_2/MWh was assumed for manufacturing-related consumptions, corresponding to an energy mix with a significant proportion of wind power. For operation-related consumptions, the 370 kg CO_2/MWh emission factor reported by the Ministry of Agriculture, Food and Environment from the Government of Spain ("Ministerio de Agricultura, Alimentación y Medio Ambiente") [43] was used. This factor must be altered accordingly for clusters operating under different energy mixes.

As a consequence of this, in this paper it is proposed that three different aspects are taken into account in the model of the emissions of an HPC:

1. Dependence between QoS and primary consumption of energy. The primary savings are about 370 g of CO_2 for each kWh of electrical energy that is saved during the operation of the HPC.
2. Dependence between the QoS and the lifespan of the equipment. According to our own experience, the average life of a compute node in an HPC cluster is between 4 and 5 years. The number of failures of a given node during its whole lifetime is typically two or three. A rough estimation of the average number of failures of a single node during a year is 0.5 failures/year (thus 0.5 failures/year * 5 years = 2.5 failures). Both the life extent and the quantity of failures depend on the number of power-on and power-off cycles. Heavily loaded nodes might suffer from 0.75 failures/year and a shorter lifespan of 3 years. Assuming that the most common failures are power supplies, motherboards, and disk drives, the typical cost of a reparation can be estimated in 5% of the acquisition cost, *i.e.*, about 20 kg of CO_2 are saved for each failure that is prevented. Each additional year of use of a compute node saves more than 80 kg of CO_2 (approx. 22% of the total manufacturing emissions if the life is between 4 and 5 years, as mentioned). This includes the primary energy used for manufacturing a new node and the recycling costs of the discarded equipment.
3. Dependence between the QoS and the lifespan of the support equipment. An additional 1% was added for each saved kWh (2.2 g CO_2) and 1 g CO_2 for each saved power cycle. In the first case, this models the improved failure rate of support equipment such as cooling fans and air conditioning. The second case models different failures in the datacenter that may be caused by current surges when a large number of compute nodes are powered on or off at the same time.

The emissions model described in this section will be applied in the experiments of Section 6 to estimate global energy savings and carbon footprint reductions as a result of adopting of the proposed system.

3. Architecture

Computing clusters are a type of computer system consisting of multiple computers interconnected that, together, work as a single computing resource [44]. High Performance Computing (HPC) clusters are a particular type of cluster whose main purpose is to address complex and computationally-demanding problems, such as new material, semiconductors, or drugs design, cardiovascular engineering, new combustion systems, cancer detection and therapies, CO_2 sequestration, *etc.* [45].

HPC clusters typically combine a master node and several compute nodes. The master node is the only one accessible by the users and is tasked with the cluster management using various software components, including the Resource Management System (RMS) and monitoring tools (such as Ganglia, Nagios, Zabbix), among others. The RMS is a software layer which abstracts users from the cluster underlying hardware by providing a mechanism where they can submit resource requests to run any supplied software program (hereafter denoted as *jobs*). It is worth noting that cluster resources are represented logically by a number of slots which, depending on the RMS configuration, can depict form a single CPU core to a whole compute node. The RMS working cycle consists of (1) gathering job submissions in an internal queue; (2) running a job scheduling algorithm to find the best possible matching between the resources available in the compute nodes and the slots requested by each job and (3) assigning slots and dispatching the job to the compute nodes (see Figure 1). Data and results are passed between the master and the compute nodes through a shared network storage space by means of a network file system or a storage area network.

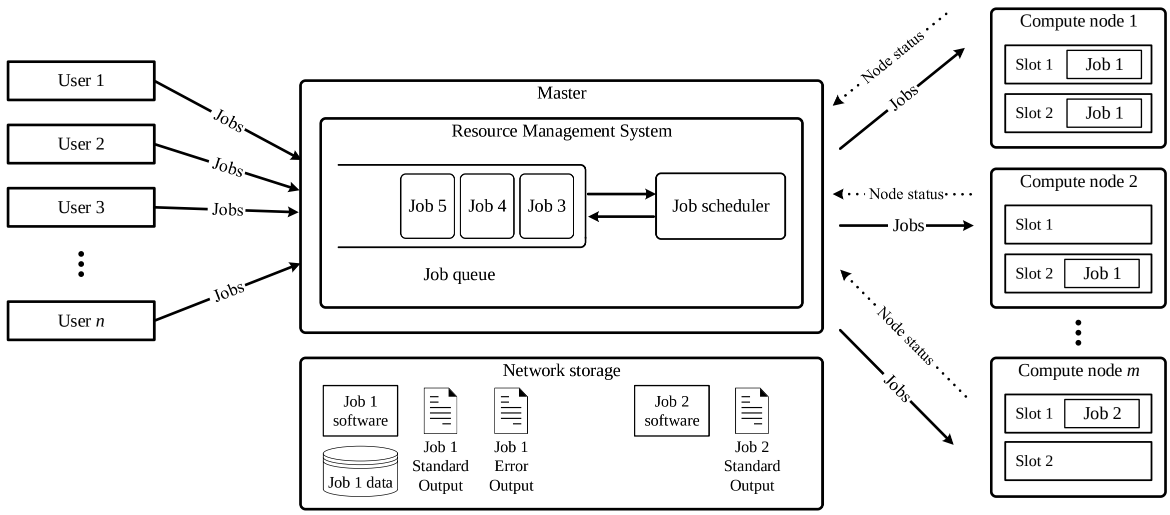


Figure 1. Resource Management System (RMS) components.

The EECluster tool is a solution which can reduce the carbon footprint of ordinary HPC clusters running open-source RMS, such as OGE (Oracle Grid Engine, Open Grid Engine/SGE (Sun Grid Engine, Son of Grid Engine) and PBS (Portable Batch System)/TORQUE (Terascale Open-source Resource and QUEue Manager) by implementing an intelligent mechanism to adapt the cluster resources to the current workload, saving energy when the cluster is underused. Specifically, the prototype of EECluster features only two out-of-the-box connectors for OGE/SGE and PBS/TORQUE, as these are two of the most used RMS worldwide in HPC infrastructures. As mentioned in reference [46], the OGE/SGE family (including its multiple branches of products and projects, such as Sun Grid Engine, Oracle Grid Engine, Open Grid Engine, and Son of Grid Engine) is a suitable choice for small and medium sites because it is easy to deploy and operate, what has leaded to a very substantial expansion over the last decade in HPC centres. On the other hand, TORQUE (Terascale Open-source Resource and QUEue Manager) is the open-source RMS based on the original PBS project (Portable Batch System), and it is arguably the most widely-used batch system nowadays in HPC grid infrastructures and also in small and medium site [46]. It is noteworthy that EECluster can be potentially integrated with any RMS as long as it provides a suitable interface in the form of either a series of command-line utilities or an API (Application Programming Interface) that allows EECluster to obtain the required information for its operation (detailed below).

EECluster is composed of a service (EEClusterd) and a learning algorithm, coupled with a Database Management System (DBMS) as the persistence system, and a web-based administration dashboard. The EEClusterd service periodically updates an internal set of cluster status records by retrieving information from multiple command-line applications. This information, which is stored in a Database Management System, is used by the EECluster decision-making mechanism to dynamically reconfigure the cluster resources by issuing power-on or shutdown commands to the compute nodes using the Power Management module. The learning algorithm mission is to find a set of optimal configurations for the decision-making mechanism from which the administrator can choose one according to its preferences in terms of impact in the service quality, energy savings, and node reconfigurations.

A functional prototype of EECluster can be downloaded via web [47,48], where can also be found a brief description of the software, quick start guides, contact address and acknowledgements.

Figure 2 provides a high-level overview of the system components. A detailed description of this architecture is out of the scope of this paper and can be found in reference [49].

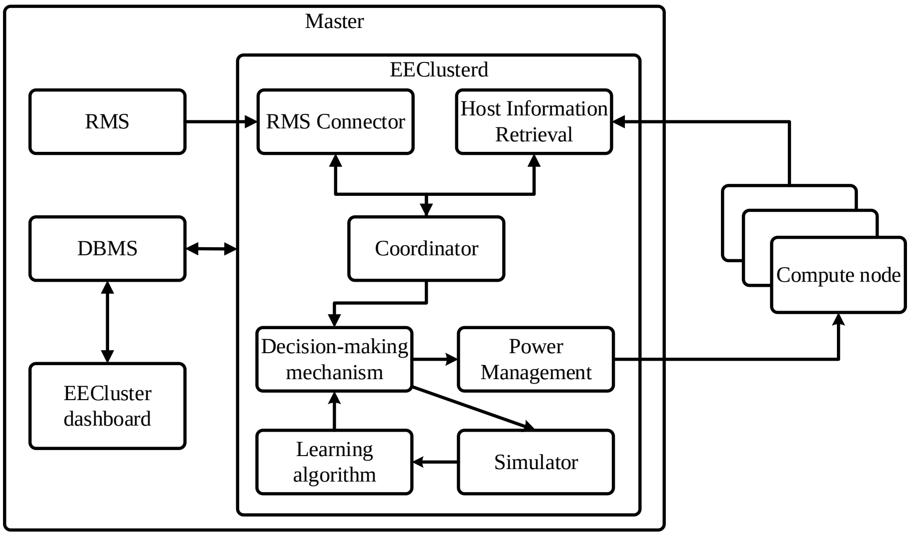


Figure 2. EECluster Tool: System components overview. DBMS: Database Management System.

4. Decision-Making Mechanism

The essential component in an adaptive resource cluster solution is the decision-making mechanism, for it is the one that determines the amount of nodes that will be available at every moment. As mentioned earlier, multiple approaches have been proposed previously based on sets of expert-defined rules, such as "if the node i has been idle for more than a t time threshold, it must be powered off". Closed expert systems like these have the advantage of coping better with unforeseen workload scenarios over systems learnt automatically in a machine learning approach. This is because machine-learnt systems are more likely to overtrain, especially in scenarios with great changes in the pattern of job arrivals. However, this advantage of simple expert systems comes at the price of low flexibility to adapt to both the sharp changes inherent to ordinary HPC clusters due to the large granularity of its workload (low number of concurrent jobs and multiple resources requested by each job), and the complex set of constraints implicit in the preferences of the cluster administrator. The first limitation leads to worse results than the ones obtainable with a pure machine learning approach. The second limitation leads to a potentially inadmissible solution for the cluster administrator if it does not comply with his or her tolerance of negative impacts on service quality and node reliability.

In order to avoid overtraining and achieve good generalization and flexibility capabilities, EECluster decision-making mechanism is a Hybrid Genetic Fuzzy System (HGFS) composed of both a set of crisp expert-defined rules and a set of fuzzy rules elicited automatically from previous workload records in a machine learning approach. The first set of human-generated expert rules was adapted from reference [36], and can be defined as follows:

- If the current number of resources are insufficient to run every queued job in a sequential manner, then keep powered on at least the highest number of slots requested by any queued job, as long as that amount does not exceed the total number of slots in the cluster.
- If the average waiting time for the queued jobs is higher than a given threshold t_{max} or if the number or queued jobs is higher than a given threshold n_{max}, then power on one slot.

- If the average waiting time for the queued jobs is lower than a given threshold t_{min} or if the number of queued jobs is lower than a given threshold n_{min}, then power off one slot.

The mission of this rule set is to assure that minimum working conditions for the cluster are met, avoiding undesired behaviours in unforeseen scenarios, which may lead to a dramatic impact in the service quality or to a node thrashing effect reducing node reliability and causing early damages in the hardware equipment.

The purpose of the second set of computer-generated rules is the progressive shutdown of idle nodes when the cluster load decreases. Each rule in this set defines the degree of truth of the assertion "the *i*-node must be powered off". This degree of truth is computed using a zero-order Tagaki–Sugeno–Kang fuzzy model [50,51] with N triangular fuzzy subsets on the domain of the nodes idle times and N weights between 0 and 1. For instance, if $N = 3$ then the computer-generated rules would consider three linguistic terms regarding the total amount of time that the *i*-node has been at idle state, with N_1 being "SHORT", N_2 being "MEDIUM" and N_3 being "LARGE". In this case, the degree of truth of the aforementioned assertion would be computed as:

$$\text{off}(\text{node } i) = \widetilde{\text{SHORT}}(\text{idle}_i) \cdot w_1 + \widetilde{\text{MEDIUM}}(\text{idle}_i) \cdot w_2 + \widetilde{\text{LARGE}}(\text{idle}_i) \cdot w_3 \tag{1}$$

idle_i is the amount of time that the *i*-node has been idle, $\widetilde{\text{SHORT}}$, $\widetilde{\text{MEDIUM}}$, and $\widetilde{\text{LARGE}}$ are fuzzy sets with triangular memberships [50], and w_1, w_2, w_3 are the N weighs taking values between 0 and 1.

Once each rule has been computed, results are combined so that the number of nodes to be powered off is the sum of the values off for each node. As can be seen, this second set of fuzzy rules does not require nodes to reach a certain crisp value before they can be selected for shutting down, but rather is applied to the cluster as a whole, as opposed to the rules proposed previously to power off idle nodes, such as in reference [36]. This approach allows the system to respond to smaller changes in the cluster load more frequently, thus progressively adapting its resources to better match workload valleys produced when jobs release their resources upon completion. Further information on the Hybrid Genetic Fuzzy System can be found in references [38,39].

Once a decision is made by the HGFS determining the number of slots that must be powered on/off, then this decision is translated to a set of physical compute nodes that will be chosen for reconfiguration. This is done considering the state of each node (only idle nodes can be shutdown and only powered-off nodes are candidates for power-on) plus two additional values: the node efficiency, measured as $\frac{\text{performance}}{\text{power consumption}}$, and the timestamp of its last failure. The latter is used to determine how likely it is for a given node to fail upon a reconfiguration request, in such a way that if a node was recently issued a power on/off command and it failed to comply with it, then the same problem is expected also to occur in the near future. However, if the node failed a long time ago it is more likely to have been repaired. The target nodes to be reconfigured are first split into two groups depending on whether they failed or not to comply with the last issued command. The first group, consisting of the nodes which worked correctly, are sorted according to their efficiency so that the least efficient ones are chosen to be powered off and, conversely, the most efficient ones are chosen to be powered on. If the nodes in the previous group are not enough to match the slots in the reconfiguration decision, the remaining nodes are chosen from the second group, which are sorted according to the timestamps of their failures, choosing first the nodes with the earliest values. That is, when a node is chosen from the second group it is because its last failure occurred before the failure of any other node. If this chosen node fails again, the timestamp of the last failure would be updated, thus the next time nodes are sorted, then it will be the last one. The idea behind this design is to prevent the system from selected systematically the same malfunctioning node if others are available for reconfiguration.

5. Learning Algorithm

The Hybrid GFS described is flexible enough to behave as desired in order to suit the cluster administrator preferences. However, this requires every HGFS parameter to be properly tuned, which is a complex task due to the presence of multiple conflicting objectives and to the huge amount of combinations that renders infeasible an extensive search. In particular, every instance or configuration for the previous HGFS is the combination of the following parameters:

$$(t_{min}, t_{max}, n_{min}, n_{max}, w_1, \ldots, w_N) \tag{2}$$

To address this problem, the EECluster learning algorithm uses multi-objective evolutionary algorithms (MOEAs) to find the parameters defining the HGFS by optimizing a fitness function consisting in three conflicting criteria: the quality of service (QoS), the energy saved, and the number of node reconfigurations. Specifically, EECluster uses the MOEA Framework [52] implementation of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [53] to obtain a Pareto Efficient Frontier from which the administrator can choose a suitable HGFS configuration. The Pareto Efficient Frontier is the set of configurations for the HGFS obtained in the experiment that are not worse than any other configuration in all components of the fitness function simultaneously. Every configuration in the Pareto Efficient Frontier is known to be "non-dominated". As can be seen, the result of the learning algorithm is not a single optimal solution (configuration for the HGFS) but rather a set of optimal configurations from which an expert human can pick the one that is best given his or her preferences. The reason for this is that there is no optimal solution because the three objectives involved are in conflict and attempts to apply any form of weighted sorting of the solutions obtained would lead to an inaccurate model of the preferences of the administrator.

For a given set of n jobs, where the j-th job ($j = 1 \ldots n$) is scheduled to start at time tsch_j, but effectively starts at time ton_j and stops at time toff_j, the quality of service in an HPC cluster reflects the amount of time that each job has to wait before it is assigned its requested resources. Once the job starts its execution, it will not be halted; thus, we focus only on its waiting time. Because jobs do not last the same amount of time, their waiting in the queue is better expressed as a ratio considering their execution time. It is noteworthy that the execution times of the job can differ greatly since they range from seconds to weeks or months. This can potentially lead to situations where very short jobs must wait over a hundred times their execution timespan, distorting the measurement of the quality of service and depicting inaccurately the cluster performance. Because of this, the 90 percentile is used instead of average:

$$\mathrm{QoS} = \min\left\{ p : ||\{ j \in 1 \ldots n : \frac{\mathrm{ton}_j - \mathrm{tsch}_j}{\mathrm{toff}_j - \mathrm{ton}_j} \leq p \}|| > 0.9\, n \right\} \tag{3}$$

where $||A||$ is the cardinality of the set A.

The energy saved is measured as the amount of watts-hour that were prevented from being wasted by shutting down idle nodes. Let c be the number of nodes, let $\mathrm{state}(i, t)$ be 1 if the i-th node ($i = 1 \ldots c$) is powered at time t, and 0 otherwise, let the time scale be the lapse between $\mathrm{tini}{=}\min_j\{\mathrm{tsch}_j\}$ and $\mathrm{tend}{=}\ \max_j\{\mathrm{toff}_j\}$. Lastly, let $\mathrm{power}_{idle}(i)$ be the power consumption measured in watts of the i-th node when it is at idle state. Then,

$$\text{Energy saved} = \sum_{i=1}^{c} \mathrm{power}_{idle}(i) \cdot (\mathrm{tend} - \mathrm{tini}) - \sum_{i=1}^{c} \mathrm{power}_{idle}(i) \int_{\mathrm{tini}}^{\mathrm{tend}} \mathrm{state}(i, t) \mathrm{d}t. \tag{4}$$

The node reconfigurations is the number of times that a node has been powered on or off. Let $\mathrm{nd}(i)$ be the number of discontinuities of the function $\mathrm{state}(i, t)$ in the time interval $t \in (\mathrm{tini}, \mathrm{tend})$:

$$\text{Reconfigured nodes} = \sum_{i=1}^{c} \mathrm{nd}(i) \tag{5}$$

The mission of the NSGA-II algorithm is to obtain a set of non-dominated configurations for the HGFS, guided by the previous fitness function, whose values are calculated by running a cluster simulation with a given number of nodes, slots, and job records, as seen in Figure 3.

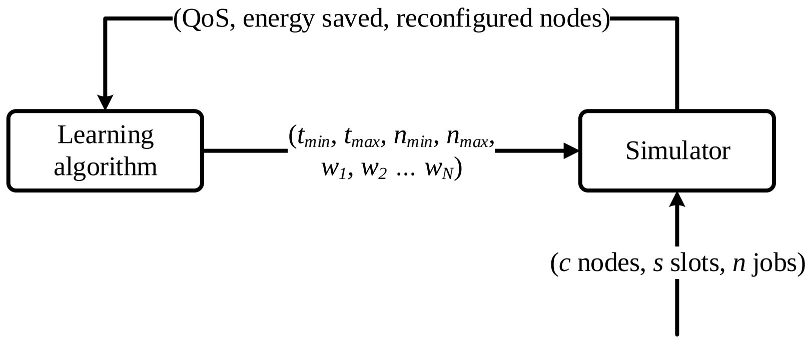


Figure 3. EECluster learning process. QoS: Quality of Service.

6. Experimental Results

In order to provide a sound answer on whether the decision-making mechanism has the required flexibility to perform correctly and suit any desired working mode, it must be tested in a range of cluster scenarios which together can build a significant representation of real-world clusters. To do so, a combination of synthetically-generated and actual cluster workloads from the Scientific Modelling Cluster (CMS) of the University of Oviedo [54] were used. Synthetic workloads represent four different scenarios with an increasing degree of fluctuation in terms of job arrival rates, each one spanning 24 months. Job arrivals in each scenario follow a Poisson process with the λ values shown in Table 1, and job run times are distributed exponentially with rate $\lambda = 10^{-5}$ s in all scenarios. As can be seen in the table, scenario 1 exhibits a cluster with a stable and sustained workload where all hours of the year have the same job arrival pattern. Scenario 2 adds a distinction between working, non-working, and weekend hours. Scenario 3 adds a substantial variation in the arrival rates depending on the week of the month, and scenario 4 increases this variation even more. On the other hand, the workloads from the CMS cluster consist of 2907 jobs spanned over 22 months. This real-world cluster, built from three independent computing clusters and five transversal queues using PBS as Resource Management System (RMS), can accurately show a very common activity pattern in most HPC clusters.

Table 1. Poisson process of job arrivals in each scenario.

Scenario	Day of Week	Hour Range	Week of Year	λ Value
1	All	All	All	2×10^{-4} s
2	Monday–Friday	8:00–20:00	All	2×10^{-4} s
	Saturday–Sunday	8:00–20:00	All	2×10^{-5} s
	Monday–Sunday	20:00–8:00	All	10^{-5} s
3	Monday–Friday	8:00–20:00	$w \% 5 = 0$	10^{-4} s
			$w \% 5 = 1$	2×10^{-4} s
			$w \% 5 = 2$	5×10^{-4} s
			$w \% 5 = 3$	5×10^{-4} s
			$w \% 5 = 4$	2×10^{-4} s
	Monday–Sunday	20:00–8:00	All	2×10^{-5} s
	Monday–Friday	8:00–20:00	All	10^{-5} s
4	Monday–Friday	8:00–20:00	$w \% 5 = 0$	10^{-4} s
			$w \% 5 = 1$	10^{-4} s
			$w \% 5 = 2$	5×10^{-4} s
			$w \% 5 = 3$	5×10^{-4} s
			$w \% 5 = 4$	10^{-4} s
	Monday–Sunday	20:00–8:00	All	2×10^{-5} s
	Monday–Friday	8:00–20:00	All	10^{-5} s

To assess the flexibility of the Hybrid GFS to suit the desired behaviour of the cluster administrator under different scenarios, a wide range of preferences must be tested. Given that these preferences are inherently subjective, a set of five different synthetic preferences were used in the experiments allowing us to measure to what extent the HGFS complies with them in each scenario as well to compare results across different scenarios. These synthetic preferences, denoted as "Hybrid GFS (QoS_{max}, $reconfigurations_{max}$)", determine that the HGFS configuration chosen in each case is the one with the maximum energy savings that has a service quality below QoS_{max} and a number of node reconfigurations not higher than $reconfigurations_{max}$ over the timespan of the validation and test sets. It must be noted that values used for these parameters in the experiment differ slightly from one scenario to another in order to adapt to the different average loads and arrival rates in each scenario.

To measure the effect of applying each HGFS configuration, a cluster simulator has been developed for both training and testing, so that every model can be evaluated in the three criteria of the fitness function. The holdout method was used for validation, with a 50/25/25% split in training, validation, and test, respectively. Results obtained from the experiment using the test set of each workload are shown in the following tables. They are measured in terms of impact in the service quality and reduction in the cluster carbon footprint achieved according to the CO_2 emission factors for the current energy mix, and the number node reconfigurations. Observe that the "Energy saved" column depends on this energy mix that has been estimated according to the 370 kg CO_2/MWh emission factor reported by the Ministry of Agriculture, Food and Environment from the Government of Spain ("Ministerio de Agricultura, Alimentación y Medio Ambiente") [43], but some of the greenhouse gas is not emitted at Spain; the primary energy consumption is assumed to be served by Spanish sources but secondary sources are located at the country where the compute nodes are manufactured. In this respect, Megawatt-hours of saved energy are understood as the amount of energy that would be saved according to the Spanish mix of generation; the actual amount of globally saved energy is possibly higher. Also, charts have been generated to depict the evolution over time of the active cluster resources and the requested slots by the queued and running jobs. In particular, results for scenario 1 are displayed in Table 2 and in Figure 4, scenario 2 in Table 3 and in Figure 5, scenario 3 in Table 4 and in Figure 6, and scenario 4 in Table 5 and in Figure 7. Lastly, results obtained for the CMS cluster recorded workloads are displayed in Table 6 and in Figure 8.

Table 2. Experiment results for the test set of scenario 1.

	Scenario 1 Test Set				
	QoS	**Energy Saved (%)**	**Energy Saved (MWh)**	**Carbon Reduction ($MtCO_2$)**	**Reconfigurations**
Hybrid GFS (0.00, 1000)	0.00×10^{0}	28.62%	8.30	3.07	487
Hybrid GFS (0.005, 1250)	4.25×10^{-3}	33.19%	9.63	3.56	945
Hybrid GFS (0.01, 1500)	9.32×10^{-3}	35.63%	10.34	3.82	1412
Hybrid GFS (0.015, 2000)	1.35×10^{-2}	37.16%	10.78	3.99	1921
Hybrid GFS (0.02, 3000)	1.92×10^{-2}	38.50%	11.17	4.13	2835

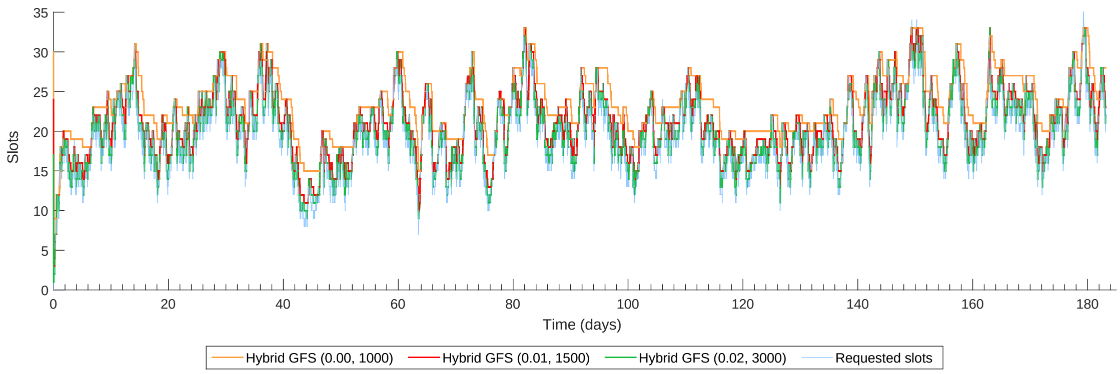


Figure 4. Cluster simulation trace for the test set of scenario 1. GFS: Genetic Fuzzy System.

Table 3. Experiment results for the test set of scenario 2.

	Scenario 2 Test Set				
	QoS	**Energy Saved (%)**	**Energy Saved (MWh)**	**Carbon Reduction ($MtCO_2$)**	**Reconfigurations**
Hybrid GFS (0.00, 500)	0.00×10^{0}	60.07%	17.43	6.45	218
Hybrid GFS (0.005, 500)	4.88×10^{-3}	65.46%	18.99	7.03	384
Hybrid GFS (0.01, 750)	9.45×10^{-3}	69.50%	20.17	7.46	635
Hybrid GFS (0.015, 1000)	1.43×10^{-2}	71.61%	20.78	7.69	838
Hybrid GFS (0.02, 1500)	1.84×10^{-2}	73.44%	21.31	7.88	1082

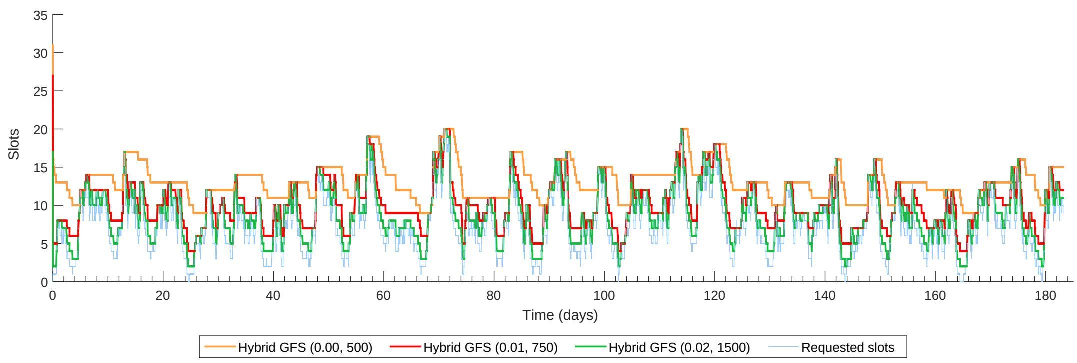


Figure 5. Cluster simulation trace for the test set of scenario 2.

Table 4. Experiment results for the test set of scenario 3.

	Scenario 3 Test Set				
	QoS	**Energy Saved (%)**	**Energy Saved (MWh)**	**Carbon Reduction ($MtCO_2$)**	**Reconfigurations**
Hybrid GFS (0.00, 500)	0.00×10^0	27.99%	8.12	3.00	194
Hybrid GFS (0.005, 500)	2.19×10^{-3}	41.53%	12.05	4.46	342
Hybrid GFS (0.01, 750)	8.12×10^{-3}	50.73%	14.72	5.45	671
Hybrid GFS (0.015, 1000)	1.17×10^{-2}	56.02%	16.25	6.01	1006
Hybrid GFS (0.02, 1500)	1.69×10^{-2}	59.44%	17.25	6.38	1334

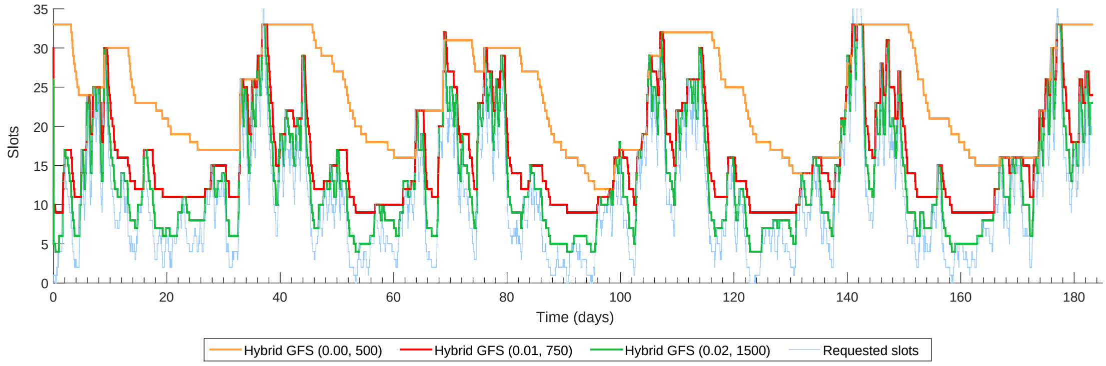


Figure 6. Cluster simulation trace for the test set of scenario 3.

Table 5. Experiment results for the test set of scenario 4.

	Scenario 4 Test Set				
	QoS	**Energy Saved (%)**	**Energy Saved (MWh)**	**Carbon Reduction ($MtCO_2$)**	**Reconfigurations**
Hybrid GFS (0.00, 500)	0.00×10^0	25.00%	7.25	2.68	173
Hybrid GFS (0.005, 500)	2.48×10^{-3}	46.09%	13.37	4.95	367
Hybrid GFS (0.01, 750)	4.23×10^{-3}	51.50%	14.94	5.53	473
Hybrid GFS (0.015, 1000)	1.43×10^{-2}	60.06%	17.42	6.45	1010
Hybrid GFS (0.02, 1500)	1.92×10^{-2}	63.51%	18.43	6.82	1253

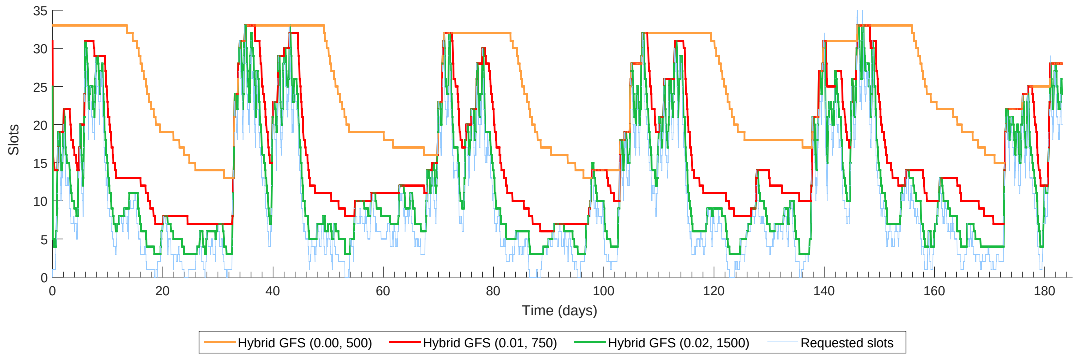


Figure 7. Cluster simulation trace for the test set of scenario 4.

Table 6. Experiment results for the test set of the Scientific Modelling Cluster (CMS) workload records.

	CMS Cluster Test Set				
	QoS	**Energy Saved (%)**	**Energy Saved (MWh)**	**Carbon Reduction ($MtCO_2$)**	**Reconfigurations**
Hybrid GFS (0.00, 100)	0.00×10^0	46.67%	13.38	4.95	42
Hybrid GFS (0.05, 500)	4.46×10^{-2}	64.26%	18.42	6.82	293
Hybrid GFS (0.10, 500)	6.82×10^{-2}	68.60%	19.67	7.28	361
Hybrid GFS (0.25, 750)	1.49×10^{-1}	70.34%	20.17	7.46	363
Hybrid GFS (0.50, 750)	1.95×10^{-1}	75.54%	21.66	8.01	590

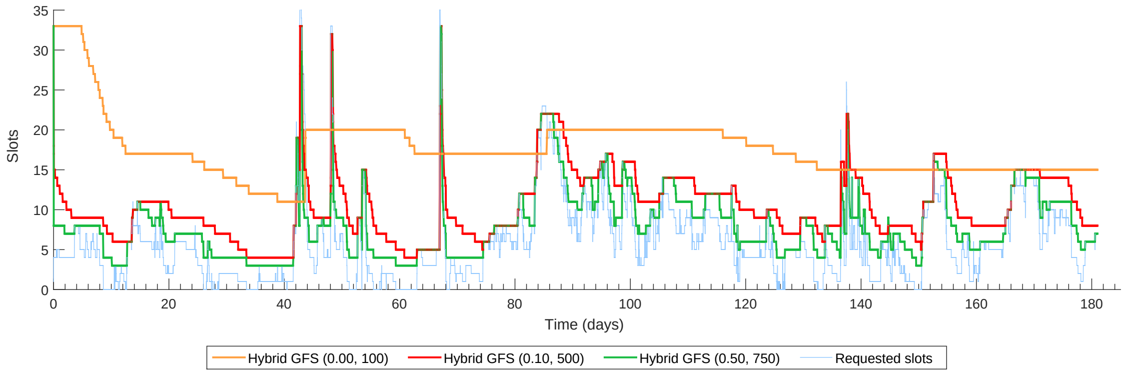


Figure 8. Cluster simulation trace for the test set of the CMS cluster workload records.

As can be seen from these results, in every cluster scenario used in the experiments, the learning algorithm found a configuration for the HGFS that achieves significant energy savings without any noticeable impact in service quality. Also, additional configurations were found that comply with the synthetic administrator preferences defined, increasing energy savings while strictly complaining with the constraints set in the aforementioned preferences in terms of QoS and node reconfigurations. It should be noted that the five configurations displayed in the previous tables are only a small selection of the vast set obtained in the Pareto Efficient Frontier, and many other solutions are available that can save even more energy at the cost of a higher penalty in service quality.

The experiments also show that the results obtained differ significantly depending on the characteristics of the workload. In scenario 1, the regular job arrival rate depicts a workload where the distances between peaks are very short and valleys tend to be shallow. This leads to HGFS configurations with a higher average number of reconfigurations and a relatively low amount of saved energy. Also, as can be seen in Table 2, rising the degree of tolerance for impact in QoS from 0.0 to 0.02 and node reconfigurations up to 3000 only allows an increase in the overall energy savings of 9.88%. Results improve progressively as job arrival patterns vary over time and the workload becomes more irregular with deeper valleys and longer distances between peaks. For instance, scenario 2 allows an increase in energy savings of 13.37% between the HGFS configurations obtained for a QoS of 0.0 and a QoS of 0.02. In scenario 3, the difference grows up to 31.45%, and in scenario 4 reaches 38.51%.

These two last scenarios show important results since they represent the workload patterns most likely to occur in real-world HPC clusters. This can be verified by the results obtained using actual records from the CMS cluster of the University of Oviedo, where the workload is even sharper than in scenario 4, with energy savings between 46.67% and 75.54%, depending on the administrator preferences. These values can be translated to actual power savings ranging from 13.38 MWh to 21.66 MWh over the course of the test set, and figures of carbon reduction between 4.95 and 8.01 tonnes of CO_2.

7. Concluding Remarks

The EECluster tool has been designed to reduce the carbon footprint of HPC clusters by improving their energy efficiency. This software package implements the adaptive resource cluster technique in clusters running OGE/SGE or PBS/TORQUE as RMS, allowing for practical application in real-world scenarios, owing to the flexibility of its sophisticated machine-learnt decision-making mechanism to comply with cluster administrator preferences. This mechanism, based on Computational Intelligence techniques, is learnt by means of multi-objective evolutionary algorithms to assure finding a suitable configuration that maximises energy savings within the tolerance region of the administrator in terms of service quality and node reliability.

Thorough experimental studies based on both synthetic and actual workloads from the Scientific Modelling Cluster of Oviedo University [54] provide empirical evidence of the ability of EECluster to deliver good results in multiple scenarios, supporting the adoption of EECluster to reduce the environmental impact of real world clusters.

Acknowledgments: This work has been partially supported by the Ministry of Economy and Competitiveness ("Ministerio de Economía y Competitividad") from Spain/FEDER under grants TEC2012-38142-C04-04, TEC2015-67387-C4-3-R and TIN2014-56967-R and by the Regional Ministry of the Principality of Asturias under grant FC-15-GRUPIN14-073.

Author Contributions: All authors participated equally to this research work from its inception, with L.S. providing the design of the decision-making mechanism and learning algorithm, J.R. providing the architectural design of the solution for OGE/SGE and PBS/TORQUE clusters, and A.C.-F. carrying out software implementation and experimentation. All authors also prepared, reviewed and agreed the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Acronyms

HPC	High Performance Computing
RMS	Resource Management System
OGE	Oracle Grid Engine/Open Grid Engine
SGE	Sun Grid Engine/Son of Grid Engine
PBS	Portable Batch System
TORQUE	Terascale Open-source Resource and QUEue Manager
DBMS	Database Management System
IPMI	Intelligent Platform Management Interface
WOL	Wake On Lan
SSH	Secure SHell
HGFS	Hybrid Genetic Fuzzy System
TSK	Tagaki-Sugeno-Kang
QoS	Quality of Service
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
MOEAs	MultiObjective Evolutionary Algorithms

References

1. Delforge, P.; Whitney, J. *Issue Paper: Data Center Efficiency Assessment Scaling up Energy Efficiency across the Data Center Industry: Evaluating Key Drivers and Barriers*; Technical Report; Natural Resources Defense Council (NRDC): New York, NY, USA, 2014.
2. U.S. Environmental Protection Agency. *Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431*; Technical Report; ENERGY STAR Program: Washington, DC, USA, 2007.
3. Ebbers, M.; Archibald, M.; Fonseca, C.F.F.D.; Griffel, M.; Para, V.; Searcy, M. *Smarter Data Centers: Achieving Greater Efficiency*; Technical Report; IBM Redpaper: Research Triangle Park, NC, USA, 2011.
4. The Economist Intelligence Unit. *IT and the Environment: A New Item on the CIO's Agenda?* Technical Report; The Economist: London, UK, 2007.
5. Gartner. *Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO_2 Emissions*; Gartner: Stamford, CT, USA, 2007.
6. Forrest, W.; Kaplan, J.M.; Kindler, N. *Data Centers: How to Cut Carbon Emissions and Costs*; Technical Report; McKinsey & Company: New York, NY, USA, 2008.
7. Valentini, G.L.; Lassonde, W.; Khan, S.U.; Min-Allah, N.; Madani, S.A.; Li, J.; Zhang, L.; Wang, L.; Ghani, N.; Kolodziej, J.; *et al*. An overview of energy efficiency techniques in cluster computing systems. *Clust. Comput.* **2013**, *16*, 3–15.
8. Haring, R.; Ohmacht, M.; Fox, T.; Gschwind, M.; Satterfield, D.; Sugavanam, K.; Coteus, P.; Heidelberger, P.; Blumrich, M.; Wisniewski, R.; *et al*. The IBM blue gene/Q compute chip. *IEEE Micro* **2012**, *32*, 48–60.

9. IBM Systems and Technology Group. *IBM System Blue Gene/Q*; Technical Report; IBM: Somers, NY, USA, 2011.
10. Hsu, C.H.; Kremer, U. The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction. *ACM SIGPLAN Not.* **2003**, *38*, 38–48.
11. Hsu, C.H.; Feng, W.C. A Power-Aware Run-Time System for High-Performance Computing. In Proceedings of the ACM/IEEE SC 2005 Conference (SC '05), Seattle, WA, USA, 12–18 November 2005; IEEE: Washington, DC, USA, 2005; p. 1.
12. Freeh, V.W.; Lowenthal, D.K.; Pan, F.; Kappiah, N.; Springer, R.; Rountree, B.L.; Femal, M.E. Analyzing the Energy-Time Trade-off in High-Performance Computing Applications. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 835–848.
13. Lim, M.; Freeh, V.; Lowenthal, D. Adaptive, Transparent Frequency and Voltage Scaling of Communication Phases in MPI Programs. In Proceedings of the ACM/IEEE SC 2006 Conference (SC '06), Tampa, FL, USA, 11–17 November 2006; IEEE: Tampa, FL, USA, 2006; p. 14.
14. Chen, Y.; Zeng, Y. Automatic energy status controlling with dynamic voltage scaling in poweraware high performance computing cluster. In Proceedings of the Parallel and Distributed Computing, Applications and Technologies (PDCAT), Gwangju, Korea, 20–22 October 2011; IEEE: Gwangju, Korea, 2011; pp. 412–416.
15. Ge, R.; Feng, X.; Feng, W.C.; Cameron, K.W. CPU MISER: A Performance-Directed, Run-Time System for Power-Aware Clusters. In Proceedings of the 2007 International Conference on Parallel Processing (ICPP 2007), Xi'an, China, 10–14 September 2007; IEEE: Xi'an, China, 2007; pp. 18–25.
16. Huang, S.; Feng, W. Energy-Efficient Cluster Computing via Accurate Workload Characterization. In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai, China; 18–21 May 2009; IEEE: Shanghai, China, 2009; pp. 68–75.
17. Chetsa, G.L.T.; Lefrvre, L.; Pierson, J.M.; Stolf, P.; Da Costa, G. A Runtime Framework for Energy Efficient HPC Systems without a Priori Knowledge of Applications. In Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems, Singapore, 17–19 December 2012; IEEE: Singapore, 2012; pp. 660–667.
18. Zong, Z.; Ruan, X.; Manzanares, A.; Bellam, K.; Qin, X. Improving Energy-Efficiency of Computational Grids via Scheduling. In *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing*; Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A., Eds.; IGI Global: Hershey, PA, USA, 2010; Chapter 22.
19. Zong, Z.; Nijim, M.; Manzanares, A.; Qin, X. Energy efficient scheduling for parallel applications on mobile clusters. *Clust. Comput.* **2007**, *11*, 91–113.
20. Bash, C.; Forman, G. Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In Proceedings of the 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference, Santa Clara, CA, USA, 17–22 June 2007; USENIX Association: Berkeley, CA, USA, 2007; pp. 29:1–29:6.
21. Tang, Q.; Gupta, S.K.S.; Varsamopoulos, G. Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 1458–1472.
22. Alonso, P.; Badia, R.M.; Labarta, J.; Barreda, M.; Dolz, M.F.; Mayo, R.; Quintana-Orti, E.S.; Reyes, R. Tools for Power-Energy Modelling and Analysis of Parallel Scientific Applications. In Proceedings of the 2012 41st International Conference on Parallel Processing; IEEE: Pittsburgh, PA, USA, 10–13 September 2012; pp. 420–429.
23. Schubert, S.; Kostic, D.; Zwaenepoel, W.; Shin, K.G. Profiling software for energy consumption. In Proceedings of the 2012 IEEE International Conference on Green Computing and Communications, GreenCom 2012, Besancon, France, 20–23 November 2012; IEEE: Besancon, France; pp. 515–522.
24. Freeh, V.W.; Lowenthal, D.K. Using multiple energy gears in MPI programs on a power-scalable cluster. In Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming—PPoPP '05, Chicago, IL, USA, 15–17 June 2005; ACM Press: New York, NY, USA, 2005; pp. 164–173.
25. Li, D.; Nikolopoulos, D.S.; Cameron, K.; de Supinski, B.R.; Schulz, M. Power-aware MPI task aggregation prediction for high-end computing systems. In Proceedings of the 2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS), Atlanta, GA, USA, 19–23 April 2010; IEEE: Atlanta, GA, USA, 2010; pp. 1–12.

26. Xian, C.; Lu, Y.H.; Li, Z. A programming environment with runtime energy characterization for energy-aware applications. In Proceedings of the 2007 International Symposium on Low Power Electronics and Design—ISLPED '07, Portland, OR, USA, 27–29 August 2007; ACM Press: New York, NY, USA, 2007; pp. 141–146.
27. Pinheiro, E.; Bianchini, R.; Carrera, E.V.; Heath, T. Load balancing and unbalancing for power and performance in cluster-based systems. In Proceedings of the Workshop on Compilers and Operating Systems for Low Power, Barcelona, Spain, 9 September 2001; Volume 180, pp. 182–195.
28. Das, R.; Kephart, J.O.; Lefurgy, C.; Tesauro, G.; Levine, D.W.; Chan, H. Autonomic multi-agent management of power and performance in data centers. In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track—AAMAS '08, Estoril, Portugal, 2008; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2008; pp. 107–114.
29. Elnozahy, E.; Kistler, M.; Rajamony, R. Energy-efficient server clusters. In *Power-Aware Computer Systems*; Springer-Verlag: Berlin/Heidelberg, Germany, 2003; pp. 179–197.
30. Berral, J.L.; Goiri, Í.; Nou, R.; Julià, F.; Guitart, J.; Gavaldà, R.; Torres, J. Towards energy-aware scheduling in data centers using machine learning. In Proceedings of the 1st International Conference on EnergyEfficient Computing and Networking—e-Energy '10, Passau, Germany, 13–15 April 2010; ACM Press: New York, NY, USA, 2010; Volume 2, p. 215.
31. Lang, W.; Patel, J.M.; Naughton, J.F. On energy management, load balancing and replication. *ACM SIGMOD Rec.* **2010**, *38*, 35–42.
32. Entrialgo, J.; Medrano, R.; García, D.F.; García, J. Autonomic power management with self-healing in server clusters under QoS constraints. In *Computing*; Springer: Vienna, Austria, 2015; pp. 1–24.
33. VMware. VMware Distributed Power Management Concepts and Use. Available online: http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf (accessed on 1 March 2016).
34. Citrix Systems. XenServer - Server Virtualization and Hypervisor Management. Available online: http://www.citrix.com/products/xenserver/overview.html (accessed on 1 March 2016).
35. Alvarruiz, F.; de Alfonso, C.; Caballer, M.; Hernández, V. An Energy Manager for High Performance Computer Clusters. In Proceedings of the 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications, Leganes, Spain, 10–13 July 2012; IEEE: Leganés, Spain, 2012; pp. 231–238.
36. Dolz, M.F.; Fernández, J.C.; Iserte, S.; Mayo, R.; Quintana-Ortí, E.S.; Cotallo, M.E.; Díaz, G. EnergySaving Cluster experience in CETA-CIEMAT. In Proceedings of the 5th Iberian GRID Infrastructure Conference, Santander, Spain, 8–10 June 2011.
37. Kiertscher, S.; Zinke, J.; Gasterstädt, S.; Schnor, B. Cherub: Power Consumption Aware Cluster Resource Management. In Proceedings of the 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom) Green Computing and Communications (GreenCom), Hangzhou, China, 18–20 December 2010; pp. 325–331.
38. Cocaña-Fernández, A.; Ranilla, J.; Sánchez, L. Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling. *J. Supercomput.* **2014**, *71*, 1163–1174.
39. Cocaña-Fernández, A.; Sánchez, L.; Ranilla, J. Leveraging a predictive model of the workload for intelligent slot allocation schemes in energy-efficient HPC clusters. *Eng. Appl. Artif. Intell.* **2016**, *48*, 95–105.
40. Cocaña-Fernández, A.; Ranilla, J.; Sánchez, L. Energy-Efficient Allocation of Computing Node Slots in HPC Clusters through Evolutionary Multi-Criteria Decision Making. In Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2014, Cádiz, Spain, 3–7 July 2014; pp. 318–330.
41. Hendrik, A.; Bidwell, V.R. *Measuring Eco-Efficiency: A Guide to Reporting Company Performance*; World Business Council for Sustainable Development: Geneva, Switzerland, 2000.
42. Deng, L.; Babbitt, C.W.; Williams, E.D. Economic-balance hybrid LCA extended with uncertainty analysis: Case study of a laptop computer. *J. Clean. Prod.* **2011**, *19*, 1198–1206.
43. Ministerio de Agricultura, Alimentación y Medio Ambiente. *Factores de Emisión: Registro de Huella de Carbono, Compensación y Proyectos de Absorción de Dióxido de Carbono*; Technical Report; Ministerio de Agricultura, Alimentación y Medio Ambiente: Madrid, Spain, 2015.

44. Yeo, C.S.; Buyya, R.; Hossein. P.; Rasit, E.; Graham, P.; Sommers, F. Cluster Computing: High-Performance, High-Availability, and High-Throughput Processing on a Network of Computers. In *Handbook of Nature-Inspired and Innovative Computing*; Zomaya, A., Ed.; Springer US: New York, NY, USA, 2006; pp. 521–551.
45. National Science Foundation. *Advisory Committee for Cyberinfrastructure Task Force on Grand Challenges*; Technical Report; National Science Foundation: Arlington, VA, USA, 2011.
46. Cacheiro, J. *Analysis of Batch Systems*; Technical Report; CESGA: Santiago de Compostela, Galicia, Spain, 2014.
47. IRPC Group. EECluster: A Software Tool to Efficiently Manage the Energy Consumption of HPC Clusters. Available online: http://pirweb.edv.uniovi.es/eecluster (accessed on 1 March 2016).
48. SourceForge. EECluster download | SourceForge.net. Available online: http://sourceforge.net/projects/eecluster/ (accessed on 1 March 2016).
49. Cocaña-Fernández, A.; Sánchez, L.; Ranilla, J. A software tool to efficiently manage the energy consumption of HPC clusters. In Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Istanbul, Turkey, 2–5 August 2015; pp. 1–8.
50. Ishibuchi, H.; Nakashima, T.; Nii, M. *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing)*; Springer-Verlag New York, Inc.: Secaucus, NJ, USA, 2004.
51. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132.
52. MOEA Framework, a Java library for multiobjective evolutionary algorithms. Available online: http://moeaframework.org/ (accessed on 1 March 2016).
53. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197.
54. University of Oviedo. Scientific Modelling Cluster. Available online: http://cms.uniovi.es (accessed on 1 March 2016).

© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).

Título

Multicriteria design of cost-conscious fuzzy rule-based classifiers

Autores

Alberto Cocaña-Fernández, José Ranilla, Roberto Gil-Pita and Luciano Sánchez

Journal

***International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems**, Aceptado, 2017*

Ranking

Factor de impacto (JCR 2015): 1,000

Categorías:

Computer Science, Artificial Intelligence, 87/130 (Cuartil Q3)

Carta de aceptación

Ref.: Ms. No. IJUFKS-D-16-00166R2
Multicriteria design of cost-conscious fuzzy rule-based classifiers
International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

Dear Mr. Alberto Cocaña,

I am pleased to inform you that your work has now been accepted for publication in International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.

It was accepted on Apr 20, 2017.

If you wish to have your article published as Open Access, please note the Article-Processing Charge (APC) is USD1500.
You may contact us or visit http://www.worldscientific.com/page/open for more details.

Thank you for submitting your work to this journal.
Yours sincerely,

Bernadette Bouchon-Meunier

Editor-in-Chief

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems
© World Scientific Publishing Company

Multicriteria design of cost-conscious fuzzy rule-based classifiers

Alberto Cocaña-Fernández
Departamento de Informática, Universidad de Oviedo
33204 Gijón, Asturias, Spain
cocanaalberto@gmail.com

José Ranilla
Departamento de Informática, Universidad de Oviedo
33204 Gijón, Asturias, Spain
ranilla@uniovi.es

Roberto Gil-Pita
Departmento de Teoría de la Señal y Comunicaciones, Universidad de Alcalá
28871 Alcalá de Henares, Madrid, Spain
roberto.gil@uah.es

Luciano Sánchez
Departamento de Informática, Universidad de Oviedo
33204 Gijón, Asturias, Spain
luciano@uniovi.es

Received (received date)
Revised (revised date)

Many real-world classification systems must comply with a series of inherent restrictions to the problem at hand such as response times, power consumptions or computational costs. This poses a fundamental limitation to traditional performance-driven classifiers and learning algorithms by restraining their applicability in cost-sensitive scenarios. Because of this, fuzzy systems are leveraged to learn cost-conscious multi-stage classifiers through multiobjective optimization to find a set of optimal tradeoffs between accuracy and any related cost. This approach allows to find a suitable balance between all objectives regardless of the scenario. Experimental evaluations were done for Sound Environment Classification in modern battery-powered hearing aids by jointly optimising classification accuracy and computational costs.

Keywords: Cost-conscious machine learning, Multi-stage classification, Fuzzy rule learning, Embedded devices, Hearing aids

1. Introduction

Intelligent machine learning systems have already been widely adopted in a number of data-intensive fields such as astronomy, biology, climatology, medicine, finance and economy. Nevertheless, most of these systems are performance-driven, focusing solely in classification and prediction accuracy.[1] This approach, though, poses a fundamental limitation for many real-world classification systems that must account explicitly for the total amount of time, memory, storage, power consumption and/or monetary costs. This is the case when these are inherent restrictions to the problem at hand, whether it is due to the limited computational resources or battery capacity of the underlying hardware (e.g. sensors in embedded devices), a fast response time is required for real-time operation (e.g. credit card transaction vetting or speech-to-speech translation), feature extraction is either obtrusive or involve significant acquisition costs (medical diagnosis), CPU time is expensive and must be budgeted for (e.g. search engines and product recommendations), etc.[17,24,35,45,60]

Let's take the example of machine learning in low-power embedded, mobile and/or wearable devices. There has been a growing interest in this research field given that the reduced cost, small size and general availability allow for the development of a wide range of biomedical, communication, location-based, business and surveillance applications.[8,32,40,63] An example of these applications is the implementation of Human Activity Recognition systems in smartphones and wearables for remote pacient monitoring, allowing a continuous health and well-being supervision of disabled and elderly people,[6] monitoring the severity of symptoms and motor complications in patients with Parkinson disease,[42] real-time control of physiological signals through electrocardiography,[43,52] etc. However, the fundamental challenge for all these applications are the energy and computational constraints of the battery-powered underlying devices. As a result of this, there is crucial demand for energy-efficient classifiers capable of reducing the computational costs and finding a suitable trade-off between accuracy and power consumption, thus maximizing battery life to a feasible extent for a practical use.[5,6,8,32,40–42,52]

Reducing the computational complexity of classification algorithms has been thoroughly studied for many years and there is extensive literature regarding this subject. A well-known approach is feature selection, which consists of selecting a subset of the attributes describing each instance to focus the attention of the classification over the most relevant ones.[31] This technique is adopted in order to improve the classification performance of the classifiers, and also learn faster and more computationally-efficient predictors.[31,59] A vast number of methods proposed for feature selection can be found in the literature, but given the idea driving this paper, the solution presented in Reference 10 is particularly interesting for it describes both a wrapper and a ranker method that explicitly and directly minimise the total acquisition cost of the selected features.

Efficient implementations of commonly-used classifiers such as Support Vector Machines (SVM) or Artificial Neural Networks have also been proposed to reduce their computational complexity. See, for example, a reduced version of the SVM in Reference 33, an approximation of SVM solutions in Reference 30, or an approximation of the Gaussian RBF kernel is proposed to speed up SVM evaluations in Reference 46.

The adoption of multi-stage architectures where a series of individual classifiers with increasing computational costs are combined in a hierarchical structure, such as a cascade or a tree can also improve the efficiency of traditional classifiers. The idea is to progressively improve accuracy with every new stage, at the expense of more data and/or computational costs.[47] This technique allows the classification process to stop whenever has reached a desired certainty, reducing the classification and feature-related costs for simple instances, and only requiring maximum costs for the complex ones, thus improving overall efficiency. Examples of this technique can be found in References 45, 47, 54, 57, 61. Evolving FRB classifiers also consider computational cost as they are based on recursive calculations that allow concomitant training and inference at the expense of less accuracy in the first samples 15, 16, 29, 44.

Classifiers ensembles built using techniques such as bagging, boosting or stacking can also be trimmed to improve their efficiency. Unnecessarily large ensembles can be pruned to find efficient sub-ensembles that can perform as well as the original ensemble with a fraction of the memory and computational requirements.[39,64] See, for instance, References 64, 39, or 36.

So far, every method described attempts to improve the efficiency of classifiers by reducing their computational complexity with a negligible or minimal impact in their accuracy. Notwithstanding this, finding a suitable solution in every potential scenario requires the problem to be addressed not a single-objective optimization one, but rather as a multi-objective one where the result is a set of optimal tradeoffs between classification performance and every related cost that must be accounted for. This idea is key in order to find the most suitable solution to every problem by complying with both its inherent constraints (maximum response times, memory requirements, floating-point operations per second, etc.) and the subjective preferences of a human expert (such as the optimal balance between accuracy and battery life, power consumptions, monetary costs, etc.). Examples regarding this multiobjective approach can be found in the literature regarding feature selection,[62] ensemble learning,[55] and also classifier design.[2,3,18,21,25–27]

Despite this, and according to the best of our knowledge, it has not been yet presented a holistic and flexible-enough approach to learn cost-aware classifiers suitable for all scenarios. The reason for this is that no previously proposed method is capable of finding the optimal and desired balance between accuracy and every explicit cost relevant to the problem at hand (both regarding the class-inferencing process and the feature extraction), whether these are computational costs, time, power consumption, monetary units, degree of obtrusiveness, etc. Because of this we propose a cost-conscious multi-stage Fuzzy Rule-Based Classifier (FRBC) learned by means of multi-objective evolutionary algorithms (MOEAs) jointly maximising classification performance and minimising every cost that must be accounted for. The main characteristic of this FRBC is the flexibly to achieve a great coverage and diversity of the objective space in the Pareto Efficient Frontier, thus maximising the likelihood that a human expert would find a suitable solution that befits its subjective set of preferences given in any scenario.

The remainder of the paper is as follows. Section 2 explains the multi-stage Fuzzy Rule-Based Classifier proposed. Section 3 explains the learning algorithm used. Section 4 shows the experimental results. Section 5 concludes the paper.

2. Fuzzy Rule-Based Classifiers

The proposed system is a multi-stage Fuzzy Rule-Based Classifier (FRBC) with a tree-like rule base elicited from data through Multiobjective Evolutionary Programming, as explained in the next section. First we introduce the rule base of a single-stage FRBC and then its generalization to n stages.

2.1. *Single-stage FRBC*

Let's state that the classification problem consists of assigning a class $\hat{c}$ from the finite set $C = \{c_1 \cdots c_K\}$ blue to an F-dimensional instance $\mathrm{x} = \{x_1 \cdots x_F\}$ denoting the set of features characterising each example. Given so, the rule base of the proposed FRBC can be simplified as K *if-then* rules such:

$$\text{Rule } Rj: \quad \textbf{if } x_1 \text{ is } A_{j1} \textbf{ and } \cdots \textbf{ and } x_F \text{ is } A_{jF} \textbf{ then } \hat{c} \text{ is } c_j \tag{1}$$

where A_{jn} is a linguistic term given for the the n-th attribute.

Note that the classification capabilities of the FRBC lie in the complexity of the antecedent of each rule and not in the amount of rules. Each rule antecedent is built from multiple terms combined through a series of connectives, including the logical conjunction (intersection $a \wedge b$), the logical disjunction (union $a \vee b$), the logical negation (complement $!\,a$) and the subtraction $(a - b)$. The leaf terms in each antecedent are assertions such as "x is A", where x is an input attribute and A is a fuzzy set representing a linguistic value like SMALL, MEDIUM, LARGE, etc. These fuzzy sets are defined by either triangular or trapezoidal membership functions μ mapping the normalized input attributes x in the domain $[0,1]$ to a membership grade between 0 and 1 in respect to the corresponding fuzzy set. Also, in order to maximize the discrimination power, the fuzzy partitions for each input attribute are non-uniform as they are composed of a variable number of diversely shaped fuzzy sets.

According to this, the FRBC rule bases are valid chains in the context-free grammar defined by the following production rules:

$$
\begin{array}{ll}
\text{RULE } R_j \rightarrow & \text{if CONDITION then class is } C_j \\
\text{CONDITION} \rightarrow & (\text{CONDITION} \wedge \text{CONDITION}) \\
 & \mid (\text{CONDITION} \vee \text{CONDITION}) \\
 & \mid (\text{CONDITION} - \text{CONDITION}) \\
 & \mid \,! \text{ CONDITION} \\
 & \mid \text{ASSERTION} \\
\text{ASSERTION} \rightarrow & x_i \textit{ is } \widetilde{Lt}(a,b) \mid x_i \textit{ is } \widetilde{Tr}(a,b,c) \mid x_i \textit{ is } \widetilde{Rt}(a,b)
\end{array}
$$

where $\widetilde{Lt}$, $\widetilde{Rt}$, $\widetilde{Tr}$ are, respectively, left trapezoidal, right trapezoidal and triangular

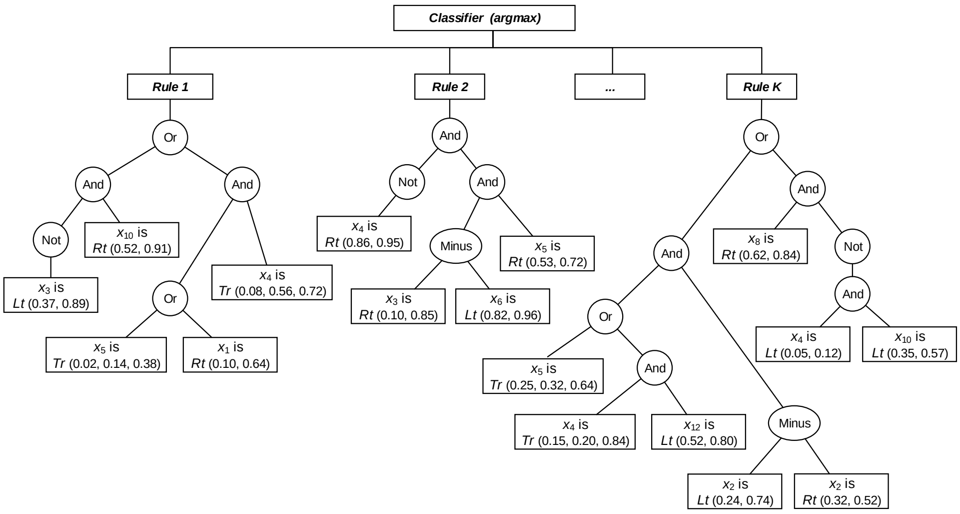


Fig. 1: Genotype of single-stage FRBC individual

fuzzy sets.[28] The genotype of an individual generated according to this grammar is showed in Figure 1 and some of its fuzzy partitions are depicted in Figure 2.

Fuzzy reasoning is done using the single winner-based method.[28] Essentially, the FRBC classifies instances according to the rule with the highest compatibility grade, so, given an input instance x, the output class $\hat{c}$ is determined by:

$$\hat{c} = \arg\max_{j} \mu_{R_j}(x) \tag{2}$$

where

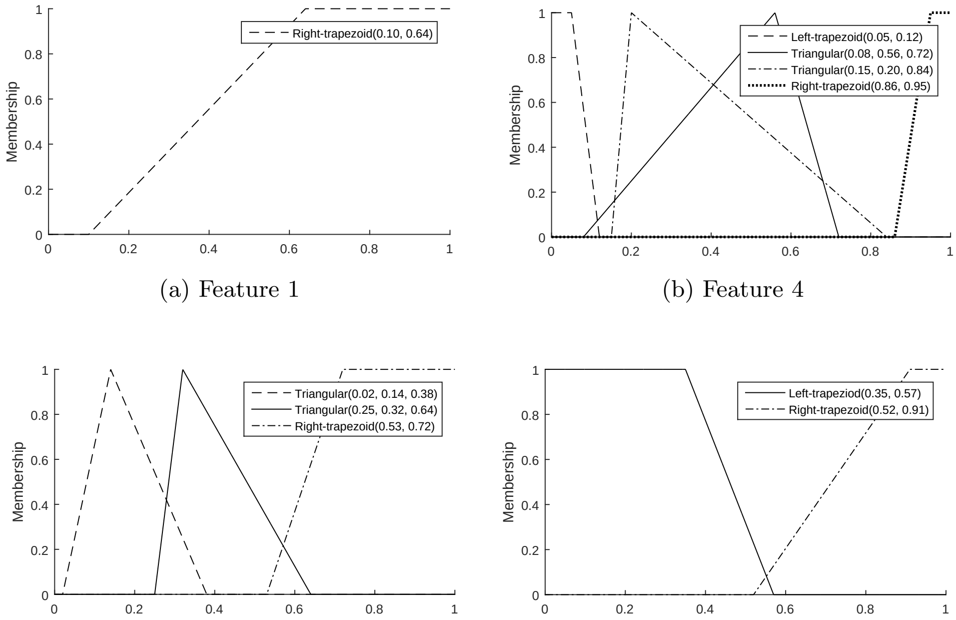
$$\mu_{R_j}(x) = \begin{cases} \mu_{AND}(x, Ca, Cb) & \text{for } Ca \wedge Cb \\ \mu_{OR}(x, Ca, Cb) & \text{for } Ca \vee Cb \\ \mu_{NOT}(x, Ca, Cb) & \text{for } !\,Ca \\ \mu_{SUB}(x, Ca, Cb) & \text{for } Ca - Cb \\ \mu_{\widetilde{Lt}(a,b)}(x) & \text{for left trapezoidal fuzzy set} \\ \mu_{\widetilde{Tr}(a,b,c)}(x) & \text{for triangular fuzzy set} \\ \mu_{\widetilde{Rt}(a,b)}(x) & \text{for right trapezoidal fuzzy set} \end{cases} \tag{3}$$

$$\mu_{AND}(x, Ca, Cb) = \mu_{Ca}(x) \cdot \mu_{Cb}(x) \tag{4}$$

$$\mu_{OR}(x, Ca, Cb) = \mu_{Ca}(x) + \mu_{Cb}(x) - \mu_{Ca}(x) \cdot \mu_{Cb}(x) \tag{5}$$

$$\mu_{NOT}(x, Ca) = 1 - \mu_{Ca}(x) \tag{6}$$

$$\mu_{SUB}(x, Ca, Cb) = \mu_{Ca}(x) - \mu_{Cb}(x) \tag{7}$$



(a) Feature 1

(b) Feature 4

(c) Feature 5

(d) Feature 10

Fig. 2: Representation of the non-uniform fuzzy partitions defined for features 1, 4, 5 and 10 in the rule base depicted in Figure 1

2.2. *Multistage FRBC*

The multistage generalization of the proposed FRBC consists of n stages in increasing order of computational and feature acquisition costs (see Figure 3 for a graphical representation of a 3-stage FRBC). Each stage is a full fuzzy classifier capable of providing an output class for every input. However, everyone but the last stage has the reject option, meaning that if any of these classifiers is not sufficiently certain that their predicted class is the correct one then the responsibility will be passed down to the next stage. To do so, every rule in the intermediate classifiers will not simply return a compatibility grade value, but rather an interval $[\mu_{R_j}(x)-e, \mu_{R_j}(x)+e]$, where e is a learned parameter for the classifier representing the expected error made by the stage due to its limited accuracy. These intervals are sorted incrementally and the highest valued ones are checked for overlapping. If there is a clear winner rule (the highest interval does not overlap with another), then the stage is said to be "certain" and the output predicted class will be the one of the winner rule. On the other hand, if the highest valued intervals overlap, then the stage will take the reject option. Therefore, the output of each intermediate

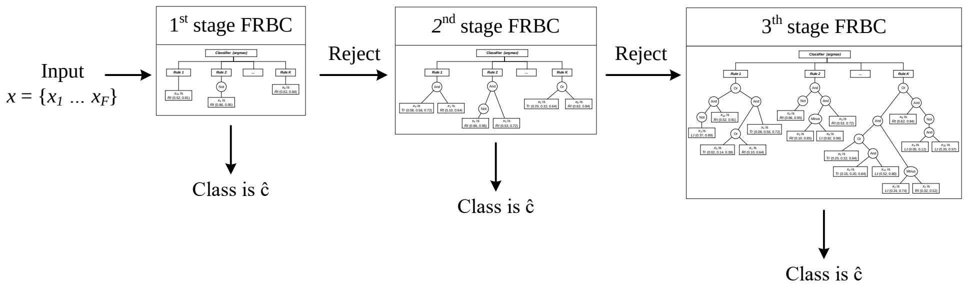


Fig. 3: Representation of a 3-stage FRBC

stage can be expressed as:

$$\text{output} = \begin{cases} \hat{c} = \arg\max_j I_j = [\mu_{R_j}(x) - e, \mu_{R_j}(x) + e] & I_j \cap I_i = \emptyset,\ j \neq i \\ & \forall i \in \{1 \cdots K\} \\ \text{reject} & \text{otherwise} \end{cases} \tag{8}$$

Moreover, to avoid unnecessary computation, if an intermediate stage is certain that a particular class or classes are incorrect (because their compatibly grade intervals are clearly lower than the others), then the rules associated to these classes will not be computed in the next stages. Therefore, if any stage excludes any given class, then the following stages will not consider this class, thus achieving additional cost-savings. It is worth noting that the last stage works in the same manner as the single-stage FRBC described previously.

Learning is done by means of a Pittsburgh-style multiobjective evolutionary algorithm (MOEA),[28] where each individual in the population represents a full n-stage FRBC. This means that all stages, along with its corresponding error parameters, are jointly learned in a single execution of the MOEA algorithm. It is remarked that an incremental learning of n individual classifiers was not performed, due to the tight relation between the confusion matrices of the different stages. In short, if an ordinary incremental learning was applied, then the i-th stage would also be more accurate than the $(i-1)$-th stage, but in the problem being considered this is not the only requirement: the $(i-1)$-th stage must also classify those instances rejected by the latter. By allowing a joint evolution, re-adjustments of subsequent stages as a result of an improvement in their preceding one are performed in a more efficient manner.

Given n stages and K classes, the genotype of an individual is given by the following coding structure:

$$\{(R_1^1 \dots R_K^1)\ \dots\ (R_1^n \dots R_K^n),\ (e^1 \dots e^{n-1})\} \tag{9}$$

where R_j^i is the rule subtree of the i-th stage FRBC defining the compatibility grade of an input instance with the j-th class. Each of these R_j^i rules are valid chains in the context-free grammar detailed in Section 2.1. The e^i parameter is the expected error made by the i-th stage FRBC. Note that there is no e^n parameter as the last classifier has no reject option.

New individuals are generated from the existing ones through mutation operators. In particular, R_j^i rules are structurally mutated by means of a strongly-typed subtree crossover with a randomly generated individual. On the other hand, the chain of numbers $(e^1 \ldots e^{n-1})$ encoding the expected errors is parametrically mutated through an arithmetic crossover with a randomly generated chain the same length.

3. Multiobjective Evolutionary Programming Classifiers

As stated earlier, standard classification schemes blue focus mainly on maximizing classification performance with disregard to the computational costs involved, or the energy consumption of the underlying hardware. Although limits can be imposed to the complexity of the learned classifier (e.g. number of rules, height of a decision tree, number of hidden nodes in a neural network, etc.) or in the size of the feature space, such limits only affect indirectly to the efficiency of the classifier, as these are blind to energy restrictions which are not accounted for during either the feature selection or the learning stage.

Therefore, building efficient classifiers requires classification-related costs to be jointly optimised with classification accuracy. However, these objectives are inherently in conflict with each other. As a result of this, the problem of finding cost-conscious classifiers is a multiobjective one that consists of seeking the set of optimal tradeoffs between accuracy and any related costs. This set, which is denoted as the *Pareto-optimal set*, is often excessively expensive or infeasible to find.[11,53,58,65] Because of this, it is usually approximated through multiobjective optimization algorithms capable of finding a suitable set of solutions without biasing to any objective while also maintaining a sufficient diversity and distribution of Pareto-optimal solutions.[51]

In this paper we adopt a multiobjective form of the Simulated Annealing (SA) global search approach combined with a grammar-tree-based representation to learn a set of non-dominated FRBCs. Essentially, we use Simulated Annealing Programming (SA-P) to find a set of non-dominated FRBC individuals, each one having a tree-shaped genotype of variable size and shape representing the rule base of the FRBC defined in Section 2. The reason why SA was chosen instead of Genetic Programming (GP) is the simplicity of the first. This simplicity was leveraged to avoid both structural and parametric crossover operators, focusing solely on the mutation one, what eases the encoding of numerical parameters in the leaves of the tree instead of in a separate chain aside. The numbers in this chain codify the parameters on which terminal symbols TR, LT and RT depend, i.e. the definition of the fuzzy partitions defining the linguistic variables. This allows the learned FRBCs to use significantly more flexible and compact rule bases through the use non-uniform fuzzy partitions, as described in the grammar presented the preceding section. Further details regarding multiobjective versions of the SA algorithm, can be found in References 50, 7 and 49.

The fitness function is used to assess the quality of each individual by mapping their genotype into an objective vector comprised of multiple values. These values are grouped into two categories: classification accuracy and classification-related costs. The first category of values are used to determine the ability to classify correctly new instances, while the second one accounts for the resources required to perform the feature extraction or the class-inferencing process itself. Note that resources may range from floating-point operations or watts, to even monetary units in certain cases. Due to the potential heterogeneity of these, the fitness function must handle multiple cost-related values as their addition in not always possible.

The classification quality of an individual is defined by three components, as proposed in Reference 56. This triplet (e, m_e, m_c) consists of the error rate over the training set, the error margin and the classification margin. The training error rate e, or resubstitution error,[22] is the proportion of misclassifications made over the whole set of training instances. The error margin m_e gives a measure of the distance between the incorrectly classified instances and their nearest decision surface, having this surface formed by the points where the two FRBC rules with the highest compatibility grades are in a tie. The smaller this margin is, the closer is the FRBC from correctly classifying misclassified instances, and vice versa. On the other hand, the classification margin m_c estimates the distance between the correctly classified instances and their nearest decision surface. Therefore, the higher this latter margin is, the further is the FRBC from mistakenly classifying new instances.

Given any instance x, its individual margin with the decision surface is calculated as the difference between the highest and the second highest compatibility grades of x with the FRBC rules. The global error margin is approximated as the 90 percentile of the individual margins for the training subset of incorrectly classified instances X_e:

$$\begin{aligned} m_e = \min \big\{ p : ||\{x \in X_e,\ j \in 1 \ldots n,\ M^j = \{\mu_{R_0}(x_j) \ldots \mu_{R_K}(x_j)\} : \\ M^j_{(n)} - M^j_{(n-1)} \le p\}|| > 0.9\, n \big\} \end{aligned} \tag{10}$$

where n is the size of the set X_e, $M_{(n)}$ is the n-th order value of the set M and $||A||$ is the cardinality of the set A.

In the same manner, the global classification margin is approximated as the 10 percentile of the individual margins for the training subset of correctly classified instances X_c:

$$\begin{aligned} m_c = \min \big\{ p : ||\{x \in X_c,\ j \in 1 \ldots n,\ M^j = \{\mu_{R_0}(x_j) \ldots \mu_{R_K}(x_j)\} : \\ M^j_{(n)} - M^j_{(n-1)} \le p\}|| > 0.1\, n \big\} \end{aligned} \tag{11}$$

Note that the 90 and 10 percentiles were used instead of the maximum and minimum due to the potential presence of outlier values.

The second group of fitness values consists of a series of costs $\{C_1, \ldots, C_n\}$ that are sought to be minimized in order to reduce the computational, energy and/or monetary-related resources required for the classification.

Finally, let's state that the objective vector z onto which the fitness function maps the genotype of an individual is a set $\{(e, m_e, m_c),\ C_1, C_2 \ldots C_n\}$. As explained in 65, a classifier f^1 is said to dominate another classifier f^2 if z^1 is not worse than z^2 in all components and better in at least one. Regarding the triplet (e, m_e, m_c), the three values are sorted lexicographically, so we define "better" as having a lower error rate. If there is a tie in the error rates, the lower error margins are preferred. If both error rates and error margins are tied, then the one with the highest classification margin is chosen. As for the cost components of the objective vector, "better" is traduced as being smaller.

4. Experimental results

The experimental setup used to assess the performance of the multi-stage FRBC classifier is based on current state-of-the-art hearing aids. These battery-powered devices are capable of automatically identifying the acoustic environment surrounding the user through Sound Environment Classification (SEC), and then tuning the amplification parameters to best user's comfort.[23] In this experiment, sound signals are preprocessed to extract a set of features based on temporal statistics of the Mel Frequency Cepstral Coefficients (MFCCs).[20] Succinctly explained, the goal of this problem is to find a set of optimal tradeoffs between classification accuracy and computational costs, allowing a human expert to choose a suitable solution, achieving good SEC accuracy within the computational and power limitations of the underlying hearing aid device, while also maximising battery life and user comfort. In this scenario, both the feature acquisition and classification costs in the fitness function are measured in terms of floating-point operations. In fact the overall cost of any learned classifier can be computed as:

$$C_T = \frac{2F_s}{N_s}\left(\sum_{m=1}^{F'} C_M(m)\right) + \frac{2F_s F'}{N_s T} C_S + \frac{2F_s}{N_s T} C_C \tag{12}$$

where $C_M(m)$ is the computational cost associated to the computation of the m-th MFCC, C_S is the computational cost associated to the evaluation of the statistics of each MFCC, C_C is the computational cost associated to the classifier, F_s is the sampling frequency, F' is the number of selected MFCCs, and N_s is the number of samples obtained dividing the input sound signals into segments at the first step of the MFCC calculation.

In particular, the same experiment described in References 13 and 14 is analysed. The approach presented in this paper is an evolution of these works, where standard FRBC were applied to balance computational burden and energy consumption of the hardware but the imbalance in the probabilities of the different classes was not exploited with a multi-stage classifier, as done in this paper. The database comprises

2362 seconds of audio including samples of the three considered classes ($N_c = 3$): speech (including speech in a quiet environment, speech in a noisy ambient, and speech combined with music), music (including vocal music and instrumental music) and noise.

The specification of the SA algorithm follows. An exponential annealing schedule with a constant cooling factor is used,

$$T_k = T_0 c^k. \tag{13}$$

The annealing is stopped when the total number of evaluations reaches a limit. Neighbouring states of the current state at each search thread are the result of applying either structural mutation (strongly-typed subtree crossover) or parametric mutation, defined as the arithmetic crossover of the chain p of numbers that encodes the expected errors made by each intermediate stage due to their limited accuracy with a randomly generated chain q of the same length, with a random constant $\alpha \in [0, 1]$,

$$\text{mutation}(p) = \alpha p + (1 - \alpha)q. \tag{14}$$

The specification of these parameters is displayed in the following tableaus:

Parameter	Value	Parameter	Value
Initial temperature T_0	1.00	Cooling factor c	0.9999968
Final temperature	0.10	Evaluations of fitness	$< 10^7$
Structural mutation	0.75	Parametric mutation	0.25
Maximum population size	12	Acceptance probability	1 if $f' \succ f$,
Number of random seeds	300		$e^{-\frac{d(f',f)}{T}}$ else

where f' is a new mutated individual, f is its corresponding predecessor and d is the supremum of the distances in all the components of the objective vector. The number of seeds determines the amount of times the algorithm is rerun using a new and randomly generated initial population. The non-dominated individuals obtained with each seed are then combined according to the domination criterion to produce a single Pareto front. The random initialization of the individuals depends on the following set of parameters:

Parameter	Value
Maximum rule-tree height	7
Node types	OR; AND; MINUS; NOT; LT; TR; RT
Probability of each node	.15 .15 .14 .14 .14 .14 .14
Minimum value of constants	0
Maximum value of constants	1

The maximum tree height sets the maximum distance allowed between any ASSERTION node and the root node of a R_j rule-tree generated with the grammar in Section 2.1. Its purpose is to prevent overfitting in the learned classifiers by limiting their complexity.

Each experiment was repeated 10 times. The experimental design is cross-validation based (10-cv). A graphical view of the average of train and test results is displayed in Figure 4a. These are combined Pareto fronts of the 10 rounds of cross validation.

In order to make a numerical comparison of the results at each round of cross validation, unary indicators could have been used to convert each Pareto front into a representative value, and these values can be compared with standard techniques. However, there are studies supporting the notion that unary indicators cannot show all the dominance relations that can occur between Pareto fronts.[65] In this paper, the binary ϵ-indicator that is used to compare the average improvement between one algorithm and the other is defined as follows: given two Paretos A and B, if $I_\epsilon(A,B) < 1$ and $I_\epsilon(B,A) > 1$ (or if $I_{\epsilon+}(A,B) < 0$ and $I_{\epsilon+}(B,A) > 0$ for additive indexes) we can state that A dominates B. In other words: Let $p_A(B)$ be 1 if A dominates B (i.e. when $I_\epsilon(A,B) > 1$ and $I_\epsilon(B,A) < 1$), 0 otherwise. Given 10 repetitions $B_1,\ldots,B_{10}$ of an algorithm B, let

$$P_A(B) = \frac{1}{10}\sum_{i=1}^{10} p_A(B_i). \tag{15}$$

Given other 10 repetitions $A_1,\ldots,A_{10}$ of an algorithm A, let

$$\mathbf{P}_A(B) = (P_{A_1}(B), P_{A_2}(B), \ldots, P_{A_{10}}(B)). \tag{16}$$

The vector $\mathbf{P}_A(B)$ is the fraction of times that the output of the algorithm A dominates the algorithm B. If the expectation of $\mathbf{P}_A(B)$ is greater than the expectation of $\mathbf{P}_B(A)$, then we can state that the algorithm A is better than the algorithm B, since it is more likely that results of the former improve those of the latter than the opposite. A Wilcoxon test (null hypothesis $E(\mathbf{P}_A(B)) = E(\mathbf{P}_B(A))$, alternate hypothesis $E(\mathbf{P}_A(B)) > E(\mathbf{P}_B(A))$) can then be used to assess the relevance of the differences.

Comparisons were done between the Pareto Efficient Frontier obtained for the multistage FRBC described in Section 2 (labelled "Multi-stage FRBC") and the cost-based filter and ranker feature selection methods proposed in Reference 10 (labelled "Cost-based CFS" and "Cost-based mRMR", respectively), combined with a SVM classifier and evaluated over a set of λ values ranging between 0 and 10 with 10^{-2} increments. Results obtained for each algorithm are displayed in Figure 4 and also summarised in Table 1 showing the number of non-dominated solutions for each dataset and Table 2 showing the ϵ-domination factor according to the I_ϵ binary indicator. In all cases, the p-value of the statistical tests was lower than 0.05.

Observe that the approximation of the Pareto-optimal set achieved with the multi-stage FRBC is significantly better that the ones found with the cost-based filter and ranked methods. In fact, not only it can be appreciated how nearly every solution from the FRBC has either lower cost or error rate, but also how the coverage of the Pareto-optimal set is substantially better, reaching a higher diversity and density in its approximations. Note that Pareto approximations with a higher density (number of solutions) lead to a wider range of tradeoffs between accuracy and costs, further improving the practical use of the learned classifiers by increasing the probability of finding a suitable balance between all objectives. Table 2 gives the factor by which every approximation set is worse than every other one with respect to all objectives.[65] In particular for the test dataset, we can say that that the approximation set found with the cost-based CFS and mRMR are 10.784-dominated and 1.324-dominated, respectively, by the approximation set of the multistage FRBC, thus the later is consistently dominant over the aforementioned feature selection methods.

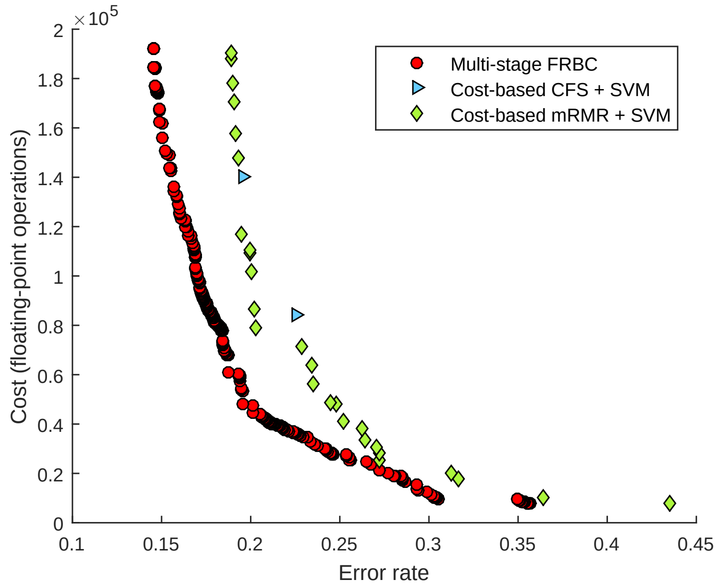
Dataset	Number of non-dominated solutions		
	Multi-stage FRBC	Cost-based CFS	Cost-based mRMR
Training set	**580**	2	27
Test set	**154**	2	20

Table 1: Number of non-dominated solutions for each dataset.

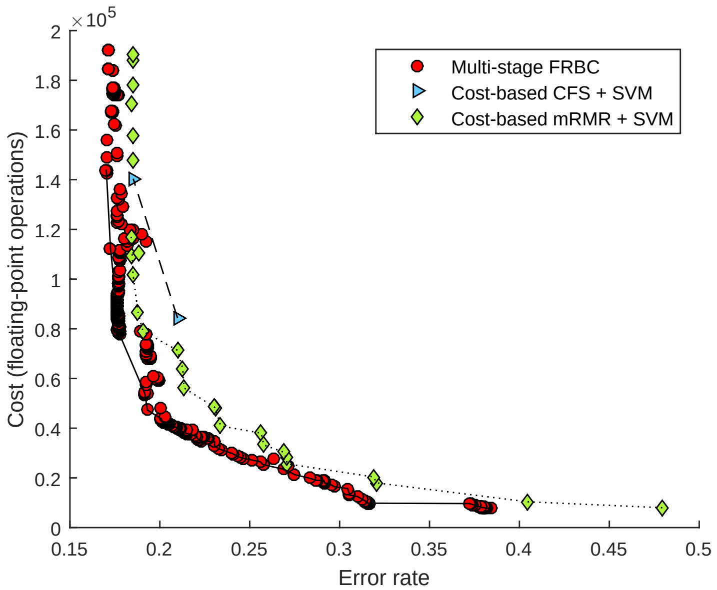
Indicator	Training set	Test set
I_ϵ(Multi-stage FRBC, Cost-based CFS)	**0.842**	**0.932** (0.02)
I_ϵ(Cost-based CFS, Multi-stage FRBC)	10.784	10.784 (0.00)
I_ϵ(Multi-stage FRBC, Cost-based mRMR)	**0.977**	**0.986** (0.01)
I_ϵ(Cost-based mRMR, Multi-stage FRBC)	1.300	1.324 (0.02)
I_ϵ(Cost-based CFS, Cost-based mRMR)	10.531	10.531 (0.00)
I_ϵ(Cost-based mRMR, Cost-based CFS)	**0.996**	**0.994** (0.02)

Table 2: Mean values of the ϵ-domination factor between each pair of classifiers for each dataset. The standard deviations of the test results at each fold are enclosed in parenthesis.

Lastly, Table 3 summarizes the most accurate multistage FRBC found in the experiment along with the least costly one. This table shows how the multi-stage classifier structure proposed jointly optimise cost and classification performance. Let's focus first on the most accurate classifier. It can be appreciated how its first stage has a significantly lower complexity than its second stage in terms of the number of nodes. This leads to reduced costs since a lower node count implies less conditional terms in the antecedent of the rules as well a fewer assertions (leaf nodes), which in turn reduces the cost of feature computation costs. Nevertheless, and given that



(a) Training dataset



(b) Test dataset

Fig. 4: Combined/average error/cost charts displaying the fitness values of each solution in the Pareto Efficient Frontier obtained for each classifier.

this classifier is targeted at maximising accuracy, its simpler first stage is bounded a have a limited certainty to avoid misclassifications, thus rejecting the classification for almost 80% of the input instances. Despite so, 20% of instances were classified with less than half of the cost that would have been required if only the last stage was used (or an equivalent single-stage FRBC), leading to noticeable cost savings. In addition to this, the first instance is also capable of discarding classes so the second stage can focus on only those that are still in question, what further increases the accuracy of the second stage along with the overall classification performance of the multistage FRBC. Note that the learned certainty parameter (the expected error e) is key in achieving a very low misclassification rate in the first stage (0.026) by rejecting those instances for which it is not suitable. It is also noteworthy that the second stage has a higher error rate is than that of the 2-stage FRBC as a whole, since it is tasked with classifying the most difficult instances.

In the case of the learned 2-stage FRBC with the lowest cost, a different behaviour is shown. In order to achieve a greater cost reduction, the certainty parameter (e) is substantially smaller, what further reduces the overall overlapping of the compatibility grade intervals, and thus increases the likelihood that the first stage is “certain”. By doing so, a vast majority of instances are classified in the first stage (over 99.9%) and with a lower classification cost, though at the price of a higher misclassification rate.

The remaining individuals in the Pareto front balance accuracy and cost by learning FRBCs with different degrees of complexity in terms of the rule trees, along with the value of the expected error e. It is noteworthy that the multistage approach offers greater flexibility by tuning the e parameter, hence finding Pareto fronts with a higher density and coverage of the Pareto-optimal set.

5. Concluding remarks and future work

A cost-concious multi-stage fuzzy classifier learned by means of multiobjective evolutionary algorithms has been proposed to jointly optimise classification accuracy along with any cost that must be accounted for. This holistic cost-aware approach based on multicriteria classifiers is capable of finding optimal cost-accuracy tradeoffs in such a way that a human expert can choose suitable solutions befitting his or her subjective preferences under any scenario. This idea is key to maximise the practical application of classification systems in resource-constrained problems such as battery-powered devices. The proposed method has been successfully tested for Sound Environment Classification in modern hearing aids, finding approximation sets with high coverage, diversity and density of Pareto-optimal solutions.

The current approach for implementing the reject option (the difference between the activations of winner and second winner rules) will be revisited in future works, and different structures of the FRBC will be researched. The multiple consequent approach, and its associated techniques for solving conflicts and ignorance are a natural extension,[37] and the same can be said about multi-class TSK rules[4] or soft

Test dataset values	2-stage FRBC samples Lowest error rate	Lowest cost
Multistage FRBC error rate	0.170	0.381
Multistage FRBC classification cost	146121.87	7830.86
Number of nodes in Stage 1	31	6
Number of nodes in Stage 2	123	24
Expected Stage 1 error (e)	0.033	9.7×10^{-5}
1^{st} stage classification cost (FLOPs)	69000	7812.5
2^{nd} stage classification cost (FLOPs)	166375	49125
1^{st} stage classification error rate	0.026	0.382
2^{nd} stage classification error rate	0.208	0.154
Percentage of 1^{st} stage classifications	20.80%	99.96%
Percentage of 2^{nd} stage classifications	79.20%	0.04%

Table 3: Summary of two representative multi-stage FRBCs from the Pareto Efficient Frontier learned in the experiment: the most accurate one and the one with the lowest classification cost.

labels.[9] Finally, a post-hoc reduction of the complexity of the Knowledge Base is possible if certain techniques designed for improving the linguistic interpretability are applied.[12,19,34,38,48]

Acknowledgements

We thank the reviewers for their valuable comments, that have prompted new ideas for future work and contributed to extend this paper to new domains of application.

This work has been partially supported by the Ministry of Economy and Competitiveness ("Ministerio de Economía y Competitividad") from Spain/FEDER under grants TEC2015-67387-C4-3-R, TEC2015-67387-C4-4-R and TIN2014-56967-R and by the Regional Ministry of the Principality of Asturias under grant FC-15-GRUPIN14-073.

References

1. Omar Y. Al-Jarrah, Paul D. Yoo, Sami Muhaidat, George K. Karagiannidis, and Kamal Taha. Efficient Machine Learning for Big Data: A Review. Big Data Research, 2(3):87–93, apr 2015.
2. R Alcalá, M J Gacto, and F Herrera. A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. IEEE Transactions on Fuzzy Systems, 19(4):666–681, 2011.
3. R Alcalá, M J Gacto, F Herrera, and J Alcalá-Fdez. A multi-objective genetic

algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems, 15(5):539–557, 2007.
4. P. Angelov, E. Lughofer, and X. Zhou. Evolving fuzzy classifiers using different model architectures. Fuzzy Sets and Systems, 159(23):3160–3182, dec 2008.
5. D Anguita, A Ghio, L Oneto, X Parra, and J L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7657 LNCS:216–223, 2012.
6. D Anguita, A Ghio, L Oneto, X Parra, and J L Reyes-Ortiz. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. Journal of Universal Computer Science, 19(9):1295–1314, 2013.
7. S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb. A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. IEEE Transactions on Evolutionary Computation, 12(3):269–283, jun 2008.
8. A Betancourt, M M Lopez, C S Regazzoni, and M Rauterberg. A sequential classifier for hand detection in the framework of egocentric vision. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 600–605, 2014.
9. F. Beukema and R. Duin. Dealing with A Priori Knowledge by Fuzzy Labels. Pattern Recognition, 14(1–6):111–115, 1981.
10. V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos. A framework for cost-based feature selection. Pattern Recognition, 47(7):2481–2489, jul 2014.
11. Andreas Bölte and Ulrich Wilhelm Thonemann. Optimizing simulated annealing schedules with genetic programming. European Journal of Operational Research, 92(2):402–416, 1996.
12. J. Casillas, O. Cordón, F. Herrera, and L. Magdalena. Interpretability Issues in Fuzzy Modeling. Springer Verlag, Berlin Heidelberg, 2003.
13. A. Cocaña-Fernández, L. Sánchez, J. Ranilla, R. Gil-Pita, and D. Ayllón. Energy-Efficient Sound Environment Classifier for Hearing Aids Based on Multi-objective Simulated Annealing Programming. Springer International Publishing, 2015.
14. A. Cocaña-Fernández, L. Sánchez, J. Ranilla, R. Gil-Pita, and H. Sanchez-Hevia. Improving learning efficiency in multi-objective simulated annealing programming for sound environment classification. In 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), pages 1–5, Río de Janeiro, Brazil, 2016. IEEE.
15. Bruno Sielly Jales Costa, Plamen Parvanov Angelov, and Luiz Affonso Guedes. Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. Neurocomputing, 150:289–303, 2015.
16. Bruno Sielly Jales Costa, Clauber Gomes Bezerra, Luiz Affonso Guedes, and Plamen Parvanov Angelov. Unsupervised classification of data streams based on Typicality and Eccentricity Data Analytics. In 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pages 58–63, Vancouver, BC, jul 2016. IEEE.
17. C Demir and E Alpaydin. Cost-conscious classifier ensembles. Pattern Recognition Letters, 26(14):2206–2214, 2005.
18. M Fazzolari, R Alcala, Y Nojima, H Ishibuchi, and F Herrera. A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. IEEE Transactions on Fuzzy Systems, 21(1):45–65, 2013.
19. M.J. Gacto, R. Alcalá, and F. Herrera. Interpretability of linguistic fuzzy rule-

based systems: An overview of interpretability measures. Information Sciences, 181(20):4340–4360, 2011.
20. Roberto Gil-Pita, Enrique Alexandre, Lucas Cuadra, Raúl Vicen, and Manuel Rosa-Zurera. Analysis of the Effects of Finite Precision in Neural Network-Based Sound Classifiers for Digital Hearing Aids. EURASIP Journal on Advances in Signal Processing, 2009(1):456945, October 2009.
21. O Giustolisi. Using a multi-objective genetic algorithm for SVM construction. Journal of Hydroinformatics, 8(2):125–139, 2006.
22. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software. ACM SIGKDD Explorations Newsletter, 11(1):10, nov 2009.
23. V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass. Signal Processing in High-End Hearing Aids: State of the Art, Challenges, and Future Trends. EURASIP Journal on Advances in Signal Processing, 2005(18):2915–2929, January 2005.
24. B Hui, Y Yang, and G I Webb. Anytime classification for a pool of instances. Machine Learning, 77(1):61–102, 2009.
25. C Igel. Multi-objective model selection for support vector machines. In Lecture Notes in Computer Science, volume 3410, pages 534–546, 2005.
26. H Ishibuchi and Y Nojima. Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. International Journal of Approximate Reasoning, 44(1):4–31, 2007.
27. Hisao Ishibuchi. Multiobjective Genetic Fuzzy Systems: Review and Future Research Directions. In 2007 IEEE International Fuzzy Systems Conference, pages 1–6. IEEE, jun 2007.
28. Hisao Ishibuchi, Tomoharu Nakashima, and Manabu Nii. Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing). nov 2004.
29. Dmitry Kangin, Plamen Angelov, and José Antonio Iglesias. Autonomously evolving classifier TEDAClass. Information Sciences, 366:1–11, 2016.
30. S S Keerthi, O Chapelle, and D DeCoste. Building support vector machines with reduced classifier complexity. Journal of Machine Learning Research, 7:1493–1515, 2006.
31. R Kohavi and G H John. Wrappers for feature subset selection. Artificial Intelligence, 97(1-2):273–324, 1997.
32. Ó D Lara and M A Labrador. A survey on human activity recognition using wearable sensors. IEEE Communications Surveys and Tutorials, 15(3):1192–1209, 2013.
33. Yuh-Jye Lee and Olvi L Mangasarian. RSVM: Reduced Support Vector Machines. In SDM, volume 1, pages 325–361, 2001.
34. Gang Leng, Xiao-Jun Zeng, and John A. Keane. An improved approach of self-organising fuzzy neural network based on similarity measures. Evolving Systems, 3(1):19–30, mar 2012.
35. L Li, U Topkara, B Coskun, and N Memon. CoCoST: A computational cost sensitive classifier. In Proceedings - IEEE International Conference on Data Mining, ICDM, pages 268–277, 2009.
36. Chen Lin, Wenqiang Chen, Cheng Qiu, Yunfeng Wu, Sridhar Krishnan, and Quan Zou. LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy. Neurocomputing, 123:424–435, 2014.
37. Edwin Lughofer. Single-pass active learning with conflict and ignorance. Evolving Systems, 3(4):251–271, 2012.

38. Edwin Lughofer. On-line assurance of interpretability criteria in evolving fuzzy systems Achievements, new concepts and open issues. Information Sciences, 251:22–46, 2013.
39. G Martínez-Muñoz and A Suárez. Using boosting to prune bagging ensembles. Pattern Recognition Letters, 28(1):156–165, 2007.
40. Taiwoo Park, Jinwon Lee, Inseok Hwang, Chungkuk Yoo, Lama Nachman, and Junehwa Song. E-Gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11, page 260, New York, New York, USA, 2011. ACM Press.
41. J Pärkkä, L Cluitmans, and M Ermes. Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree. IEEE Transactions on Information Technology in Biomedicine, 14(5):1211–1215, 2010.
42. S Patel, K Lorincz, R Hughes, N Huggins, J Growdon, D Standaert, M Akay, J Dy, M Welsh, and P Bonato. Monitoring motor fluctuations in patients with parkinsons disease using wearable sensors. IEEE Transactions on Information Technology in Biomedicine, 13(6):864–873, 2009.
43. P Pierleoni, L Pernini, A Belli, and L Palma. An android-based heart monitoring system for the elderly and for patients with heart disease. International Journal of Telemedicine and Applications, 2014, 2014.
44. Mahardhika Pratama, Sreenatha G. Anavatti, Meng Joo, and Edwin David Lughofer. pClass: An Effective Classifier for Streaming Examples. IEEE Transactions on Fuzzy Systems, 23(2):369–386, apr 2015.
45. J Pujara, H Daumé III, and L Getoor. Using classifier cascades for scalable e-mail classification. In ACM International Conference Proceeding Series, pages 55–63, 2011.
46. M Ring and B M Eskofier. An approximation of the Gaussian RBF kernel for efficient classification with SVMs. Pattern Recognition Letters, 84:1339–1351, 2016.
47. T.E. Senator. Multi-Stage Classification. In Fifth IEEE International Conference on Data Mining (ICDM'05), pages 386–393. IEEE, 2005.
48. M Setnes, R Babuška, and H.B Verbruggen. Complexity reduction in fuzzy modeling. Mathematics and Computers in Simulation, 46(5):507–516, 1998.
49. K.I. Smith, R.M. Everson, J.E. Fieldsend, C. Murphy, and R. Misra. Dominance-Based Multiobjective Simulated Annealing. IEEE Transactions on Evolutionary Computation, 12(3):323–342, jun 2008.
50. B Suman and P Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society, 57(10):1143–1160, oct 2006.
51. Balram Suman. Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. Computers & Chemical Engineering, 28(9):1849–1871, 2004.
52. F.-T. Sun, C Kuo, and M Griss. PEAR: Power efficiency through activity recognition (for ECG-based sensing). In 2011 5th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2011, pages 115–122, 2011.
53. Ozan Tekinalp and Gizem Karsli. A new multiobjective simulated annealing algorithm. Journal of Global Optimization, 39(1):49–77, jul 2007.
54. Kirill Trapeznikov, Venkatesh Saligrama, and David Castañón. Multi-Stage Classifier Design. In JMLR W&CP, volume 25, pages 459–474, 2012.
55. K Trawiński, O Cordón, A Quirin, and L Sánchez. Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is

the additional diversity? Knowledge-Based Systems, 54:3–21, 2013.
56. Krzysztof Trawinski, Oscar Cordon, Luciano Sanchez, and Arnaud Quirin. A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers. IEEE Transactions on Fuzzy Systems, 21(5):950–965, oct 2013.
57. Swagath Venkataramani, Anand Raghunathan, Jie Liu, and Mohammed Shoaib. Scalable-effort classifiers for energy-efficient machine learning. In Proceedings of the 52nd Annual Design Automation Conference on - DAC '15, pages 1–6, New York, New York, USA, jun 2015. ACM Press.
58. Stefan Voß. Meta-heuristics: The State of the Art. In Alexander Nareyek, editor, Local Search for Planning and Scheduling, pages 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
59. Ian H. Witten, Eibe Frank, and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, 2011.
60. Z Xu, M J Kusner, K Q Weinberger, and M Chen. Cost-sensitive tree of classifiers. In 30th International Conference on Machine Learning, ICML 2013, number PART 1, pages 133–141, 2013.
61. Z E Xu, M J Kusner, K Q Weinberger, M Chen, and O Chapelle. Classifier cascades and trees for minimizing feature evaluation cost. Journal of Machine Learning Research, 15:2113–2144, 2014.
62. Bing Xue, Mengjie Zhang, and Will N Browne. Particle swarm optimization for feature selection in classification: a multi-objective approach. IEEE transactions on cybernetics, 43(6):1656–71, dec 2013.
63. L Zhang, J Liu, H Jiang, and Y Guan. SensTrack: Energy-efficient location tracking with smartphone sensors. IEEE Sensors Journal, 13(10):3775–3784, 2013.
64. Y Zhang, S Burer, and W N Street. Ensemble pruning via semi-definite programming. Journal of Machine Learning Research, 7:1315–1338, 2006.
65. E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, 7(2):117–132, apr 2003.

Título

A software tool to efficiently manage the energy consumption of HPC clusters

Autores

Alberto Cocaña-Fernández, Luciano Sánchez and José Ranilla

Journal

Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Páginas 1–8, Istanbul, 2017*

DOI: 10.1109/FUZZ-IEEE.2015.7338079

Ranking

Rank CORE 2017: A

Rank CORE 2014: A

Rank CORE 2013: A

Rank ERA 2010: A

Rank CORE 2008: A

Categorías:

Artificial Intelligence and Image Processing

A software tool to efficiently manage the energy consumption of HPC clusters

Alberto Cocaña-Fernández
Departamento de Informática
Universidad de Oviedo
Gijón, Spain
Email: cocanaalberto@gmail.com

Luciano Sánchez
Departamento de Informática
Universidad de Oviedo
Gijón, Spain
Email: luciano@uniovi.es

José Ranilla
Departamento de Informática
Universidad de Oviedo
Gijón, Spain
Email: ranilla@uniovi.es

***Abstract*—Today, High Performance Computing clusters (HPC) are an essential tool owing to they are an excellent platform for solving a wide range of problems through parallel and distributed applications. Nonetheless, HPC clusters consume large amounts of energy, which combined with notably increasing electricity prices are having an important economical impact, forcing owners to reduce operation costs. In this work we propose a software, named EECluster, to reduce the high energy consumption of HPC clusters. EECluster works with both OGE/SGE and PBS/TORQUE resource management systems and automatically tunes its decision-making mechanism based on a machine learning approach. The quality of the obtained results using this software are evaluated by means of experiments made using actual workloads from the Scientific Modelling Cluster at Oviedo University and the academic-cluster used by the Oviedo University for teaching high performance computing subjects.**

I. Introduction

As High Performance Computing (HPC) clusters become the main architecture for supercomputers (see Top500 architecture distribution[1]) because of the lower price/performance ratio, the performance of current commodity microprocessors, and the availability of standard parallel and distributed computing tools [1], their high energy consumptions have grown to a major issue. The consequence of using these power-hungry computing elements, whose aggregated consumption in the U.S. in 2013 is estimated in 91 billion kilowatt-hours [2], is a great economical impact for IT companies in terms of operation costs due to electricity bills [3], [4]. Moreover, this brings along an environmental impact equivalent to the industry of aviation, as clusters are responsible for the emission of 100 million metric tons of CO_2 every year [5].

These figures show accurately the magnitude of the problem at hand, which itself is the essential bottleneck constraining the expansion of supercomputing and data centres, building, therefore, an unyielding need to maximize the efficiency of these clusters to reduce operation costs, carbon footprint and to improve their reliability. Many attempts have been made over the last years in the pursuit of energy-efficient cluster computing, following both static and dynamic approaches. Static approaches are essentially the development of compute hardware targeting maximum FLOPS/watt instead of just absolute performance, while the dynamic ones seek to adapt the cluster to the current workload, saving energy when the cluster is underused by shutting down or downspeeding components. Examples of this are the Dynamic Voltage and Frequency Scaling (DVFS) technique [6]–[13], energy-efficient job schedulers [14], [15], thermal-aware methods [16], [17], or the development of energy-efficient applications [18]–[22].

Nevertheless, the technique discussed in this paper is the adaptive resource cluster, which consists of the automatic reconfiguration of the cluster resources to fit the workload at every moment by switching on or off its compute nodes, thus saving energy whenever these are idle. This technique already has been applied to Load-Balancing clusters in [23]–[27] and in VMware vSphere[2] and Citrix XenServer[3] hypervisors. Recently various software tools implementing this technique in HPC clusters have also been developed [28]–[30]. However, we consider that these solutions feature important limitations constraining their practical application along with the degree of optimality of the results achieved in terms of energy savings.

The reason for this is that these tools use as decision-making mechanisms closed sets of expert-defined rules, which do not necessarily adapt well to all cluster environments and desired working modes. These mechanisms also lack of a sufficient degree of flexibility to suit the administrator tolerance in terms of impact on QoS and node thrashing while maximizing energy savings. Moreover, and as shown in [31], the tuning of the parameters ruling these Knowledge-based Systems is a key part in obtaining admissible results. If the tuning is unsuitable, the software tool could interfere with the cluster operation reducing notably its effective productivity.

Because of this, we propose the software tool EECluster to transform OGE/SGE and PBS/TORQUE-based HPC clusters into energy-efficient adaptive resource clusters. To do so, EECluster uses a Hybrid Genetic Fuzzy System as the decision-making mechanism that elicits part of its rule base dependent on the cluster workload scenario, delivering good compliance with the administrator preferences. Table I shows a comparison of the main features between EECluster and the alternative software tools.

The remainder of the paper is as follows. Section II explains the architecture of the EECluster tool. Section III

[1]November 2014 — TOP500 Supercomputer Sites, http://www.top500.org/lists/2014/11/

[2]VMware Distributed Power Management Concepts and Use, http://www.vmware.com/files/pdf/Distributed-Power-Management-vSphere.pdf

[3]Citrix XenServer - Efficient Server Virtualization Software, http://www.citrix.com/products/xenserver/overview.html

TABLE I. COMPARISON OF SOFTWARE TOOLS FOR ADAPTIVE RESOURCE CLUSTERS

	EECluster	CLUES [28]	EnergySaving Cluster [29]	Cherub [30]
Compatible Resource Management Systems	OGE/SGE PBS/TORQUE	OGE/SGE PBS/TORQUE OpenNebula	only OGE/SGE	only PBS/TORQUE
Administration dashboard	Web-based	Web-based	Web-based	None
Decision-making mechanism	Hybrid Genetic Fuzzy System	Expert-defined rules	Expert-defined rules	Expert-defined rules
Parameter tuning	Genetic-based machine leaning multiobjective evolutionary algorithm	Manually	Manually	Manually
Power management	Ethernet Wake On Lan IPMI	Ethernet Wake On Lan IPMI PDU	Ethernet Wake On Lan	Ethernet Wake On Lan IPMI

explains how EECluster can be installed and used. Section IV discuss some use cases. Section V shows some experimental results. Section VI concludes the paper and discusses the future work.

II. ARCHITECTURE

The underlying architecture of ordinary HPC clusters combines a master node and several compute nodes. The master node is the one directly accessible by the cluster users through a remote connection such as SSH and where they submit jobs to the Resource Management System (RMS). This system is the software responsible of scheduling jobs execution across the compute nodes that, once jobs are dispatched and gets assigned slots, actually run the job. It is noteworthy that slots are a logical representation of each computing resource and, depending on the RMS configuration, can represent from a single CPU core to a an entire host.

The EECluster tool is a software solution to transform ordinary HPC clusters running OGE/SGE or PBS/TORQUE as RMS into adaptive resource clusters that can dynamically reconfigure its components to suit the cluster workload at every moment by powering on or shutting down compute nodes as required, saving sustainable amounts of energy automatically. To do so, EECluster is comprised of a service (EEClusterd) and an administration dashboard, coupled with a Database Management System (DBMS) as the persistence system. The mission of the service is the periodic retrieval of information regarding the cluster status and then use the decision-making mechanism to reconfigure the compute nodes by issuing a set of power-on or shutdown commands through the Power Management module. The learning algorithm assists the administrator in tuning the Hybrid Genetic Fuzzy System implementing the decision-making mechanism by finding a set of optimal configurations. The administration dashboard is a web application deployed on an application server that allows the cluster administrator to remotely access the information of the cluster status, nodes, job records, users and statistics, reconfigure manually compute nodes, and to tune the parameters that rule the Hybrid GFS.

Figure 1 provides a high-level overview of the system components. The working cycle of the EEClusterd service is the following:

1) Synchronize with the system current status

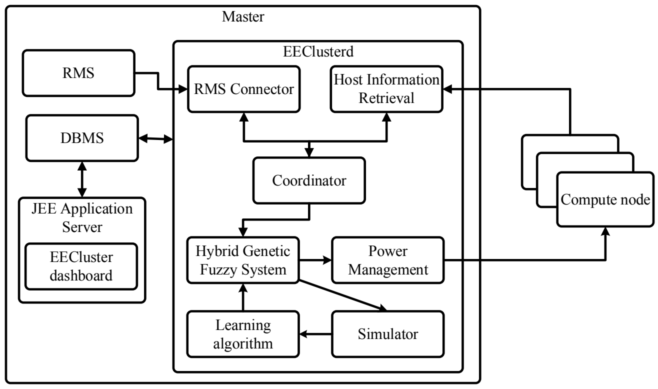


Fig. 1. EECluster Tool: System components overview

2) Use the Hybrid Genetic Fuzzy System to determine if any reconfiguration of the compute nodes must be performed
3) Select the target nodes to reconfigure according to the criteria detailed in Section II-D
4) Issue power-on/off commands to the selected nodes through the Power Management module

Each step is explained in the following subsections.

A. Synchronization

The synchronization task of the service collects and keeps updated records of the RMS and of every compute node. This information is retrieved by the service through the RMS connector, which relies on multiple command line applications bundled with each RMS. In the case of OGE/SGE the synchronization module uses

- *qhost* to get node information on its architecture, number of processors, sockets, cores, load, etc., and its relation with each cluster queue
- *qconf* to get queue and parallel environment info
- *qacct* to retrieve user and job accounting data
- *qstat* to get running and queued jobs status

As for PBS/TORQUE

- *pbsnodes* to get node info
- *qstat* for running and queued jobs status, and also queue info
- *PBS/TORQUE accounting records* to retrieve accounting records for completed jobs in the TORQUE-ROOT/server_priv/accounting/TIMESTAMP directory

Regarding each host, the Host Information Retrieval module collects information about

- CPUs (model, frequency, cache) through the */proc/cpuinfo* file
- RAM memory through the */proc/meminfo* file
- GPUs (name, % utilization, temperature, fan speed, power usage, etc.) through NVIDIA System Management Interface
- Intel Xeon Phi coprocessors (model, active cores, frequency, memory, temperature) through *micinfo*
- PSU power usage through the IPMI interface
- Host MAC address through *arp*

Once information on every cluster compute node is retrieved, their situation in regard to both the EEClusterd service and the cluster RMS is represented through a number of defined states, depicted in Figure 2. Succinctly explained, when a node is sent a power-on command, it changes its state to Starting. If the node fails to power on before a preset timeout period expires, the node returns to the powered off state and the timestamp of this failure is recorded. If the node does power on correctly, it changes its state to Running idle or Executing job depending on whether the RMS has dispatched a job to the node or not. Shutting down a node follows a similar process but with the additional intermediate state Unavailable, which implies that the node has been disabled in the RMS, thus preventing it from dispatching any job while the node is powering off.

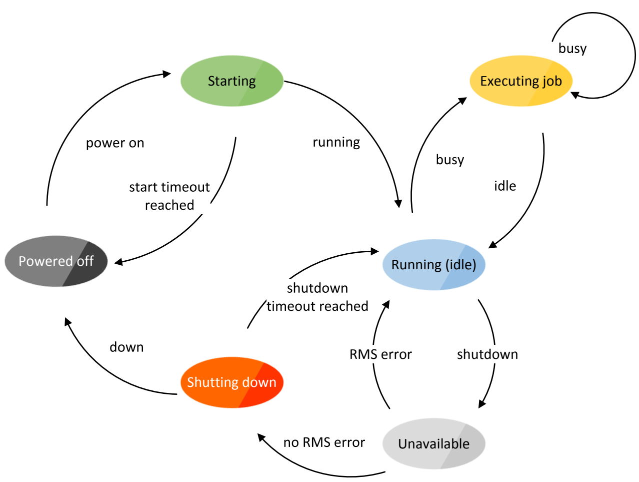


Fig. 2. Node state diagram

B. Hybrid GFS

The key component of this architecture is the Hybrid Genetic Fuzzy System (HGFS) implementing the decision-making mechanism that determines how many of the cluster compute nodes must be on and off at every moment. This system is comprised of both a set of crisp rules adapted from reference [29] and a set of fuzzy rules in the form of a zero-order Takagi-Sugeno-Kang (TSK) model [32], [33].

On the one hand, the set of crisp rules is defined as follows:

> **if** $s_{running} + s_{starting} < s_{min}$ **then** power on $(s_{min} - (s_{running} + s_{starting}))$ slots
>
> **if** $t_{avg} > t_{max}$ or $n_{queued} > n_{max}$ **then** power on 1 slot
>
> **if** $t_{avg} < t_{min}$ or $n_{queued} < n_{min}$ **then** power off 1 slot

$s_{running}$ and $s_{starting}$ are the number of slots currently running and starting. s_{min} is the minimum number of slots required to run each of the queued jobs, that is, the maximum requested slots of an individual job among the queued ones. s_{total} are the cluster total slots (running and powered off). t_{avg} is the average waiting time for the queued jobs, and t_{max} and t_{min} are, respectively, the maximum and minimum average waiting time for the queued jobs. n_{queued} is the number of queued jobs, n_{max} and n_{min} are the maximum and minimum number of queued jobs before an action is performed.

On the other hand, the TSK fuzzy model is comprised of the following rules:

> **if** t is $\widetilde{T}_1$ **then** off $= w_1$
> **if** t is $\widetilde{T}_2$ **then** off $= w_2$
> **if** $\cdots$ **then** $\cdots$
> **if** t is $\widetilde{T}_N$ **then** off $= w_N$

The fuzzy sets $\widetilde{T}_1, \ldots, \widetilde{T}_N$ have a triangular membership function and define a uniform fuzzy partition of the input variable t [32]. The defuzzified output of this model is

$$\text{defuzz}(t) = \frac{\sum_{r=1}^{N} \widetilde{T}_r(t) \cdot w_r}{\sum_{r=1}^{N} \widetilde{T}_r(t)}. \tag{1}$$

The fuzzy model is used to assign a weight to each node. These weights are added to compute the number of actually off hosts. The number of nodes in the cluster that are powered off at a certain time is therefore given by the sum of the outputs of the fuzzy model for all values of idle_h, where idle_h is the time that the host h has been at idle state:

$$\text{Powered off nodes} = \left\lfloor \sum_{h=1}^{c} \text{defuzz}(\text{idle}_h) \right\rfloor. \tag{2}$$

This particular fuzzy model was introduced in [34], and is intended to refine previous schemes where nodes were shut down on a host-by-host basis.

The operation of the fuzzy model is illustrated with the following example: Let N be 5, and $\widetilde{T}_1, \ldots, \widetilde{T}_N$ be "VERY SHORT", "SHORT", "MEDIUM", "LONG" or "VERY

LONG". The values $w_1, \dots, w_N$ are between 0 and 1 and can be understood as the degree of truth of the assert "the h-th node must be switched off". For instance, if the idle time on node 1 is "VERY HIGH" but the idle time in nodes 2 and 3 is "MEDIUM", the weight of the first node will be one, and the weights of nodes 2 and 3 will be 0.5, thus the total number of powered off nodes will be $1 + 0.5 + 0.5 = 2$. This is in contrast with the behaviour of the crisp rule base in reference [29], where the number of nodes would be $1 + 0 + 0 = 1$ if the idle times of nodes 2 and 3 were slightly lower than idle$_{max}$ but would jump from 1 to $1 + 1 + 1 = 3$ if the idle times of nodes 2 and 3 grew by a small amount and idle$_{max}$ was surpassed. As mentioned, the fuzzy model defined in [34] and implemented in the software being described in this paper does not suffer from this problem and allows for a gradual increasing of the number of nodes.

C. Learning algorithm

The advantage of the Hybrid GFS is the ability to adapt to any desired working mode for the cluster due to the many configuration parameters that rule its operation. However, finding the right set of values to match the desired working mode is far from trivial. To address this problem, the EECluster learning algorithm uses multiobjective evolutionary algorithms (MOEAs) to find the parameters defining the HGFS, by optimizing a fitness function consisting in three conflicting criteria: the quality of service, the energy saved and the number of node reconfigurations. Specifically, EECluster uses the MOEA Framework [35] implementation of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [36].

For a given set of n jobs, where the j-th job ($j = 1 \dots n$) is scheduled to start at time tsch$_j$, but effectively starts at time ton$_j$ and stops at time toff$_j$, the quality of service in a HPC cluster reflects the amount of time that each job has to wait before is assigned its requested resources. Once the job starts its execution, it will not be halted, thus we focus only on its waiting time. Because jobs do not last the same amount of time, their waiting in the queue is better expressed as a ratio considering their execution time. Finally, due to the potential existence of outlier values, the 90 percentile is used instead of average:

$$\mathrm{QoS} = \min\left\{p : ||\{j \in 1 \dots n : \frac{\mathrm{ton}_j - \mathrm{tsch}_j}{\mathrm{toff}_j - \mathrm{ton}_j} \le p\}|| > 0.9\, n\right\} \tag{3}$$

where $||A||$ is the cardinality of the set A.

The energy saved is measured as the sum of the amount of seconds that each node has been powered off. Let c be the number of nodes, let state(i, t) be 1 if the i-th node ($i = 1 \dots c$) is powered at time t and 0 otherwise. Lastly, let the time scale be the lapse between tini=$\min_j\{\mathrm{sch}_j\}$ and tend= $\max_j\{\mathrm{toff}_j\}$. Then,

$$\text{Energy saved} = c \cdot (\mathrm{tend} - \mathrm{tini}) - \sum_{i=1}^{c} \int_{\mathrm{tini}}^{\mathrm{tend}} \mathrm{state}(i, t)\mathrm{d}t. \tag{4}$$

The node reconfigurations is the number of times that a node has been powered on or off. Let nd(i) the number of discontinuities of the function state(i, t) in the time interval $t \in (\mathrm{tini}, \mathrm{tend})$:

$$\text{Reconfigured nodes} = \sum_{i=1}^{c} \mathrm{nd}(i) \tag{5}$$

A particular instance of the hybrid GFS can be expressed as a combination of the following parameters:

$$(t_{min}, t_{max}, n_{min}, n_{max}, \widetilde{T}_1, \dots, \widetilde{T}_N, w_1, \dots, w_N). \tag{6}$$

The mission of the NSGA-II algorithm is to obtain a set of non-dominated configuration of the HGFS, guided by the previous fitness function, whose values are calculated by running a cluster simulation with a given number of nodes, slots and job records, as seen in Figure 3.

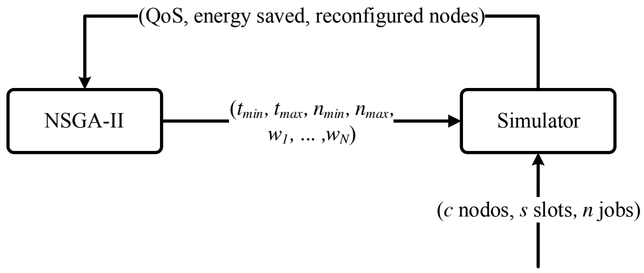


Fig. 3. EECluster learning process

It is remarked that in the current version of the learning algorithm the membership functions $\widetilde{T}_1, \dots, \widetilde{T}_N$ are not adjusted and a uniform partition is defined, but this is not a fundamental limitation; in our experimentations, any change in the membership function of these sets could be compensated by the corresponding modification in the weights w_i.

D. Node selection

Once the Hybrid GFS has determined how many slots must be powered on/off, a set of specific nodes is selected for reconfiguration accordingly to the following criteria that involves two values: the node efficiency calculated as $\frac{\mathrm{GFLOPS}}{\mathrm{Watts}}$, and the node timestamp of the last time it failed to comply with a power on/off command upon request. Firstly, hosts are split in two subgroups depending on whether they succeeded or failed to comply with the last order. Those that succeeded are sorting accordingly to their efficiency so that the most efficient are always the chosen ones to power on and the least are the ones chosen to power off. Conversely, those that failed are sorted according to the timestamps of their failures; those with the earliest values are always chosen. The idea driving this selection criteria is allowing the system to continuously iterate through the potentially malfunctioning nodes, thus increasing the possibility of finding a repaired one.

E. Power Management

After the target nodes for reconfiguration have been appointed, switch-on/off commands are sent to these nodes using the Power Management module, which can either employ Ethernet cards or IPMI cards (Intelligent Platform Management Interface). In the case of Ethernet cards, power-on is done by sending the Ethernet WOL (Wake On Lan) *magic packet* using the *ether-wake* program and selecting the network interface of the master node that leads to the same network that the target compute node is. Shutting down is simply done executing the *poweroff* command after assuring that the compute node WOL is enabled using *ethtool*. If the host features an IPMI card, the Power Management module can also use it to power it on/off using *ipmiutil*.

III. Using EECluster

Prior to its use, the EECluster tool requires the installation and configuration of the EEClusterd service and the administration dashboard along with a DBMS and a Java EE application server. This is done following the next steps:

1) Install a JDBC-compatible DBMS such as MySQL
2) Execute the provided SQL script to create the tables for both the EECluster data model and the Java EE application server security realm
3) Create the users for the security realm in the database
4) Grant select, insert, update and delete privileges on the database previously created to a DBMS user
5) Extract EEClusterd tarball into a directory and edit the configuration files in the *conf* folder. It is essential to configure the cluster RMS (OGE/SGE or PBS/TORQUE) and the DBMS url, JDBC driver and user credentials
6) Install a Java EE application server and configure the JDBC resources, connection pool and security realms required for the administration dashboard
7) Deploy the administration dashboard into the chosen domain of the application server
8) Start the EEClusterd service and the Java EE application server
9) (Optional) Copy into the *init.d* folder the scripts to automatize the start-up upon system boot of both the EEClusterd and the application server

Once installed and running, EECluster will start synchronizing with the RMS records. This information can be accessed through the administration dashboard via web browser. The main page (Figure 5) displays the status and load of each of the compute nodes, and the queued and running jobs. This page also allows the administrator to power on or off manually every compute node and to view all its information.

The compute node page (Figure 6) displays its Operating System, CPUs, Memory, GPU cards, Intel MIC cards, PSU power consumptions, load, etc., and also allows the administrator to configure the IPMI card of the node.

The node classes page (Figure 7) is used to configure a number of compute node classes, each one including a name, brief description, picture, FLOPS, power consumption (which is used to calculate the node efficiency if no IPMI card is available to monitor PSU actual consumptions), and the subset of the cluster nodes that belong to the class.

The users page displays information about every cluster user that has submitted a job, along with dynamically generated histograms of the wallclock time, memory, and input/output operations employed by the users.

The jobs page (Figure 8) displays information about every job submitted to the RMS, and also shows charts that summarises execution times, slots requested and the distribution of sequential, shared memory and distributed jobs.

The statistics page displays records of every decision that the HGFS has made, along with charts showing the evolution of running, used, and requested slots over the last hour, day and week. It is noteworthy that every table in the dashboard can generate a report with its values in XLS, PDF, CSV, and XML formats.

Lastly, the configuration page is where the HGFS is configured. As mentioned earlier, the EECluster tool assists the administrator by running the learning algorithm with the number of nodes and slots of the current cluster setup in a workload scenario characterised by a sublist of jobs from the RMS records to obtain a set of non-dominated configurations (Pareto Efficient Frontier) for the HGFS. The administrator then chooses the best configuration according to his preferences and sets the values in the configuration page of the EECluster dashboard (see Figure 4)

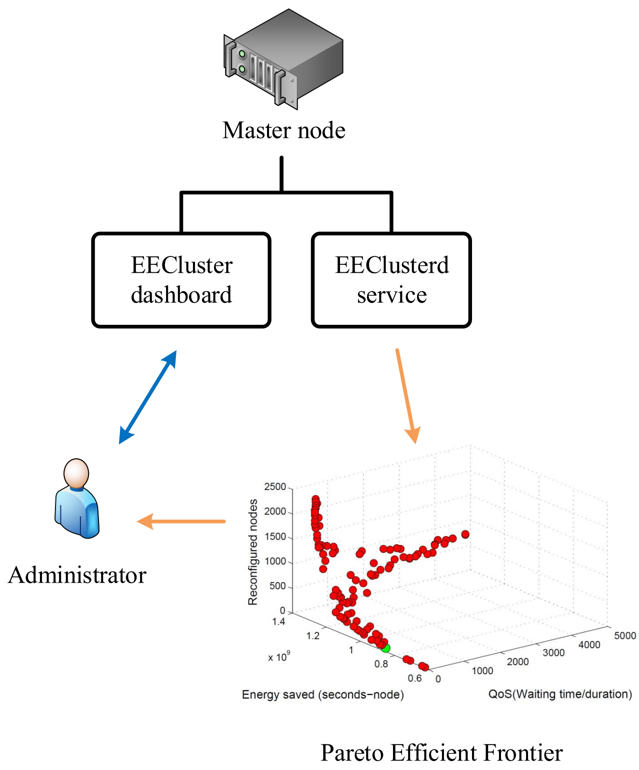


Fig. 4. Configuration of the EECluster tool

IV. Use cases

The EECluster tool has been tested in various environments including research, professional and academic clusters.

The research cluster consists in 5 computational nodes arranged in two queues running OGE/SGE as RMS. These nodes include two PowerEdge 1950 servers and one PowerEdge 2950, all with one Intel Xeon CPU E5420 @ 2.5 GHz and 16

Fig. 5. EECluster dashboard's main page

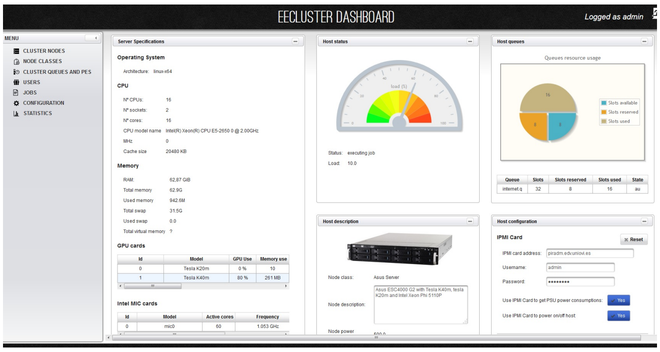


Fig. 6. Compute nodes page in the EECluster dashboard

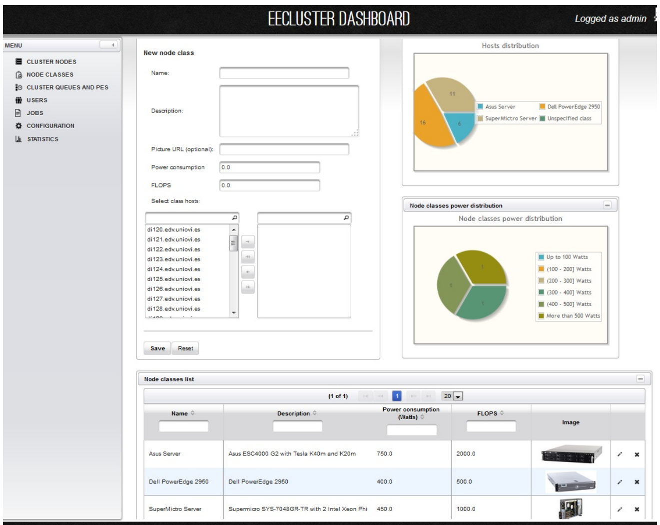


Fig. 7. Compute node classes in the EECluster dashboard

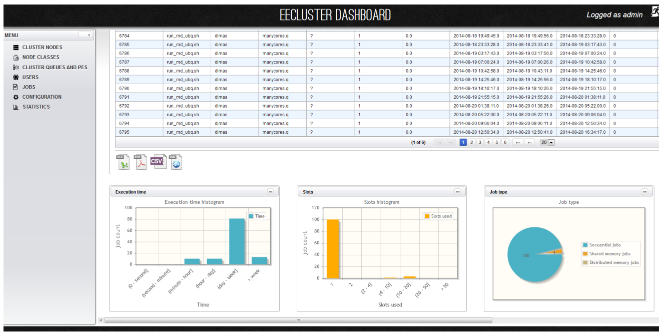


Fig. 8. Jobs page in the EECluster dashboard

GB of RAM, one ASUS server with two Intel Xeon CPU E5-2650 @ 2.0 GHz, 64 GB of RAM, one NVIDIA Tesla K40m, and one NVIDIA Tesla K20m, and one Supermicro server with two Intel Xeon CPU E5-2603 v3 @ 1.6 GHz, 32 GB of RAM, one Intel Xeon Phi 5110P, and two Intel Xeon Phi 31S1P. This cluster is used to support research in the field of algorithm parallelization in multicore, distributed, GPUs and Intel MIC environments, and also chemical computational modelling.

Another research and professional cluster is the Scientific Modelling Cluster of the University of Oviedo (CMS) which consists of three independent computing clusters and five transversal queues using PBS as a Resource Management Systems (RMS). Further information on the CMS can be found in its web site (http://cms.uniovi.es).

The academic cluster is a 34-node OGE/SGE cluster used for educational purposes, allowing students to learn and experiment with multicore, distributed and GPU computing. The nodes include both PCs with Intel Core i3-2100 CPUs @ 3.10 GHz and 4 GB of RAM, and PCs with Intel Core i7 930 CPUs @ 2.80 GHz, 12 GB of RAM and CUDA-enabled NVIDIA GeForce GTX 480 cards. The compute nodes are arranged in three different queues and their Ethernet interfaces are used to switch them on. Also, the nodes are arranged in three different queues and in different networks, what requires explicit configuration in the dashboard to specify the network interface that must be used in the master node to send the WOL packet to each node.

V. Experimental Results

To measure the effect of the EECluster tool in a real world environment and its flexibility to adapt to any desired working mode, an experiment was done using actual workloads of the aforementioned Scientific Modelling Cluster of the University of Oviedo spanning 22 months with a total of 2907 jobs, and three different configurations for the HGFS, each one corresponding to a different set of administrator preferences. The first configuration (labelled as HGFS QoS 0.0) depicts a scenario where the cluster administrator priorities QoS above all other criteria, and where energy savings are just used to break ties between QoS. The second (labelled as HGFS QoS 0.1) seeks the best energy savings as long the QoS value is below or equal to 0.1. The third one (labelled as HGFS QoS 0.5) rises the QoS boundary to 0.5. In all cases, node reconfigurations are used to break ties between energy savings.

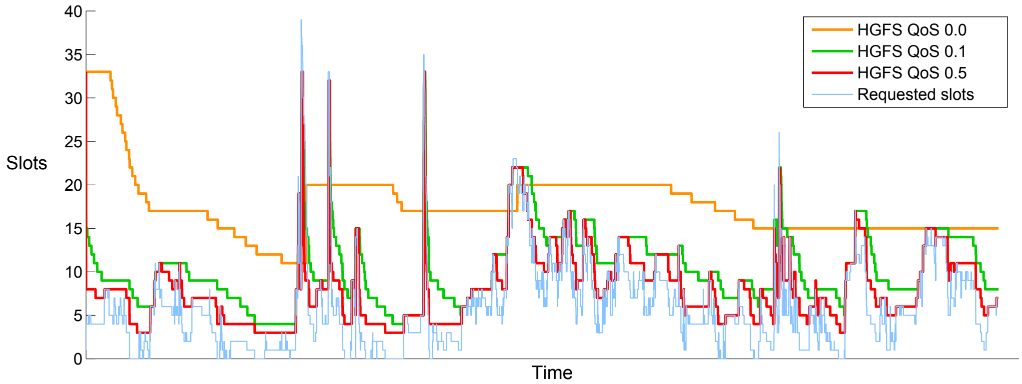


Fig. 9. Cluster simulation trace obtained in the experiment for the test set

The holdout method was used for validation, with a 70-30% split in training and test.

TABLE II. EXPERIMENT RESULTS FOR THE TRAINING SET

	Training set		
	QoS	Energy saved(s)	Reconfigurations
HGFS QoS 0.0	0.00	8.84E+08	75
HGFS QoS 0.1	0.09	1.13E+09	627
HGFS QoS 0.5	0.48	1.19E+09	929

As shown in Tables II and III, the HGFS used as the decision-making mechanism in the EECluster tool can produce very different behaviours depending on the preferences of the cluster administrator, from having no impact on the QoS to a controlled increase on the jobs waiting times and achieving extraordinary energy savings. In other words, the higher impact on the QoS is allowed, the higher energy savings are reached. This is graphically represented in Figure 9, which shows the evolution over time of the aggregated requested slots by the jobs and the slots powered on by each configuration.

TABLE III. EXPERIMENT RESULTS FOR THE TEST SET

	Test set		
	QoS	Energy saved(s)	Reconfigurations
HGFS QoS 0.0	0.00	2.41E+08	42
HGFS QoS 0.1	0.07	3.54E+08	361
HGFS QoS 0.5	0.19	3.90E+08	590

VI. CONCLUDING REMARKS

The software tool EECluster has been presented. This tool transforms OGE/SGE and PBS/TORQUE-based HPC clusters into energy-efficient adaptive resource clusters. A fuzzy rule-based system is used as the decision-making mechanism. Machine learning techniques are implemented that are based on the Hybrid Genetic Fuzzy System introduced in ref. [34]. The learned rule base automatically manages the cluster as a function of the workload scenario and the administrator preferences.

In contrast with the different alternatives in the literature, where the working parameters have to be tuned by hand, EECluster can learn the optimal setup from logged data by means of a multiobjective evolutionary algorithm, achieving very significant energy savings without measurable impact in the quality of service. Details about the practical setup were given, including some use cases and a numerical assessment.

The EECluster tool has been deployed at different clusters located at the Oviedo University, and used for research tasks in algorithm parallelization and chemical computational modelling, and also at the Scientific Modelling Cluster (http://cms.uniovi.es) of the same institution and at academic clusters used for educational purposes.

ACKNOWLEDGEMENTS

This work has been partially supported by “Ministerio de Economía y Competitividad” from Spain/FEDER under grants TIN2011-24302, TIN2014-56967-R and TEC2012-38142-C04-04, and the Regional Ministry of the Principality of Asturias under grant FC-15-GRUPIN14-073.

REFERENCES

[1] F. Yeo, CheeShin and Buyya, Rajkumar and Pourreza, Hossein and Eskicioglu, Rasit and Graham, Peter and Sommers, “Cluster Computing: High-Performance, High-Availability, and High-Throughput Processing on a Network of Computers,” in *Handbook of Nature-Inspired and Innovative Computing*, A. Zomaya, Ed. Springer US, 2006, pp. 521–551.

[2] P. Delforge and J. Whitney, “Issue Paper: Data Center Efficiency Assessment scaling up energy efficiency across the Data Center Industry: evaluating Key Drivers and Barriers,” Natural Resources Defense Council (NRDC), Tech. Rep., 2014. [Online]. Available: http://www.nrdc.org/energy/files/data-center-efficiency-assessment-IP.pdf

[3] M. Ebbers, Mike Archibald, C. F. F. da Fonseca, M. Griffel, V. Para, and M. Searcy, “Smarter Data Centers: Achieving Greater Efficiency,” IBM Redpaper, Tech. Rep., 2011.

[4] The Economist Intelligence Unit, “IT and the environment A new item on the CIOs agenda?” The Economist, Tech. Rep., 2007.

[5] Gartner, “Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions,” STAMFORD, 2007.

[6] C.-H. Hsu and U. Kremer, “The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction,” *ACM SIGPLAN Notices*, vol. 38, no. 5, p. 38, May 2003.

[7] C.-H. Hsu and W.-c. Feng, “A Power-Aware Run-Time System for High-Performance Computing,” in *ACM/IEEE SC 2005 Conference (SC’05)*. IEEE, 2005, pp. 1–1.

[8] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. L. Rountree, and M. E. Femal, "Analyzing the Energy-Time Trade-Off in High-Performance Computing Applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 6, pp. 835–848, Jun. 2007.

[9] M. Lim, V. Freeh, and D. Lowenthal, "Adaptive, Transparent Frequency and Voltage Scaling of Communication Phases in MPI Programs," in *ACM/IEEE SC 2006 Conference (SC'06)*. IEEE, Nov. 2006, pp. 14–14.

[10] Y. Cheng and Y. Zeng, "Automatic Energy Status Controlling with Dynamic Voltage Scaling in Power-Aware High Performance Computing Cluster," in *2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies*. IEEE, Oct. 2011, pp. 412–416.

[11] R. Ge, X. Feng, W.-c. Feng, and K. W. Cameron, "CPU MISER: A Performance-Directed, Run-Time System for Power-Aware Clusters," in *2007 International Conference on Parallel Processing (ICPP 2007)*. IEEE, Sep. 2007, pp. 18–18.

[12] S. Huang and W. Feng, "Energy-Efficient Cluster Computing via Accurate Workload Characterization," in *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*. IEEE, 2009, pp. 68–75.

[13] G. L. T. Chetsa, L. Lefrvre, J.-M. Pierson, P. Stolf, and G. Da Costa, "A Runtime Framework for Energy Efficient HPC Systems without a Priori Knowledge of Applications," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*. IEEE, Dec. 2012, pp. 660–667.

[14] Z. Zong, X. Ruan, A. Manzanares, K. Bellam, and X. Qin, "Improving Energy-Efficiency of Computational Grids via Scheduling," in *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing*, N. Antonopoulos, G. Exarchakos, M. Li, and A. Liotta, Eds. IGI Global, Jan. 2010, ch. 22.

[15] Z. Zong, M. Nijim, A. Manzanares, and X. Qin, "Energy efficient scheduling for parallel applications on mobile clusters," *Cluster Computing*, vol. 11, no. 1, pp. 91–113, Nov. 2007.

[16] C. Bash and G. Forman, "Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center." USENIX Association, Jun. 2007, p. 29.

[17] G. Tang, Q. and Gupta, S. K S and Varsamopoulos, "Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1458–1472, Nov. 2008.

[18] P. Alonso, R. M. Badia, J. Labarta, M. Barreda, M. F. Dolz, R. Mayo, E. S. Quintana-Orti, and R. Reyes, "Tools for Power-Energy Modelling and Analysis of Parallel Scientific Applications," in *2012 41st International Conference on Parallel Processing*. IEEE, Sep. 2012, pp. 420–429.

[19] S. Schubert, D. Kostic, W. Zwaenepoel, and K. G. Shin, "Profiling Software for Energy Consumption," in *2012 IEEE International Conference on Green Computing and Communications*. IEEE, Nov. 2012, pp. 515–522.

[20] V. W. Freeh and D. K. Lowenthal, "Using multiple energy gears in MPI programs on a power-scalable cluster," in *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming - PPoPP '05*. New York, USA: ACM Press, Jun. 2005, p. 164.

[21] D. Li, D. S. Nikolopoulos, K. Cameron, B. R. de Supinski, and M. Schulz, "Power-aware MPI task aggregation prediction for high-end computing systems," in *2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*. IEEE, 2010, pp. 1–12.

[22] C. Xian, Y.-H. Lu, and Z. Li, "A programming environment with runtime energy characterization for energy-aware applications," in *Proceedings of the 2007 international symposium on Low power electronics and design - ISLPED '07*. New York, USA: ACM Press, Aug. 2007, pp. 141–146.

[23] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Load balancing and unbalancing for power and performance in cluster-based systems," in *Workshop on compilers and operating systems for low power*, vol. 180. Barcelona, Spain, 2001, pp. 182–195.

[24] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine, and H. Chan, "Autonomic multi-agent management of power and performance in data centers," pp. 107–114, May 2008.

[25] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," pp. 179–197, Feb. 2002.

[26] J. L. Berral, I. n. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data centers using machine learning," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking - e-Energy '10*. New York, USA: ACM Press, Apr. 2010, p. 215.

[27] W. Lang, J. M. Patel, and J. F. Naughton, "On energy management, load balancing and replication," *ACM SIGMOD Record*, vol. 38, no. 4, p. 35, Jun. 2010.

[28] F. Alvarruiz, C. de Alfonso, M. Caballer, and V. Hernández, "An Energy Manager for High Performance Computer Clusters," in *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*. IEEE, Jul. 2012, pp. 231–238.

[29] M. F. Dolz, J. C. Fernández, S. Iserte, R. Mayo, E. S. Quintana-Ortí, M. E. Cotallo, and G. Díaz, "EnergySaving Cluster experience in CETA-CIEMAT," in *5th Iberian GRID Infrastructure conference*, Santander, 2011.

[30] S. Kiertscher, J. Zinke, S. Gasterstadt, and B. Schnor, "Cherub: Power consumption aware cluster resource management," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, Dec 2010, pp. 325–331.

[31] A. Cocaña Fernández, J. Ranilla, and L. Sánchez, "Energy-Efficient Allocation of Computing Node Slots in HPC Clusters through Evolutionary Multi-Criteria Decision Making," in *Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2014*, 2014, pp. 318–330.

[32] H. Ishibuchi, T. Nakashima, and M. Nii, "Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing)," Nov. 2004.

[33] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[34] A. Cocaña Fernández, J. Ranilla, and L. Sánchez, "Energy-Efficient Allocation of Computing Node Slots in HPC Clusters through Parameter Learning and Hybrid Genetic Fuzzy System Modelling," *The Journal of Supercomputing*, 2014.

[35] "MOEA Framework, a Java library for multiobjective evolutionary algorithms."

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

Título

Energy-Conscious Fuzzy Rule-based Classifiers for Battery Operated Embedded Devices

Autores

Alberto Cocaña-Fernández, José Ranilla, Roberto Gil-Pita and Luciano Sánchez

Journal

Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), *Aceptado, 2017*

Ranking

Rank CORE 2017: A

Rank CORE 2014: A

Rank CORE 2013: A

Rank ERA 2010: A

Rank CORE 2008: A

Categorías:

Artificial Intelligence and Image Processing

Carta de aceptación

Dear Author(s),

Congratulations! On behalf of the FUZZ-IEEE 2017 Technical Program Committee and Technical Chairs, we are pleased to inform you that your paper:

> *Paper ID: 157*
>
> *Author(s): Alberto Cocana, Jose Ranilla, Roberto Gil-Pita and Luciano Sanchez*
>
> *Title: Energy-Conscious Fuzzy Rule-based Classifiers for Battery Operated Embedded Devices*

has been accepted for presentation at the FUZZ-IEEE 2017 and for publication in the conference proceedings published by IEEE. This email provides you with all the information you require to complete your paper and submit it for inclusion in the proceedings.

Sincerely,
Hani Hagras and Francesco Herrera
FUZZ-IEEE 2017 Programme Co-Chairs

Energy-Conscious Fuzzy Rule-based Classifiers for Battery Operated Embedded Devices

Alberto Cocaña-Fernández
Universidad de Oviedo
Computer Science Department
Asturias, Spain
cocanaalberto@gmail.com

José Ranilla
Universidad de Oviedo
Computer Science Department
Asturias, Spain
ranilla@uniovi.es

Roberto Gil-Pita
Universidad de Alcalá
Signal Theory Department
Madrid, Spain
roberto.gil@uah.es

Luciano Sánchez
Universidad de Oviedo
Computer Science Department
Asturias, Spain
luciano@uniovi.es

***Abstract*—A fuzzy rule-based classifier is proposed in this paper where the number of rules in the knowledge base that are fired when an object is classified is anti-monotone with respect to the prior probability of its class. This classifier is intended to secure an equilibrium between accuracy and energy consumption, which is critical in battery operated embedded devices. The method is compared to legacy multi-criteria evolutionary algorithms, where a group of classifiers with different balances between accuracy and consumption are evolved, and the most accurate classifier is selected among those individuals in the Pareto front whose use of the battery does not exceed a given threshold. A significant increase in the battery life is reported without a degradation in the quality of service.**

I. Introduction

Energy use is an issue of interest in data science. The availability of massively parallel hardware has shifted the focus of many machine learning algorithms from raw numerical performance to energy-conscious architectures where the number of computations per kilowatt-hour is preferred to computations per second. In accordance with this, the assessment of these algorithms has to take into account not only the quality of the results but the amount of energy that it takes to learn the decision model, and in certain cases also the amount of energy that the decision model demands when it is operated [1], [2]. Depending on the problem at hand, the critical factor in the energy consumption may be in the learning stage (as happens for instance with big data [3]) or in the decision model. Energy requirements of the decision model cannot be overlooked in battery operated instruments, where efficient models translate into better devices with a higher standby time [4]–[6].

The energetic cost of a decision system is the sum of the costs of obtaining the input patterns and the consumption of the microprocessor implementing the classifier while the algorithm is being executed. Therefore, there are two different paths for optimizing the energy use: (a) performing a feature selection that discards the costly attributes (see [7] for a general approach to cost-based feature selection) and (b) keeping the decision algorithms simple. For example, there are efficient implementations of Support Vector Machines (SVM) [8], [9], where the cut in the number of computations would also reduce the energetic consumption of the decision system. It is also possible that the complexity of the decision algorithm depends on the class of the object; the straightforward case is the unbalanced decision tree (with short paths for frequent classes), but different structures have been studied, including hierarchical and cascade arrangement of classifiers with different computational costs [10]–[13]. Different pruning strategies exist for boosting, bagging or stacking [14] that reduce the computational cost with a small performance impact.

Having said that, none of the works mentioned so far performs a true multiple-criteria analysis of the balance between performance and energy consumption of the decision system. Although there are plenty of studies about multicriteria design of classifiers, concerning feature selection [15], classifier design [16], [17] or ensemble learning [18], these studies have not been applied yet to integrate energy savings into the design of the classifier. This paper is intended to show that it is possible to learn a classifier where the energy consumption decreases with the prior probability of the class of the object being processed, thus an equilibrium between accuracy and energy consumption is secured for either typical or worst-case scenarios. This equilibrium is critical for battery operated embedded devices, as mentioned. The proposed classifier is a grammar-based multi-stage Fuzzy Rule-Based Classifier (FRBC), where multi-objective metaheuristics are deployed to jointly maximize the performance and minimize the expected energy use.

The remainder of the paper is as follows. Section II explains the proposed FRBC. Section III details the learning algorithm. Section IV contains a numerical assessment of the method in a real-world problem on the subject of automatic equalization of hearing aids. The paper concludes in Section V.

II. Fuzzy Rule-Based Classifiers

A hierarchical architecture is adopted where different classifiers are chained, as shown in Figure 1. Each classifier can take a decision or reject the pattern, which in this case falls back to the next classifier. The design procedure seeks that the knowledge bases (KB) of the head classifiers, that are consulted often, are simple. Only the most complex instances will reach the final stage, whose energy requirements are less critical.

Each of the member classifiers is represented as a chain in a context-free grammar, and Multiobjective Evolutionary

Programming is used for obtaining a set of non dominated ensembles of classifiers, as described in the following subsections.

A. FRBCs as chains in a context-free grammar

The classification problem is formally defined as the assignment of a class $\hat{c} \in C$, with $C = \{c_1 \cdots c_K\}$, to an instance $\mathrm{x} = \{x_1 \cdots x_F\}$. This assignment is performed through a FRBC comprising "if assertion then class is c" fuzzy rules. Assertions such as "x is A" are used, where input features are compared to fuzzy linguistic terms defined by either triangular or trapezoidal membership functions. Fuzzy partitions are non-uniform and can comprise multiple shaped fuzzy sets.

A simplified representation is adopted where "OR" logical connectives are allowed, and KBs are valid chains in the context-free grammar defined by the following production rules:

$$
\begin{array}{lcl}
\text{RULE } R_j & \rightarrow & \text{if CONDITION then class is } C_j \\
\text{CONDITION} & \rightarrow & (\text{CONDITION} \wedge \text{CONDITION}) \\
 & & \mid (\text{CONDITION} \vee \text{CONDITION}) \\
 & & \mid \ !\ \text{CONDITION} \\
 & & \mid \text{ASSERTION} \\
\text{ASSERTION} & \rightarrow & x_i \ is \ \widetilde{\mathrm{Lt}}(a,b) \\
 & & \mid \ x_i \ is \ \widetilde{\mathrm{Tr}}(a,b,c) \\
 & & \mid \ x_i \ is \ \widetilde{\mathrm{Rt}}(a,b)
\end{array}
$$

where $\widetilde{\mathrm{Lt}}$, $\widetilde{\mathrm{Rt}}$, $\widetilde{\mathrm{Tr}}$ are, respectively, left trapezoidal, right trapezoidal and triangular fuzzy sets [19]. Observe that a rule-level feature selection is implicit, there are so many rules as classes, and the equivalent number of rules in Disjunctive Normal Form (DNF) representation depends on the location of the "OR" connectives.

Single winner inference [19] is applied, thus the FRBC assigns to each instance the consequent of the most compatible rule. Given an input instance x, the output class $\hat{c}$

$$\hat{c} = \arg\max_j \mu_{R_j}(x) \tag{1}$$

is determined by the recursive application of the following semantic rules:

$$
\mu_{R_j}(x) = \begin{cases}
\mu_{\text{AND}}(x, Ca, Cb) & \text{for } Ca \wedge Cb \\
\mu_{\text{OR}}(x, Ca, Cb) & \text{for } Ca \vee Cb \\
\mu_{\text{NOT}}(x, Ca, Cb) & \text{for } !\, Ca \\
\mu_{\widetilde{Lt}(a,b)}(x) & \text{for LT fuzzy sets} \\
\mu_{\widetilde{Tr}(a,b,c)}(x) & \text{for TR fuzzy sets} \\
\mu_{\widetilde{Rt}(a,b)}(x) & \text{for RT fuzzy sets}
\end{cases} \tag{2}
$$

$$\mu_{\text{AND}}(x, Ca, Cb) = \mu_{Ca}(x) \cdot \mu_{Cb}(x) \tag{3}$$

$$\mu_{\text{OR}}(x, Ca, Cb) = \mu_{Ca}(x) + \mu_{Cb}(x) - \mu_{Ca}(x) \cdot \mu_{Cb}(x) \tag{4}$$

$$\mu_{\text{NOT}}(x, Ca) = 1 - \mu_{Ca}(x) \tag{5}$$

B. Energy-conscious multistage FRBC

The proposed ensemble comprises n FRBCs in decreasing energy efficiency. Since the energy consumption depends on the cost of acquiring the features and also on the computational complexity of the decision system, if all features would cost the same then this scheme would translate into a sequence of increasingly complex classifiers, where the equivalent number of DNF rules that are fired is anti-monotonic with respect to the prior probability of the class. There is an exception to this: if a feature is abnormally expensive (in terms of energy) it may happen that a complex classifier that does not require this costly feature precedes a simple classifier that makes use of the feature. This exception is only meaningful when features are computed on demand: if a decision can be taken without the help of a certain attribute, then the code (or hardware) needed for obtaining this attribute needs not to be activated.

Every classifier but the last implement a reject option that depends on a threshold ϵ on the difference between the activation levels of the winner and second rules, as follows:

$$\text{maxact}(x) = \max_{i=1\ldots K} \mu_{R_i}(x) \tag{6}$$

$$\mathrm{W}(x) = \{i : |\mu_{R_i}(x) - \text{maxact}(x)| < \epsilon\} \tag{7}$$

$$\hat{c}(x) = \begin{cases} \arg\max_i \mu_{R_i}(x) & \#W(x) = 1 \\ \text{reject} & \text{otherwise} \end{cases} \tag{8}$$

In words, if the difference between the activations of the winner rule and the second is higher than the threshold, the instance is assigned the consequent of the winner rule. Otherwise it is passed to the next classifier. The set of discarded classes

$$\mathrm{D}(x) = \{i : |\mu_{R_i}(x) - \text{maxact}(x)| \geq \epsilon\} \tag{9}$$

is also transmitted to the next members of the ensemble, precluding the evaluation of the antecedents of the rules whose consequents have been discarded at a former stage, and therefore saving additional power.

III. Learning Algorithm

There is a wide catalog of techniques for simplifying classifiers, as mentioned in the introduction, and some of these techniques influence the energy consumption of the classification system. Limiting the height of a decision tree, the number of rules of a KB, the nodes of a neural network or the members of an ensemble reduces the computational requirements and ultimately the number of processor cycles needed for classifying an instance. A balance must also exist between the energy use of the processor and that of the hardware that is activated for obtaining the input features, as there are features requiring computations (digital filters, Fourier transforms, etc.) whose cost cannot be neglected.

Pathological cases can be envisaged where complex classifiers that depend on many inexpensive features are preferable to simple classifiers requiring few but costly features, as

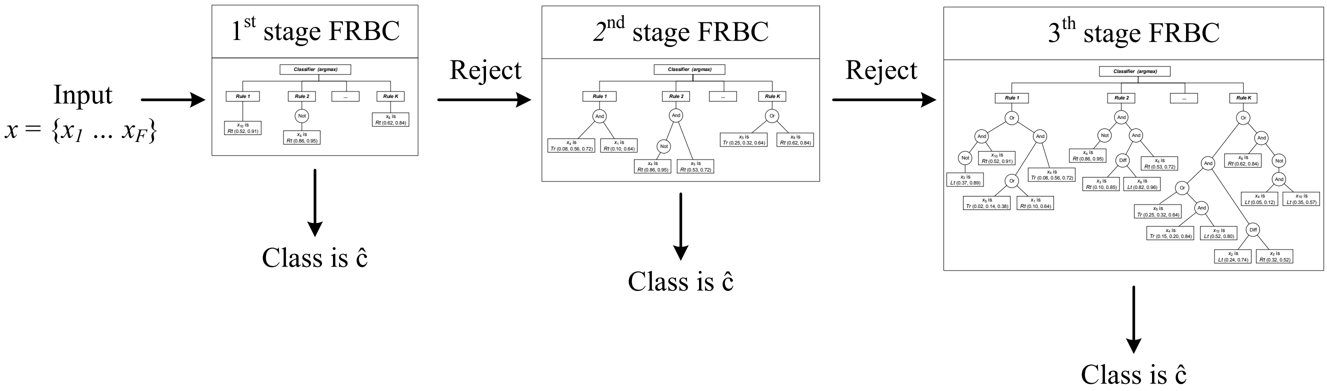


Fig. 1: Representation of a 3-stage FRBC

mentioned. As a consequence of this, classification accuracy and energy consumption must be jointly optimized. Since these objectives conflict, this is a multiple criteria problem for which an isolated solution is not possible but a Pareto-optimal set is sought, that is approximated through multiobjective algorithms [20]–[23]. The purpose of the learning is therefore to produce a well distributed sample of the Pareto-optimal solutions [24].

In this paper, Simulated Annealing (SA) is applied to find a set of non-dominated multi-stage FRBCs. Each individual has a fixed number of stages, and each stage contains K tree-shaped genotypes of variable size and shape. For space limitation reasons, algorithmic details about the multiobjective SA algorithm are not recalled here but can be found in references [25], [26] and [27]. On the contrary, the fitness function is specific of this problem and is explained in the remainder of the section.

The quality of an individual is a list of numbers, estimating the expected classification accuracy and the energy consumption of the classifier. Classification related costs include the count of misclassified instances, but also the classification margins and other indices that are related to the generalization capabilities of the system. On the other hand, the energy consumption depends on the number of floating-point operations needed for obtaining a feature or performing fuzzy inference, and also on the milliwatts used by DSPs or other additional hardware.

The classification is defined by three components (e, m_e, m_c), which are the error rate over the training set, the error margin and the classification margin [28]. e is the resubstitution error [29]. m_e measures how far misclassified instances are from their nearest decision surface (m_e should be minimized). m_c measures the distances between correctly classified instances and their nearest decision surfaces (thus m_c should be maximized). Let the distance between an instance x and its nearest decision surface be

$$d(x) = \mathrm{maxact}(x) - \max_{i=1,\ldots,K}\{\mu_{R_i} : \mathrm{cons}_i \neq \hat{c}(x)\} \qquad (10)$$

where R_i is the degree of truth of the i-th rule in the first stage where x is not rejected. Let cons_i be the consequent of the same rule. Let also G be the set of correctly classified instances, and B the set of misclassified data. Then, m_e and m_c are defined through the following percentiles:

$$m_e = \mathrm{P}_{0.90}\{d(x) : x \in B\} \qquad (11)$$

$$m_c = \mathrm{P}_{0.10}\{d(x) : x \in G\} \qquad (12)$$

that are intended to overcome the influence of the outliers.

Lastly, energy consumption is measured through a second group of coefficients measuring (i) the number of floating-point operations needed for classifying the elements of the training set, (ii) the energy used for computing the features that were demanded for each instance. Each operation and each feature are assigned its own energy cost.

IV. Practical Application and Numerical Assessment

The algorithm in this paper was motivated by a practical problem on the subject of hearing aids. Users of these devices require different adjusts depending on the ambient sound, as the audio quality needs not to be the same for speech, music or noise. Better quality demands a higher consumption. In addition to this, smaller devices are more comfortable but smaller batteries impose stricter requirements.

Modern devices do not require that the user manually adjusts the settings, thus ambient sound is constantly monitored. A classifier is executed each time a sound is detected, and the settings are automatically altered according to the output of this classifier. The power consumption of this monitor is a major restriction in the design of the hearing aid: in this section it will be shown that the fuzzy energy-conscious classifier here proposed gets a longer lifespan that the alternatives.

The experimentation setup consists in 1776 seconds of speech, music and noise tracks featuring a selection of different sound conditions. Audio files are sampled at 16 kHz, with 16 bits per sample. Features are temporal statistics applied to the successive values of different Mel Frequency Cepstral Coefficients (MFCC). A new instance is fed to the system each 16 milliseconds. The battery capacity is 145 milliamp-hour (mAh) and the current drain per floating-point operation (FLOP) of 2×10^{-5} milliamps (mA).

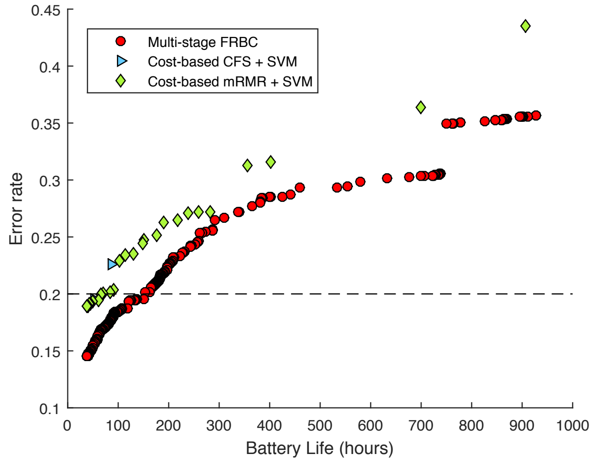
Classifiers learning was done through stratified holdout by splitting the dataset in a training and validation datasets. The first was used to find a Pareto Efficient Frontier of optimal tradeoffs between energy efficiency (in terms of battery life) and classification performance (error rate). The former set was then used to assess the generalization capabilities of each individual whenever is exposed to new sound instances, thus allowing us to filter out overfit classifiers. Three different approaches were evaluated in the experiments: the multistage FRBC described in Section II (labelled "Multi-stage FRBC") and the cost-based filter and ranker feature selection methods proposed in Reference 7 (labelled "Cost-based CFS" and "Cost-based mRMR", respectively). The last two were combined with a SVM classifier and evaluated over a set of λ values ranging between 0 and 10 with 10^{-2} increments.

Once the optimal set of tradeoffs between battery duration and accuracy have been found, then a human expert must choose a single solution according to his/her set of subjective preferences. To allow an objective comparison in the experiments, these preferences were modelled after a simple rule: the chosen classifier is the one with a highest battery life and a classification error rate below or equal to 20%. Figure 2 depicts the Pareto Efficient Frontiers obtained in the experiments, and also highlights the 20% error rate criteria.

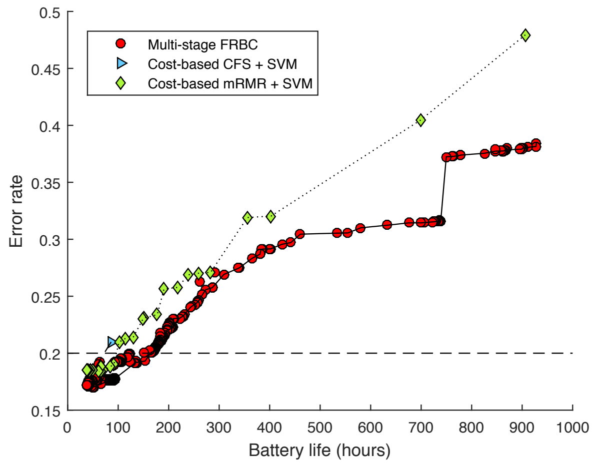
Testing the chosen solutions was done by running a simulation of the hearing aid in four different scenarios: working day, outdoor recreation, sport and a concert day. The simulations consists essentially of a pseudorandom selection of sound instances whose type and probability distributions depend on the given scenario and the time of the day within the simulation. Note that the pseudorandom generation was used to allow all approaches to be tested over the same instances. Scenarios are, therefore, characterized by a series of delimited time frames, each with a certain likelihood for each sound type. These scenarios are detailed in tables I, II, III and IV. In addition to this, Table V shows the distribution of sound classes over the test scenarios.

Speech used in all scenarios include a variety of languages and is presented with and without a noise or music background. Music include an evenly distributed variety of both vocal and instrumental audio tracks. Noise tracks are noticeable different depending on the time period. For example, the noise used for time periods such as "wake-up routine', "spare time", "dinner" or "home time" are based on home-related noise recordings. The noise in "transportation" is based on car, bus, train, train station and traffic-related audio tracks. The "work" periods include environments such as school, office, industrial, commercial, etc. "Coffee break", "lunch break" and "dinner at restaurant" are based on restaurant and cafeteria-related noises.

Simulation traces for each scenario are displayed in Figure 3 and numerical results are summarised in Table VI. First of all, the Pareto Efficient Frontiers displayed in Figure 2 show how the proposed multi-stage FRBC overperforms the other two approaches in terms of accuracy/battery life tradeoffs. This dominance can be appreciated as the set of non-dominated



(a) Training dataset



(b) Validation dataset

Fig. 2: Battery life/error charts displaying the fitness values of each solution in the Pareto Efficient Frontier obtained for each classifier. The dotted line highlights the 20% maximum error rate criteria to select the individual classifier used for testing.

Scenario 1: working day				Sampling probabilities		
N	Begin	End	Description	Speech	Music	Noise
1	0:00	6:30	Device is turned off	-	-	-
2	6:30	7:30	Wake-up routine	30%	-	70%
3	7:30	9:00	Transportation	10%	-	90%
4	9:00	11:30	Work	20%	-	80%
5	11:30	12:00	Coffee break	50%	-	50%
6	12:00	14:00	Work	20%	-	80%
7	14:00	15:00	Lunch break	50%	-	50%
8	15:00	18:00	Work	20%	-	80%
9	18:00	19:30	Transportation	10%	-	90%
10	19:30	22:00	Spare time	20%	70%	10%
11	22:00	23:00	Dinner	50%	-	50%
12	23:00	0:00	Device is turned off	-	-	-

TABLE I: Periods and sampling probabilities for each sound type in Scenario 1 (working day).

Scenario 2: outdoor recreation				Sampling probabilities		
N	Begin	End	Description	Speech	Music	Noise
1	0:00	6:30	Device is turned off	-	-	-
2	6:30	7:30	Wake-up routine	30%	-	70%
3	7:30	9:00	Transportation	10%	-	90%
4	9:00	18:00	Outdoor time	10%	-	90%
5	21:00	22:00	Transportation	10%	-	90%
6	22:00	23:00	Dinner	50%	-	50%
7	23:00	0:00	Device is turned off	-	-	-

TABLE II: Periods and sampling probabilities for each sound type in Scenario 2 (outdoor recreation).

Scenario 3: sport				Sampling probabilities		
N	Begin	End	Description	Speech	Music	Noise
1	0:00	9:00	Device is turned off	-	-	-
2	9:00	10:00	Wake-up routine	30%	-	70%
3	10:00	11:00	Transportation	10%	-	90%
4	11:00	14:00	Sport	10%	-	90%
5	14:00	15:00	Lunch break	50%	-	50%
6	15:00	20:00	Sport	10%	-	90%
7	20:00	21:00	Transportation	10%	-	90%
8	21:00	22:00	Dinner	50%	-	50%
9	22:00	0:00	Device is turned off	-	-	-

TABLE III: Periods and sampling probabilities for each sound type in Scenario 3 (sport).

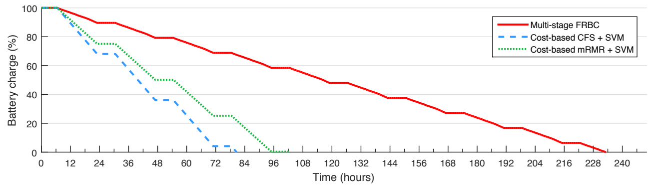
multistage FRBCs are largely below the classifiers learned with the cost-based filter and ranker methods, meaning a lower power consumption to classification performance ratio. Another example of this is that, given the aforementioned 20% error rate criteria, the chosen FRBC selected for testing attains twice as much battery life as the other two alternatives compared with a similar accuracy in the training and validation datasets. Moreover, the multi-stage FRBC approach distinctly achieves a higher coverage and density of the Pareto-optimal set thanks to its inherent flexibility, thus allowing it to comply with a greater diversity of preferences. As for the testing

Scenario 4: concert day				Sampling probabilities		
N	Begin	End	Description	Speech	Music	Noise
1	0:00	10:00	Device is turned off	-	-	-
2	10:00	16:00	Home time	15%	30%	55%
3	16:00	17:00	Transportation	10%	-	90%
4	17:00	22:00	Concert	10%	80%	10%
5	22:00	23:00	Dinner at restaurant	40%	20%	40%
6	23:00	0:00	Transportation	10%	-	90%

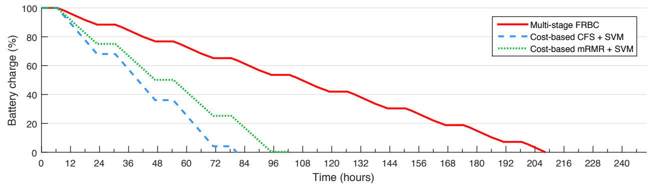
TABLE IV: Periods and sampling probabilities for each sound type in Scenario 4 (concert day).

	Speech	Music	Noise
Scenario 1: working day	20.53%	9.89%	69.58%
Scenario 2: outdoor recreation	13.52%	0.00%	86.48%
Scenario 3: sport	15.42%	0.00%	84.58%
Scenario 4: concert day	14.30%	42.68%	43.02%

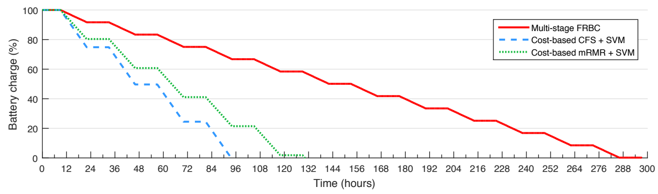
TABLE V: Class distribution over the test dataset in each scenario.



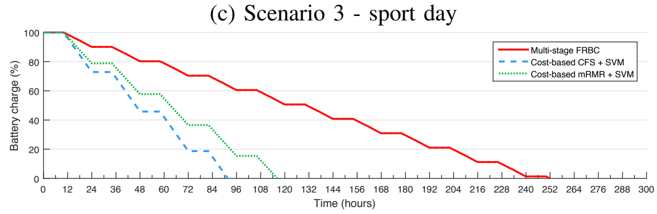
(a) Scenario 1 - working day



(b) Scenario 2 - outdoor recreation day



(c) Scenario 3 - sport day



(d) Scenario 4 - music concert day

Fig. 3: Hearing aid simulation traces for each scenario, depicting the evolution of the battery charge.

scenarios, the proposed approach substantially increases battery life compared to the other two algorithms while strictly complying with the established accuracy goal in all cases. Finally, note that the testing scenarios include periods of time where the hearing aid is switched off, thus producing slightly higher battery life values than the ones in training and validation. Moreover, the current drain in the multi-stage FRBC is instance-dependant, so its energy savings vary accordingly to the types of sound involved in each scenario, as opposed to the other two approaches featuring constant energy costs.

V. Concluding Remarks

A multi-stage FRBC has been proposed that jointly optimizes performance and energy consumption. Energy savings stem from the on-demand computation of the features and the organization of the KB as a pipeline, where most of the classifications are performed at the initial stages, that are kept simple. Only those instances that are positioned beside the decision surfaces reach the latter stages. Since most of the instances are far from these surfaces, the average number of instructions required for making a decision is kept low.

	Multi-stage FRBC	Cost-based CFS	Cost-based mRMR
Number of non-dominated solutions in training	580	2	27
Number of non-dominated solutions in validation	154	2	20
Training error rate	0.196	0.196	0.200
Validation error rate	0.192	0.185	0.185
Battery life in training scenario (hours)	136.084	51.647	66.210
Scenario 1 Battery Life (hours)	233.233	80.633	102.700
Scenario 2 Battery Life (hours)	208.367	80.633	102.700
Scenario 3 Battery Life (hours)	297.283	93.633	130.200
Scenario 4 Battery Life (hours)	251.900	91.633	116.200

TABLE VI: Experiment results in the testing scenarios.

Experimental evidence was provided to support the higher battery life of the new classifier, that more than doubles that of the conventional classifier with the best energy-accuracy balance.

ACKNOWLEDGEMENTS

This work has been partially supported by the MEIC from Spain/FEDER under grants TEC2015-67387-C4-3-R, TEC2015-67387-C4-4-R and TIN2014-56967-R and by the Principality of Asturias under grant FC-15-GRUPIN14-073.

REFERENCES

[1] Z. Xu, M. J. Kusner, K. Q. Weinberger, and M. Chen, "Cost-sensitive tree of classifiers," in 30th International Conference on Machine Learning, ICML 2013, no. PART 1, 2013, pp. 133–141.

[2] L. Li, U. Topkara, B. Coskun, and N. Memon, "CoCoST: A computational cost sensitive classifier," in Proceedings - IEEE International Conference on Data Mining, ICDM, 2009, pp. 268–277.

[3] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient Machine Learning for Big Data: A Review," Big Data Research, vol. 2, no. 3, pp. 87–93, apr 2015.

[4] Ó. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," IEEE Communications Surveys and Tutorials, vol. 15, no. 3, pp. 1192–1209, 2013.

[5] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," Journal of Universal Computer Science, vol. 19, no. 9, pp. 1295–1314, 2013.

[6] A. Betancourt, M. M. Lopez, C. S. Regazzoni, and M. Rauterberg, "A sequential classifier for hand detection in the framework of egocentric vision," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 600–605.

[7] V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A framework for cost-based feature selection," Pattern Recognition, vol. 47, no. 7, pp. 2481–2489, jul 2014.

[8] S. S. Keerthi, O. Chapelle, and D. DeCoste, "Building support vector machines with reduced classifier complexity," Journal of Machine Learning Research, vol. 7, pp. 1493–1515, 2006.

[9] M. Ring and B. M. Eskofier, "An approximation of the Gaussian RBF kernel for efficient classification with SVMs," Pattern Recognition Letters, vol. 84, pp. 1339–1351, 2016.

[10] J. Pujara, H. Daumé III, and L. Getoor, "Using classifier cascades for scalable e-mail classification," in ACM International Conference Proceeding Series, 2011, pp. 55–63.

[11] K. Trapeznikov, V. Saligrama, and D. Castañón, "Multi-Stage Classifier Design," in JMLR W&CP, vol. 25, 2012, pp. 459–474.

[12] Z. E. Xu, M. J. Kusner, K. Q. Weinberger, M. Chen, and O. Chapelle, "Classifier cascades and trees for minimizing feature evaluation cost," Journal of Machine Learning Research, vol. 15, pp. 2113–2144, 2014.

[13] S. Venkataramani, A. Raghunathan, J. Liu, and M. Shoaib, "Scalable-effort classifiers for energy-efficient machine learning," in Proceedings of the 52nd Annual Design Automation Conference on - DAC '15. New York, New York, USA: ACM Press, jun 2015, pp. 1–6.

[14] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, "LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy," Neurocomputing, vol. 123, pp. 424–435, 2014.

[15] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach." IEEE transactions on cybernetics, vol. 43, no. 6, pp. 1656–71, dec 2013.

[16] R. Alcalá, M. J. Gacto, and F. Herrera, "A fast and scalable multi-objective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems," IEEE Transactions on Fuzzy Systems, vol. 19, no. 4, pp. 666–681, 2011.

[17] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," IEEE Transactions on Fuzzy Systems, vol. 21, no. 1, pp. 45–65, 2013.

[18] K. Trawiński, O. Cordón, A. Quirin, and L. Sánchez, "Multiobjective genetic classifier selection for random oracles fuzzy rule-based classifier ensembles: How beneficial is the additional diversity?" Knowledge-Based Systems, vol. 54, pp. 3–21, 2013.

[19] H. Ishibuchi, T. Nakashima, and M. Nii, "Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing)," nov 2004.

[20] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," IEEE Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 117–132, apr 2003.

[21] S. Voß, "Meta-heuristics: The State of the Art," in Local Search for Planning and Scheduling, A. Nareyek, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–23.

[22] A. Bölte and U. W. Thonemann, "Optimizing simulated annealing schedules with genetic programming," European Journal of Operational Research, vol. 92, no. 2, pp. 402–416, 1996.

[23] O. Tekinalp and G. Karsli, "A new multiobjective simulated annealing algorithm," Journal of Global Optimization, vol. 39, no. 1, pp. 49–77, jul 2007.

[24] B. Suman, "Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem," Computers & Chemical Engineering, vol. 28, no. 9, pp. 1849–1871, 2004.

[25] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," Journal of the Operational Research Society, vol. 57, no. 10, pp. 1143–1160, oct 2006.

[26] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA," IEEE Transactions on Evolutionary Computation, vol. 12, no. 3, pp. 269–283, jun 2008.

[27] K. Smith, R. Everson, J. Fieldsend, C. Murphy, and R. Misra, "Dominance-Based Multiobjective Simulated Annealing," IEEE Transactions on Evolutionary Computation, vol. 12, no. 3, pp. 323–342, jun 2008.

[28] K. Trawinski, O. Cordon, L. Sanchez, and A. Quirin, "A Genetic Fuzzy Linguistic Combination Method for Fuzzy Rule-Based Multiclassifiers," IEEE Transactions on Fuzzy Systems, vol. 21, no. 5, pp. 950–965, oct 2013.

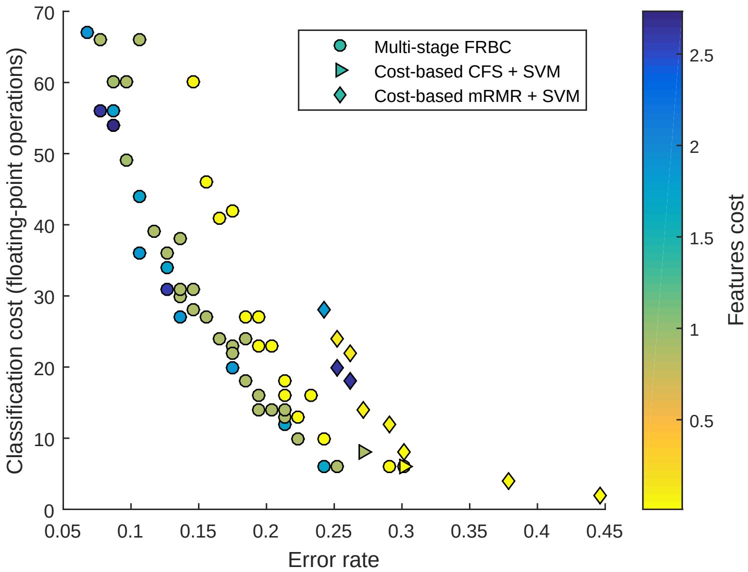
[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," ACM SIGKDD Explorations Newsletter, vol. 11, no. 1, p. 10, nov 2009.

Parte III: Apéndices

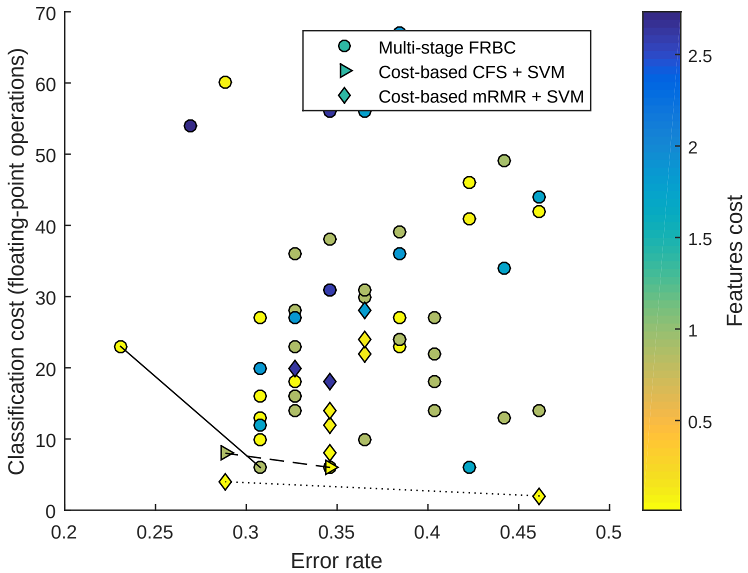
A

Resultados experimentales con conjuntos de datos heterogéneos

2De forma adicional a los resultados obtenidos en la publicación [42] para el problema SEC, se realizaron experimentos con los conjuntos de datos descritos en la Sección 3.6, comparando el clasificador MFRBC con las variantes sensibles al coste del CFS y el mRMR propuestas en [29]. A continuación se muestran los resultados para cada conjunto de datos mediante una representación gráfica de los frentes de Pareto obtenidos con cada clasificador. Por último, se incluye una tabla que recoge el número de soluciones no dominadas para cada clasificador y conjunto de datos.

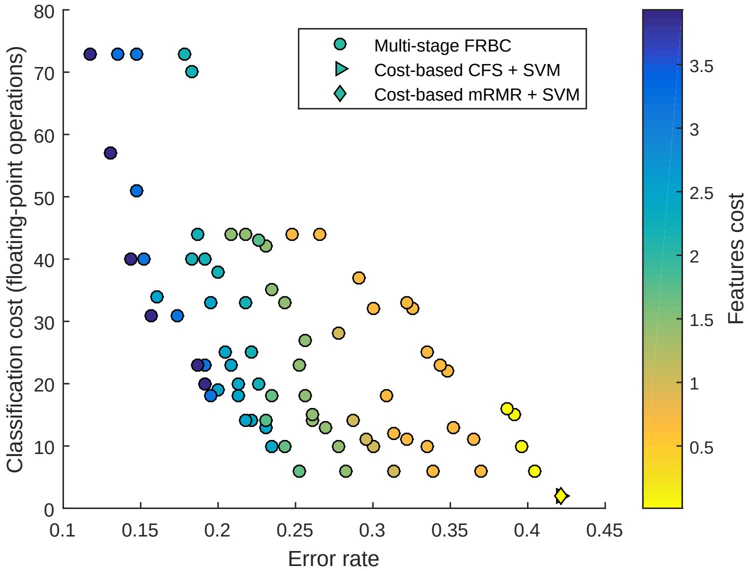


(a) Entrenamiento

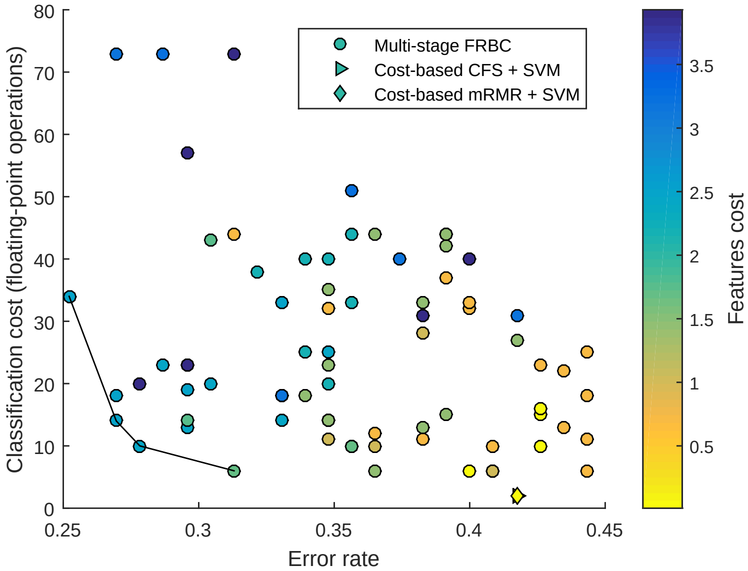


(b) Test

Figura A.1: Frentes de Pareto obtenidos para el conjunto de datos *Hepatitis*.

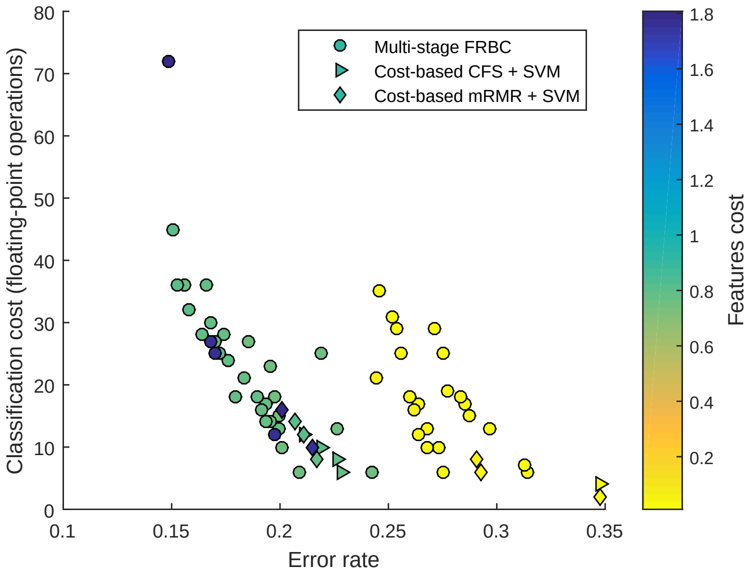


(a) Entrenamiento

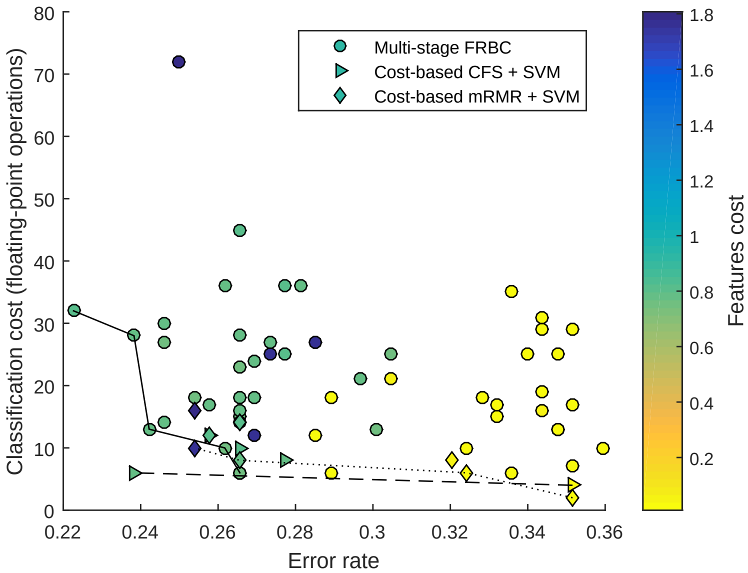


(b) Test

Figura A.2: Frentes de Pareto obtenidos para el conjunto de datos *Liver*.

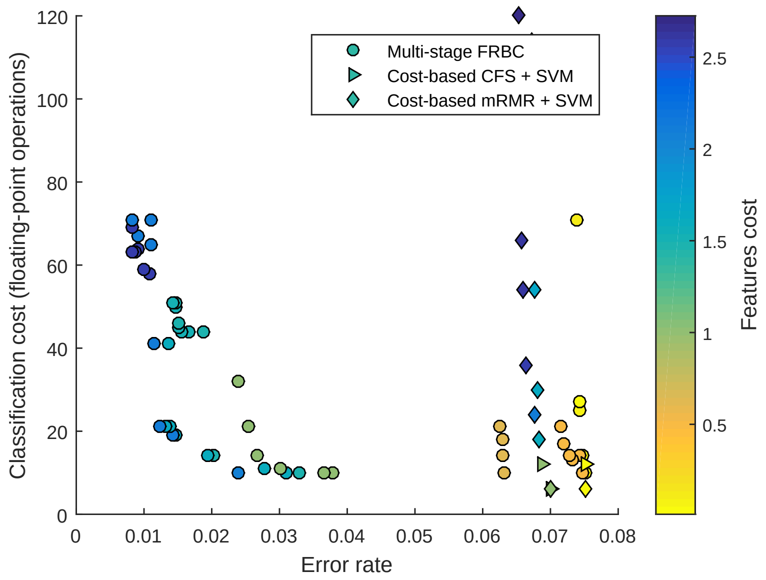


(a) Entrenamiento

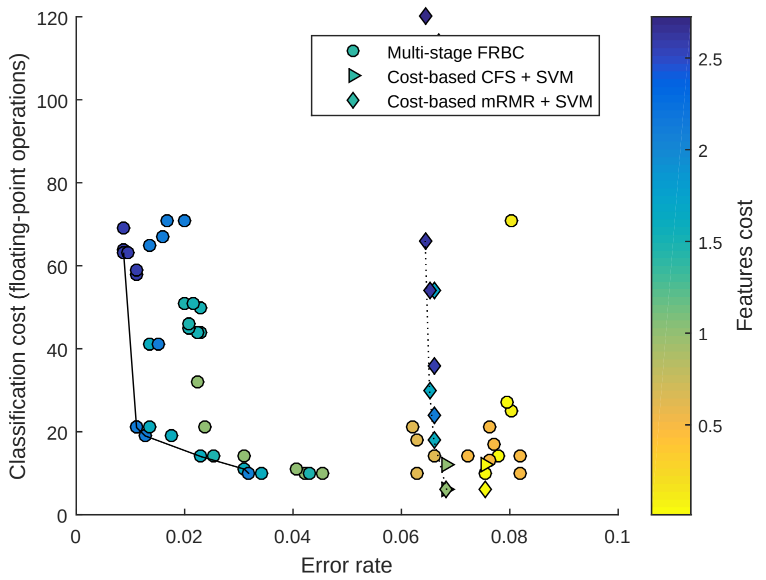


(b) Test

Figura A.3: Frentes de Pareto obtenidos para el conjunto de datos *Pima*.

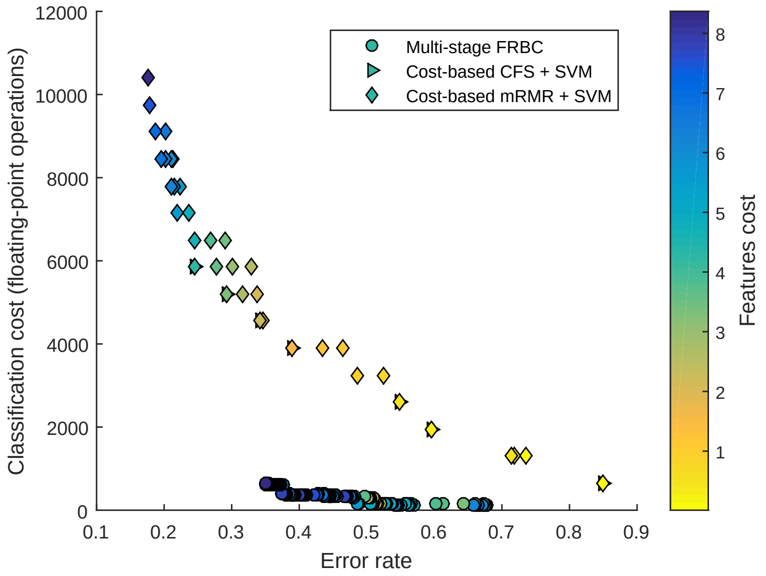


(a) Entrenamiento

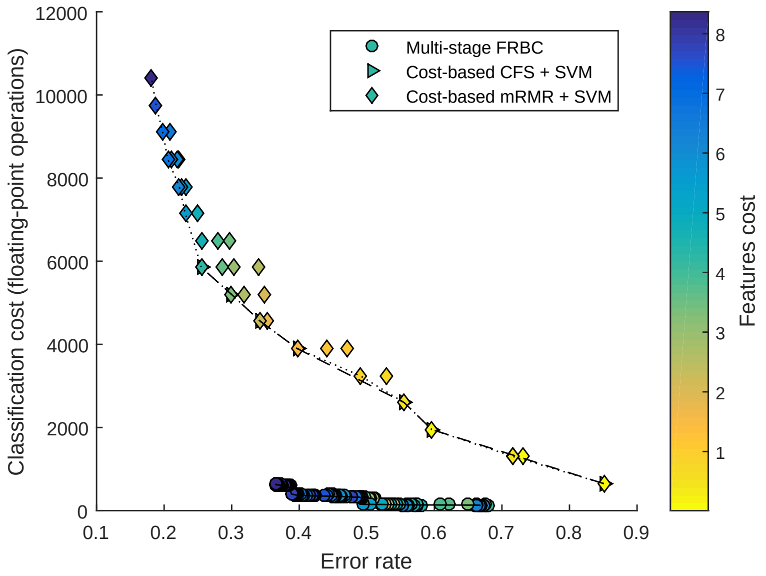


(b) Test

Figura A.4: Frentes de Pareto obtenidos para el conjunto de datos *Thyroid*.

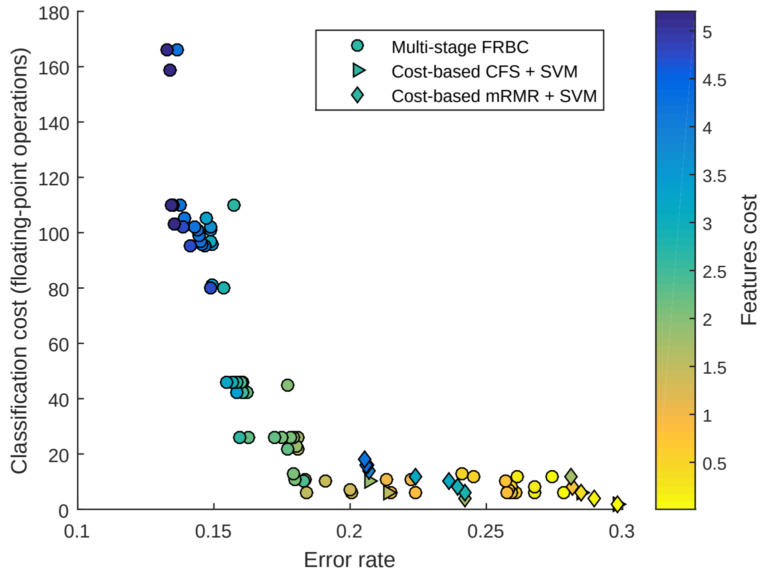


(a) Entrenamiento

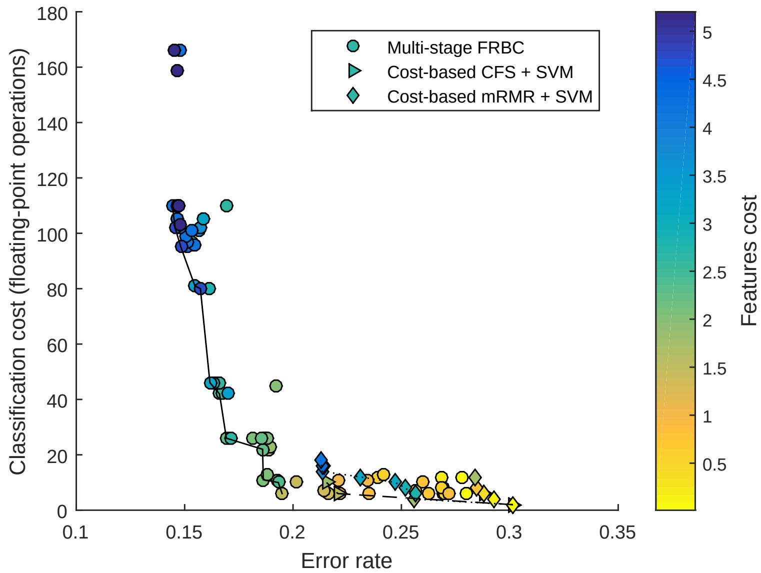


(b) Test

Figura A.5: Frentes de Pareto obtenidos para el conjunto de datos *Letter*.

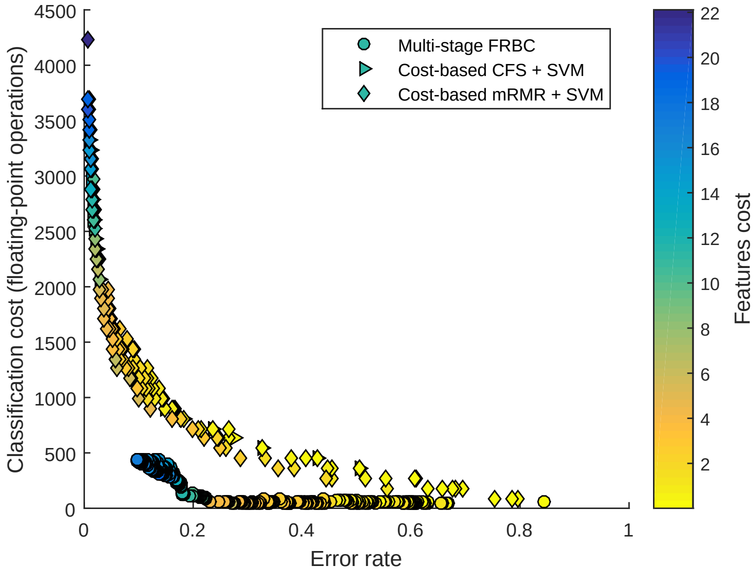


(a) Entrenamiento

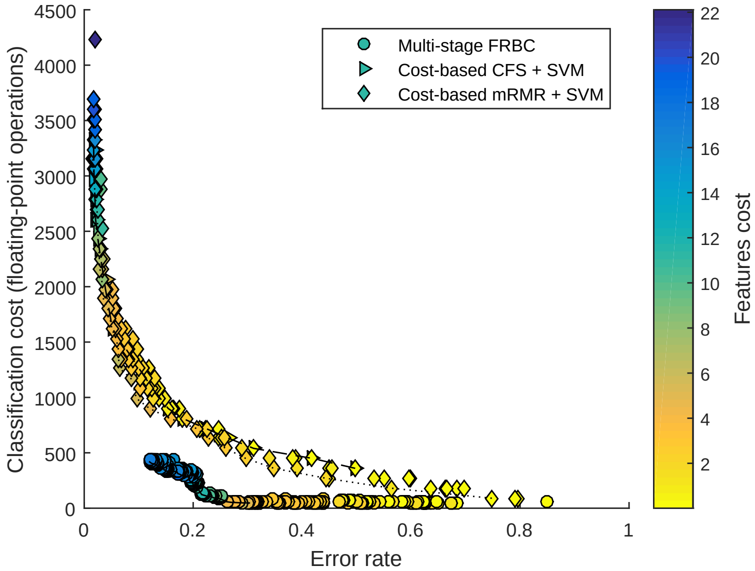


(b) Test

Figura A.6: Frentes de Pareto obtenidos para el conjunto de datos *Magic04*.

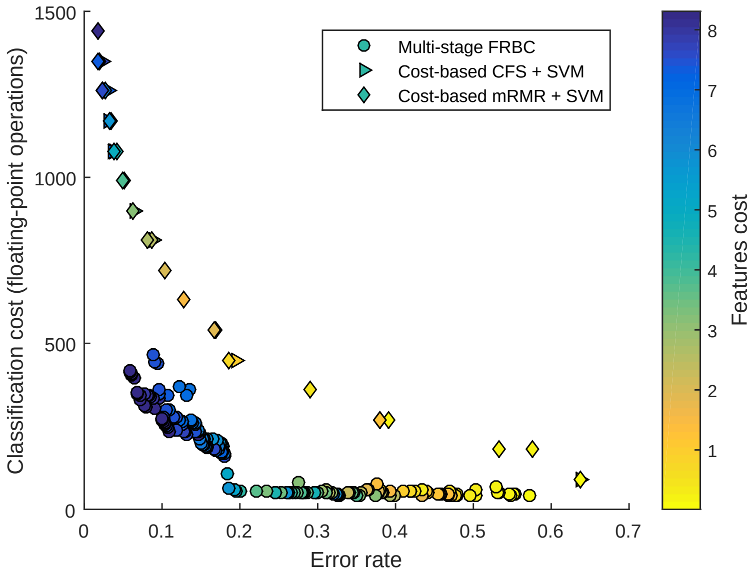


(a) Entrenamiento

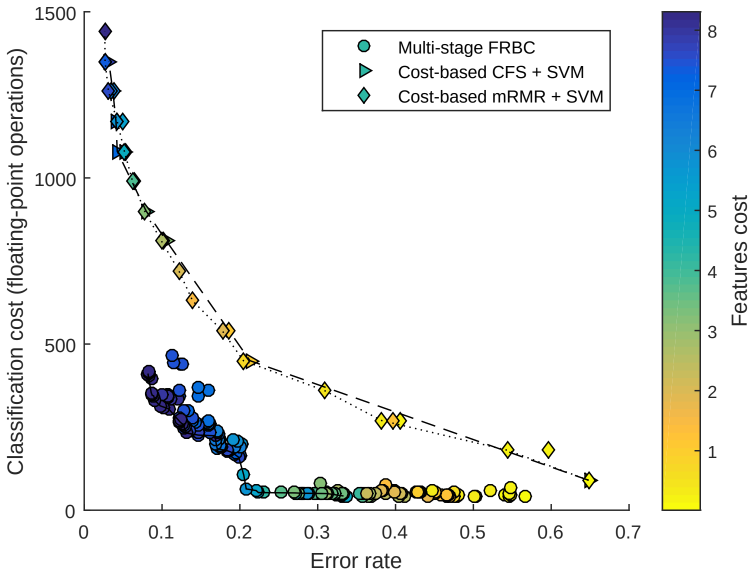


(b) Test

Figura A.7: Frentes de Pareto obtenidos para el conjunto de datos *Optdigits*.

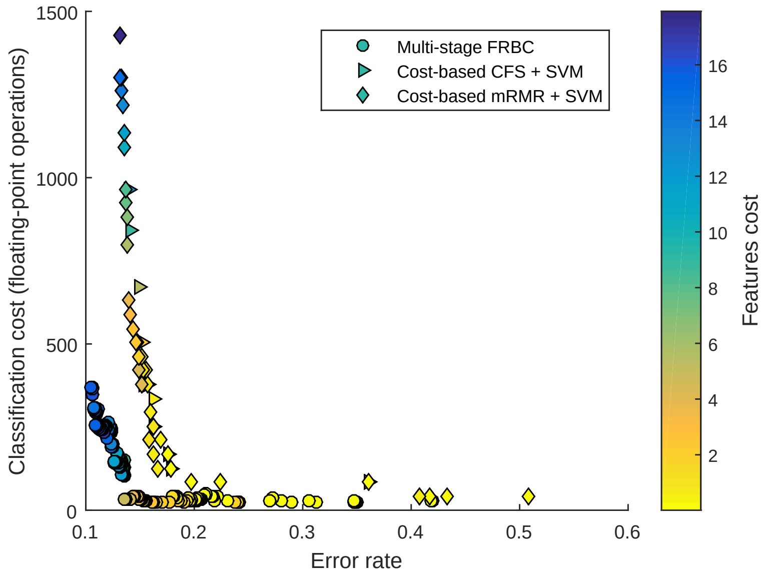


(a) Entrenamiento

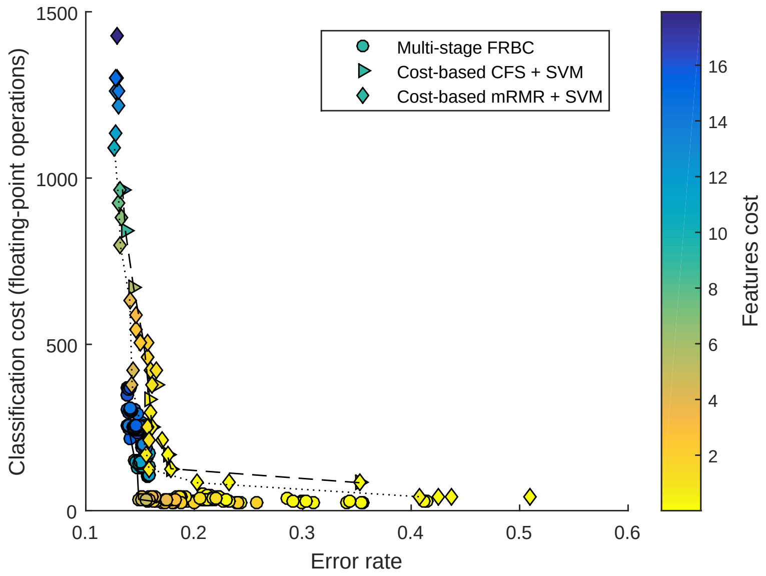


(b) Test

Figura A.8: Frentes de Pareto obtenidos para el conjunto de datos *Pendigits*.

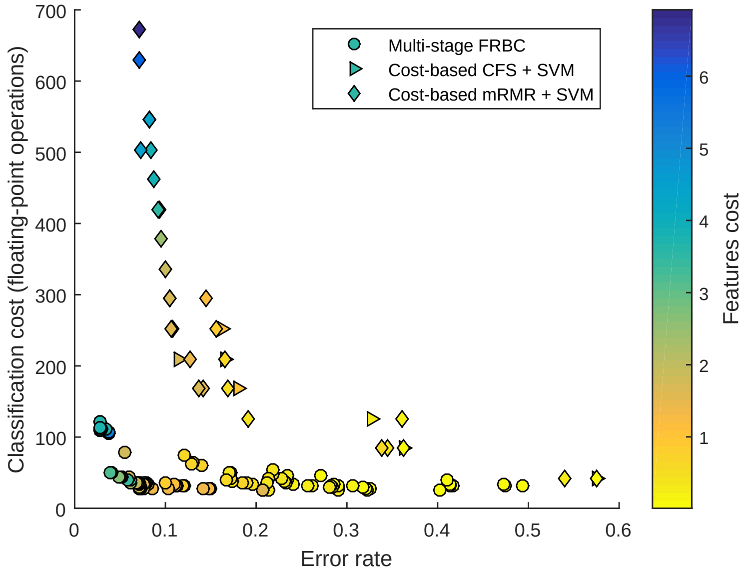


(a) Entrenamiento

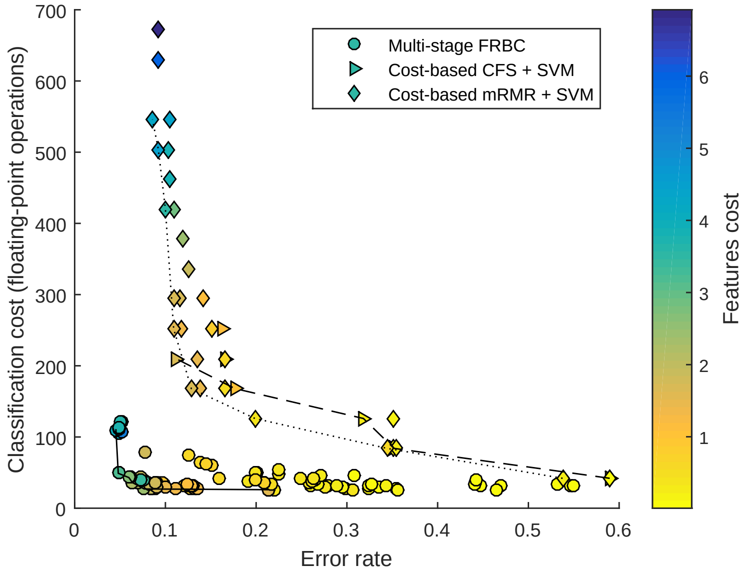


(b) Test

Figura A.9: Frentes de Pareto obtenidos para el conjunto de datos *Sat*.

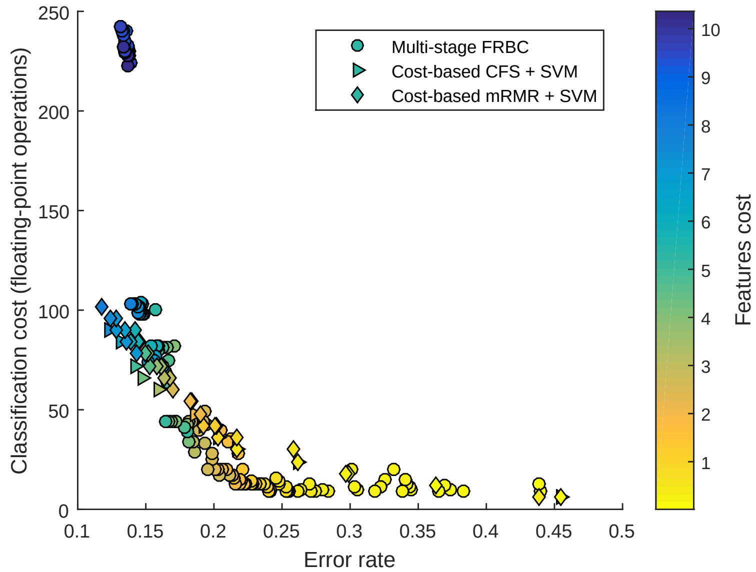


(a) Entrenamiento

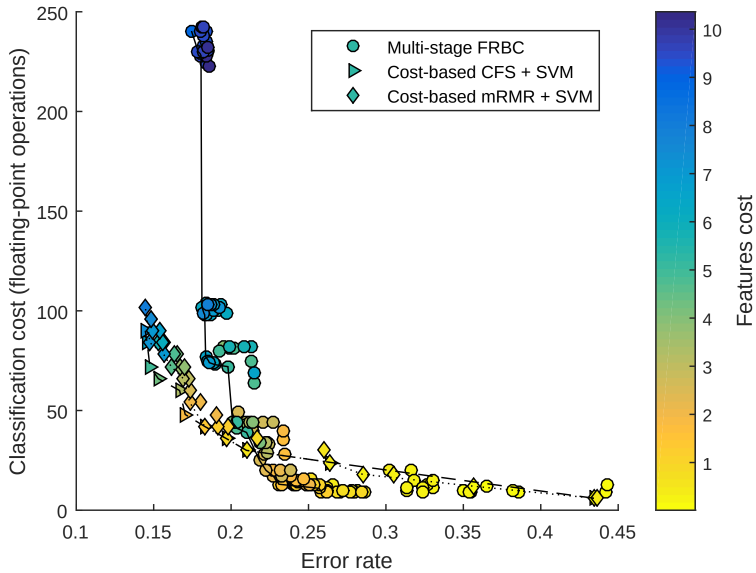


(b) Test

Figura A.10: Frentes de Pareto obtenidos para el conjunto de datos *Segmentation*.

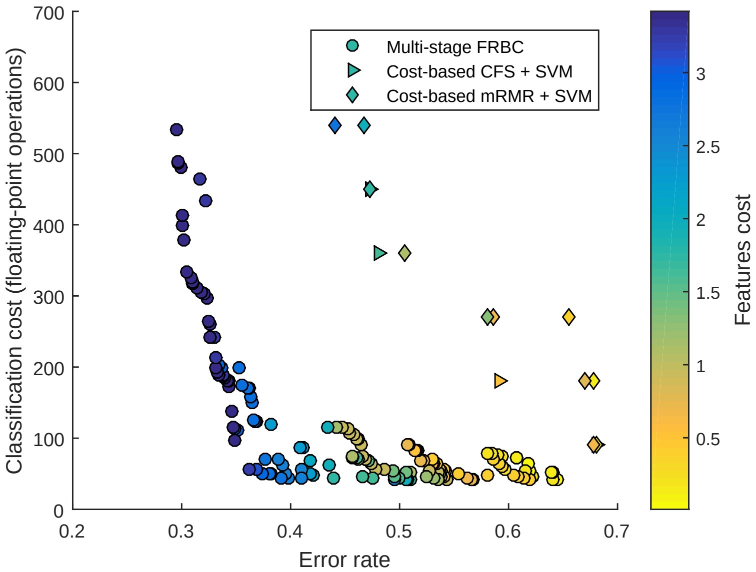


(a) Entrenamiento

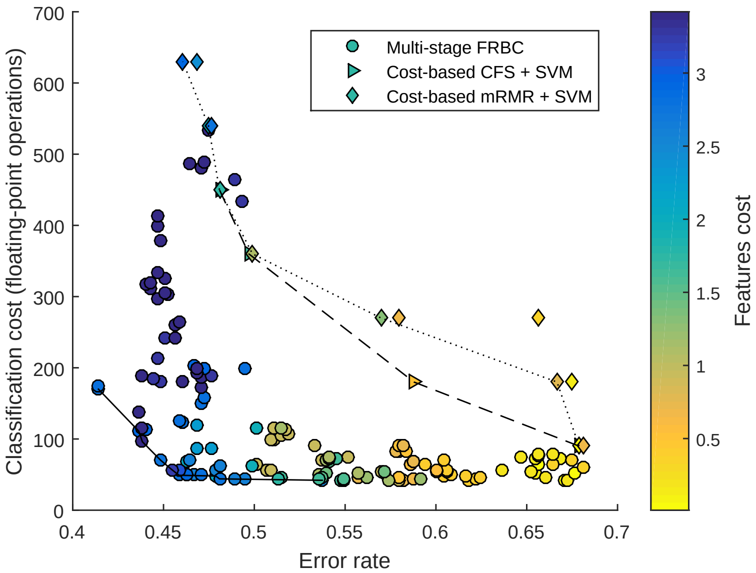


(b) Test

Figura A.11: Frentes de Pareto obtenidos para el conjunto de datos *Waveform*.

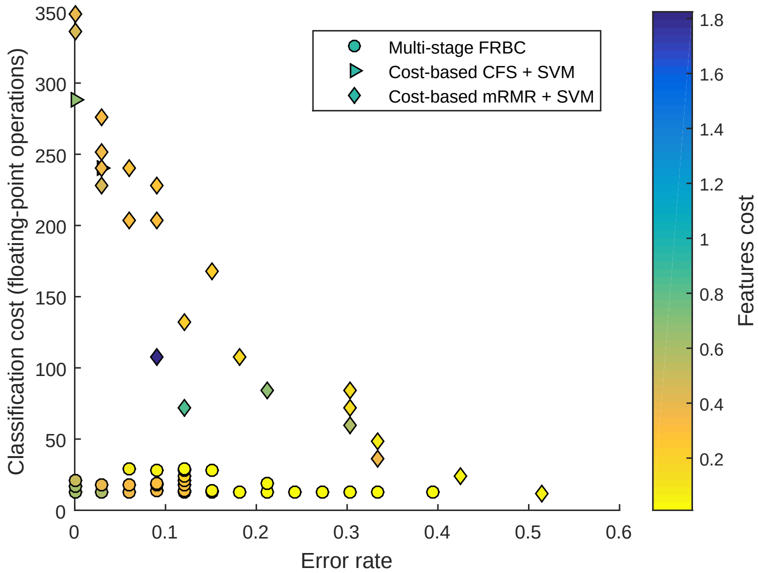


(a) Entrenamiento

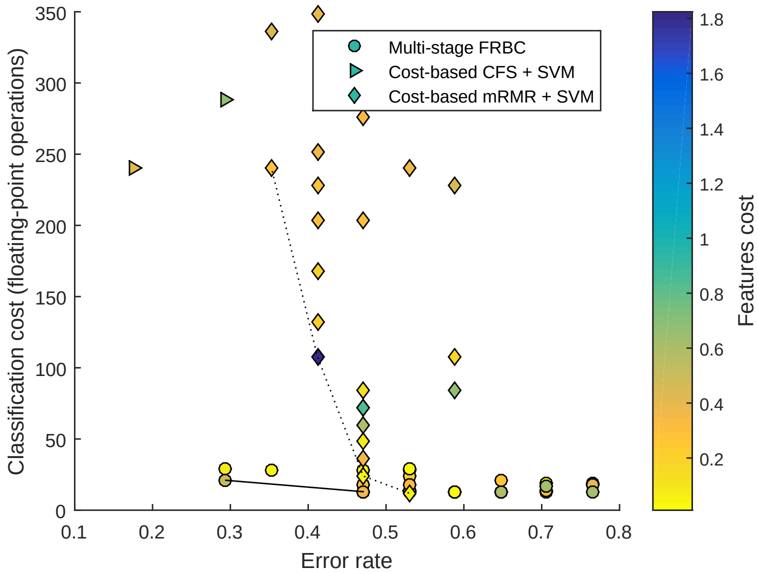


(b) Test

Figura A.12: Frentes de Pareto obtenidos para el conjunto de datos *Yeast*.

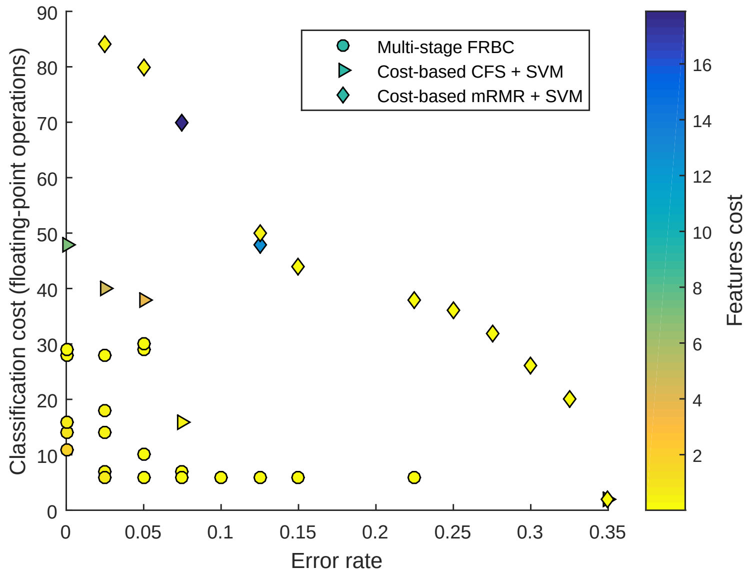


(a) Entrenamiento

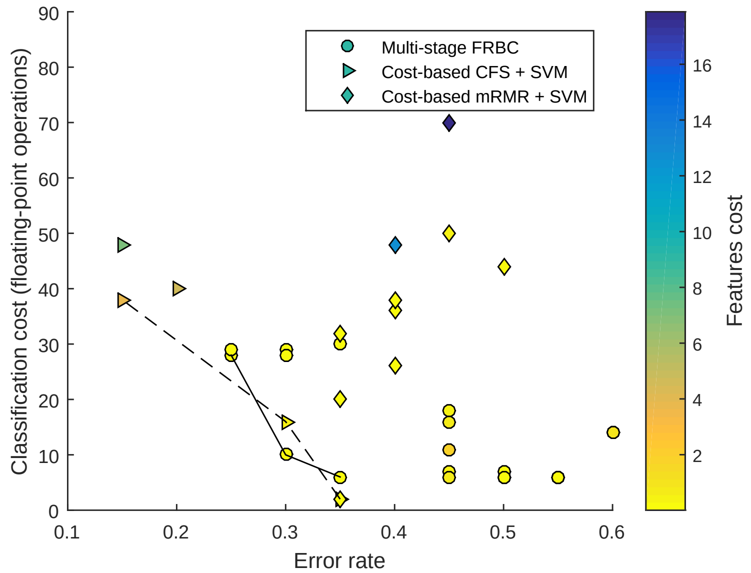


(b) Test

Figura A.13: Frentes de Pareto obtenidos para el conjunto de datos *Glioblastoma*.

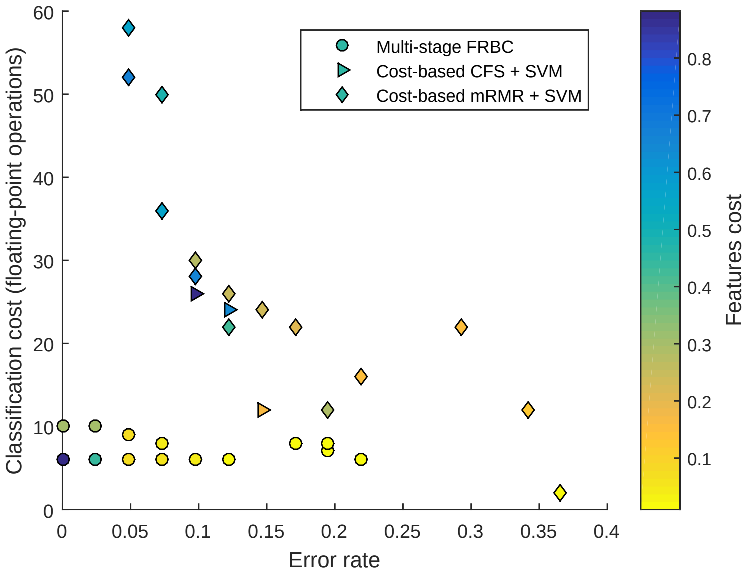


(a) Entrenamiento

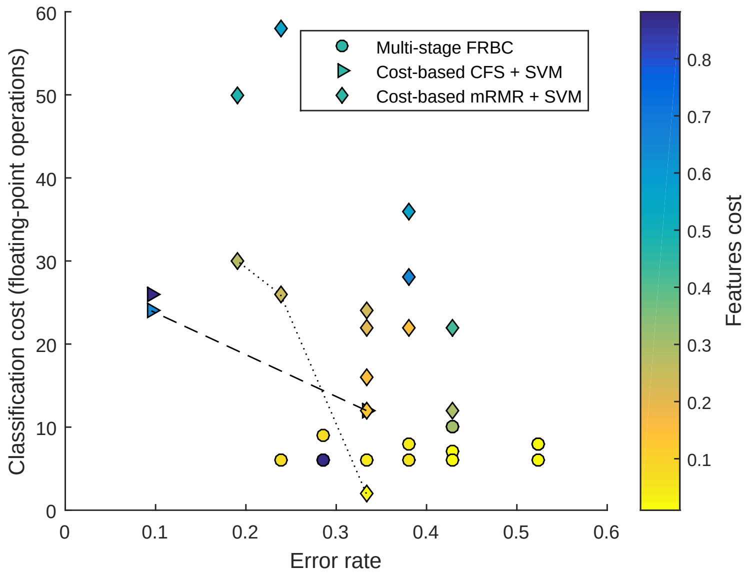


(b) Test

Figura A.14: Frentes de Pareto obtenidos para el conjunto de datos *CNS*.

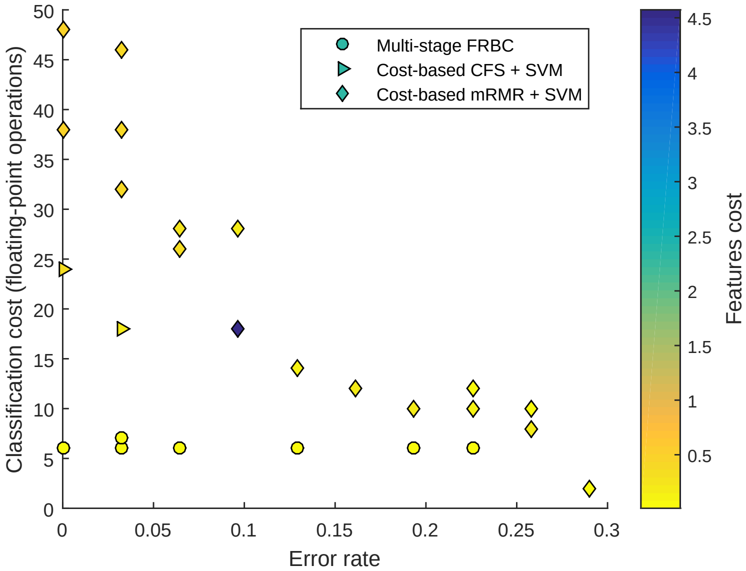


(a) Entrenamiento

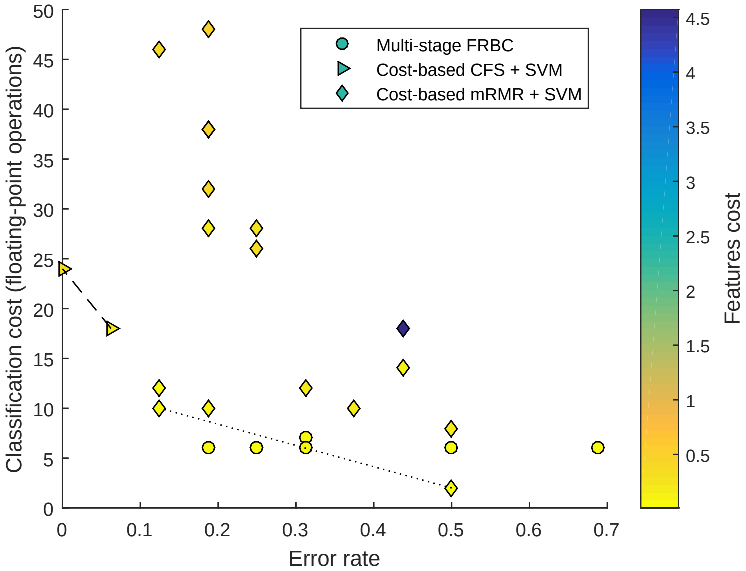


(b) Test

Figura A.15: Frentes de Pareto obtenidos para el conjunto de datos *Colon*.

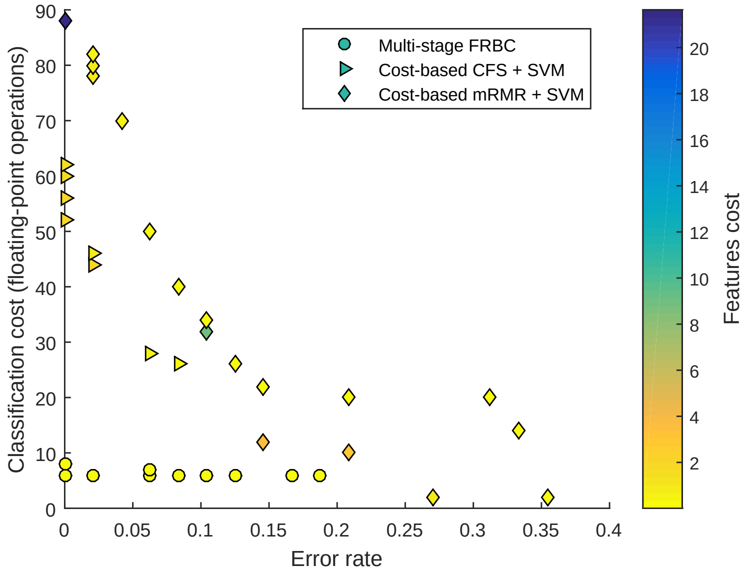


(a) Entrenamiento

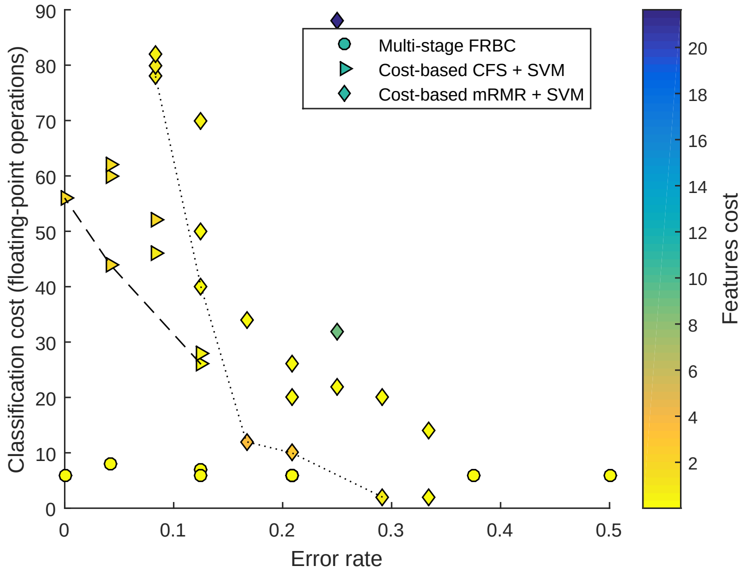


(b) Test

Figura A.16: Frentes de Pareto obtenidos para el conjunto de datos *DLBCL*.



(a) Entrenamiento



(b) Test

Figura A.17: Frentes de Pareto obtenidos para el conjunto de datos *Leukemia*.

Conjunto de datos		Número de soluciones no dominadas Multi-stage FRBC	Cost-based CFS	Cost-based mRMR
Hepatitis	Entrenamiento	**52**	2	10
	Test	**2**	**2**	**2**
Liver	Entrenamiento	**77**	1	1
	Test	**4**	1	1
Pima	Entrenamiento	**54**	5	8
	Test	**6**	2	4
Thyroid	Entrenamiento	**52**	3	11
	Test	**7**	1	4
Letter	Entrenamiento	**175**	7	36
	Test	**40**	7	15
Magic04	Entrenamiento	**71**	4	14
	Test	**12**	3	7
Optdigits	Entrenamiento	**375**	20	145
	Test	**36**	16	29
Pendigits	Entrenamiento	**181**	9	28
	Test	**31**	8	16
Sat	Entrenamiento	**176**	11	40
	Test	**10**	9	9
Segmentation	Entrenamiento	**93**	7	29
	Test	7	5	**8**
Waveform	Entrenamiento	**138**	11	35
	Test	**22**	11	13
Yeast	Entrenamiento	**149**	4	13
	Test	**9**	4	7
Glioblastoma	Entrenamiento	**30**	2	23
	Test	3	1	**4**
CNS	Entrenamiento	**20**	5	12
	Test	**3**	**3**	1
Colon	Entrenamiento	14	3	**15**
	Test	1	2	**3**
DLBCL	Entrenamiento	7	2	**17**
	Test	1	**2**	**2**
Leukemia	Entrenamiento	10	8	**18**
	Test	1	3	**5**

Tabla A.1: Número de soluciones no dominadas para cada conjunto de datos.

B

Representación gráfica del clasificador MFRBC

De forma anexa a la publicación *Multicriteria design of cost-conscious fuzzy rule-based classifiers* [42] se realizó una representación gráfica dos clasificadores MFRBC correspondientes al clasificador más preciso y el de menor coste de los encontrados en los experimentos detallados en dicha publicación. Debido a su tamaño, estas representaciones se omitieron en la versión final de dicha publicación, pero se incluyen a continuación junto con una tabla que resume sus características:

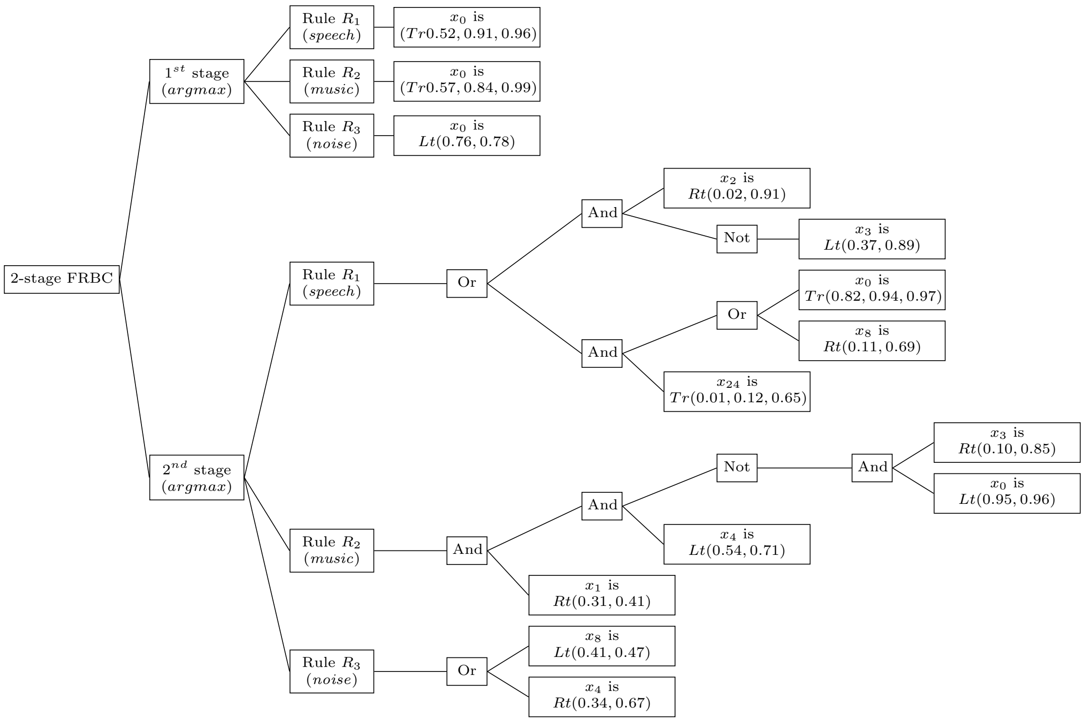


Figura B.1: Clasificador MFRBC de menor coste encontrado en el problema SEC.

Resultados para el conjunto de Test	FRBC más preciso	FRBC de menor coste
Tasa de error	0.170	0.381
Coste de clasificación	146121.87	7830.86
Número de nodos en la Etapa 1	31	6
Número de nodos en la Etapa 2	123	24
Error esperado en le Etapa 1 (e)	0.033	$9{,}7 \times 10^{-5}$
Coste de clasificación de la Etapa 1 (FLOPs)	69000	7812.5
Coste de clasificación de la Etapa 2 (FLOPs)	166375	49125
Tasa de error de la Etapa 1	0.026	0.382
Tasa de error de la Etapa 2	0.208	0.154
Porcentaje de clasificaciones en la Etapa 1	20.80 %	99.96 %
Porcentaje de clasificaciones en la Etapa 2	79.20 %	0.04 %

Tabla B.1: Resumen del clasificador MFRBC más preciso y el de menor coste aprendidos en el problema SEC.

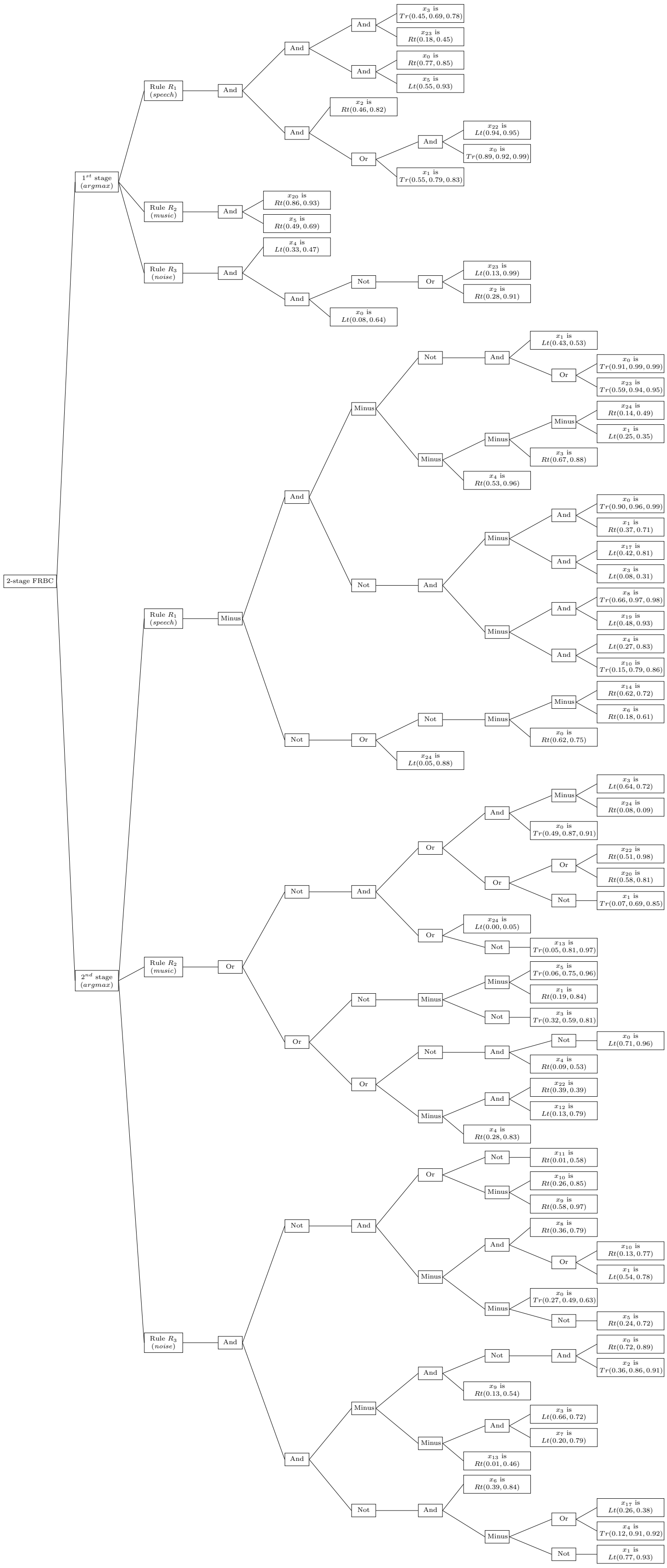


Figura B.2: Clasificador MFRBC con menor tasa de error encontrado en el problema SEC.