UNIVERSIDAD
DE GRANADA

Modelado de series temporales multivariantes y fusión de datos con regresión simbólica: Aplicación a la mejora de la eficiencia energética.

Ramón Rueda Delgado

Para obtener el grado de doctor internacional como parte del

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Directores

MANUEL PEGALAJAR CUÉLLAR
MARÍA DEL CARMEN PEGALAJAR JIMÉNEZ

Granada, septiembre de 2020

Editor: Universidad de Granada. Tesis Doctorales
Autor: Ramón Rueda Delgado
ISBN: 978-84-1306-730-8
URI: http://hdl.handle.net/10481/65407

Esta tesis doctoral ha sido financiada por el proyecto de investigación nacional TIN2015-64776-C3-1-R. El estudiante de doctorado, Ramón Rueda Delgado, ha sido beneficiario de un contrato predoctoral para formación de doctores (FPI) del Ministerio de Economía, Industria y Competitividad del Gobierno de España.

Cuando se nace pobre, estudiar es el mayor acto de rebeldía contra el sistema. El saber rompe las cadenas de la esclavitud.

Tomás Bulat

Agradecimientos

Quiero dar las gracias a mis padres, por todo el sacrificio que han hecho para que haya tenido la oportunidad de estudiar y permitir que cumpla un sueño. Agradecer también a mi hermano, por sus buenos consejos, con los que me ha ayudado a superar cada piedra que me he encontrado en el camino. A mi tío Alfonso, quién sembró en mi la curiosidad en el mundo de la informática.

A los diferentes compañeros y amigos con quiénes he tenido el placer de compartir este camino. En especial, agradecer a Rubén y Juanfra, siempre han estado a mi lado para motivarme y ayudarme en cada momento.

Agradecer también a mis directores, María del Carmen Pegalajar y Manuel Pegalajar, por ofrecerme la posibilidad de realizar esta tesis doctoral y guiarme en todo momento, por su infinita paciencia, comprensión y dedicación durante todo este tiempo.

Por último, dar las gracias a mi compañera de viaje Ana, por su cariño, paciencia, confianza y ayuda.

Gracias a todos.

Acrónimos

ACO Ant Colony Optimization

AI Artificial Intelligence

BAS Building Automation System

GP Genetic Programming

IEA International Energy Agency

IRENA International Renewable Energy Agency

ML Machine Learning

MSE Mean Square Error

SLG Straight Line Grammar

SLP Straight Line Program

SR Symbolic Regression

AIE Agencia Internacional de Energía

AG Algoritmos Genéticos

ECM Error Cuadrático Medio

IA Inteligencia Artificial

PG Programación Genética

RS Regresión Simbólica

Resumen

Alcanzar un consumo eficiente y sostenible en el sector de los edificios se ha convertido en uno de los grandes retos a resolver en esta década. En el tránsito hacia una descarbonización completa en el uso de energía, la eficiencia energética se posiciona como eje central para identificar y evitar consumos innecesarios. Como consecuencia directa, se prevé reducir la huella de carbono y minimizar los riesgos del cambio climático. Gracias al reciente avance en la tecnología de monitorización y del desarrollo de técnicas de Inteligencia Artificial, la comunidad investigadora ha centrado sus esfuerzos en el desarrollo de algoritmos inteligentes para extraer de forma automática conocimiento útil procedente de datos relacionados con el consumo energético, permitiendo identificar los factores más relevantes que ayuden a reducir el consumo.

Esta tesis se enmarca dentro del programa Horizonte Europa, y se centra en el desarrollo de técnicas de Inteligencia Artificial para construir una herramienta para el modelado y predicción de consumo energético en edificios. En concreto, perseguimos el desarrollo de una técnica interpretable que sirva como herramienta de ayuda en la toma de decisiones para el experto en la gestión de energía, siendo de utilidad para reducir el consumo energético en el caso particular de las instalaciones de la Universidad de Granada.

Abstract

Achieving an efficient and sustainable energy consumption in the building sector has become one of the main challenges to be solved in this decade. In the transition towards a complete decarbonization in the use of energy, energy efficiency is positioned as a central tool to identify and avoid unnecessary consumption. Consequently, it is expected to reduce the carbon footprint as well as minimizing the risks of climate change. Thanks to sensor technology advances and the development of Artificial Intelligence techniques, the research community has focused its efforts on the development of intelligent systems to automatically extract useful knowledge from data related to energy consumption, enabling the identification of the most relevant factors that help reduce energy consumption.

This thesis is part of the Horizon Europe program, and focuses on the development of Artificial Intelligence techniques to build a tool for modelling and forecasting energy consumption in buildings. More specifically, we attempt to develop an interpretable technique that helps the expert in energy management to make decisions, being useful to reduce energy consumption in the particular case of the facilities of the University of Granada.

Índice general

Índice de tablas

Índice de figuras

Artículo 1

Artículo 2

Artículo 3

Artículo 4

Capítulo 1

Memoria

1.1. Introducción

Debido al calentamiento global y al agotamiento de los recursos naturales, la eficiencia energética se ha posicionado como centro de atención en esta última década tanto para gobiernos como para grandes empresas. Como consecuencia, la Unión Europea ha centrado gran parte de sus actividades de investigación en un programa denominado Horizonte Europa [Com], dirigido exclusivamente a que investigadores de toda Europa concentren sus esfuerzos para encontrar una solución al calentamiento global y a la gestión eficiente de energía. Según la Directiva 2012/27/UE, el sector de los edificios representa un 40 % del consumo energético final de la Unión Europea, destacando que el elevado consumo en los edificios proviene principalmente de los sistemas de calefacción, ventilación, aire acondicionado e iluminación [Age19a]. Además, según la Agencia Internacional de Energía (AIE) [Age], los edificios y el sector de la construcción son los responsables de alrededor de un 40 % de las emisiones de CO_2. En este sentido, la Agencia Internacional de las Energías Renovables (IRENA) declara en el informe [Age19b] que el sector de la edificación sigue siendo estratégico para conseguir una descarbonización completa para 2050, centrando su interés en el diseño eficiente de edificios para reducir sus emisiones incorporadas y mejorar su gestión eficiente de energía. Para lograrlo, la AIE está implementando una serie de medidas para que el sistema energético mundial cumpla los principales objetivos establecidos en el Acuerdo de París para reducir las emisiones contaminantes y garantizar el acceso universal de energía [Cha16].

La solución a este problema es muy compleja, y requiere que grandes instituciones, públicas y privadas, colaboren para solucionarlo. En España, la Directiva 2010/31/UE establece que a partir

de 2020 todos los edificios de nueva construcción deben ser edificios de consumo energético casi nulo, apoyándose en medidas orientadas a una mayor eficiencia y ahorro en el consumo de energía en edificios de nueva construcción, y fomentando la mejora de eficiencia energética de los edificios existentes [Tra14]. Esto es posible proporcionando a los edificios las herramientas necesarias para mejorar su eficiencia energética, dotándoles de un mayor conocimiento, y por tanto control, de los sucesos implicados con el consumo energético que ocurren en las instalaciones. En este sentido, las redes de sensores inalámbricos pueden utilizarse como herramienta para monitorizar información potencialmente útil que pueda ocurrir dentro y fuera de los edificios [Sur+15]. Esta información puede ser analizada y extraer conocimiento útil para mejorar la eficiencia energética. Considerando el reciente avance en tecnologías de la computación para procesar grandes cantidades de datos, han surgido nuevas técnicas de Inteligencia Artificial (IA) y Minería de Datos para descubrir de forma automática conocimiento no trivial de los datos. En concreto, en el ámbito de la gestión de energía en edificios, han surgido diversas técnicas para resolver problemas relacionados con la predicción de la demanda de energía, para adaptar su producción y distribución; o la detección de patrones de consumo energético para detectar fraude [Mol+17]. Sin embargo, las técnicas desarrolladas para resolver este tipo de problemas, han demostrado que, a pesar de su potencial, son insuficientes cuando el tamaño del conjunto de datos sobre el consumo energético es elevado o cuando los datos proceden de fuentes heterogéneas. Como consecuencia, el desarrollo de técnicas avanzadas de IA para dar solución a este problema se ha posicionado como foco de interés por parte de empresas, instituciones e investigadores. En concreto, los principales avances se centran en el uso de redes neuronales o técnicas de *Deep Learning* [Rui+16; Rui+18; KC19; Li+17; Liu+19], debido a que son técnicas que ofrecen soluciones muy precisas ante grandes cantidades de datos, en un tiempo razonable. Sin embargo, éstos métodos suelen considerarse modelos de caja negra; lo que significa que las soluciones provistas son demasiado complicadas para la comprensión humana. Como se discute en [Rud19], la falta de transparencia de los modelos predictivos pueden desencadenar graves consecuencias. Por ejemplo, se han desarrollado modelos que han afirmado que el aire altamente contaminado era seguro para respirar [MCG18]. Sin embargo, en lugar de crear modelos interpretables, ha habido un gran interés en el desarrollo de Machine Learning (ML) explicable donde se crea un

segundo modelo para explicar el primer modelo de caja negra [Gui+18]. No obstante, como discute Rudin [Rud19], si es posible desarrollar un modelo explicativo para comprender un modelo de caja negra, entonces es posible crear un modelo inherentemente interpretable capaz de obtener resultados igual de precisos, por lo que supone actualmente un desafío para la comunidad investigadora. Puesto que una falta de transparencia en los modelos de caja negra de IA reduce la confianza por parte de los usuarios finales [VBC18; Gre18], es importante centrar los esfuerzos en diseñar y desarrollar nuevas técnicas de IA que consigan resultados tan buenos como los modelos de caja negra, pero que a su vez sean altamente interpretables.

En este contexto, este proyecto de tesis propone el diseño y desarrollo de técnicas de IA que incorporen equilibrio entre eficacia e interpretabilidad para analizar información proveniente de diversas fuentes de datos, con el objetivo de ayudar al usuario final a comprender cómo y cuándo se consume la energía, proporcionando una herramienta que sirva de ayuda para la mejora de eficiencia energética. Más concretamente, en esta tesis se plantea el estudio de técnicas del estado del arte actual en Ciencia de Datos aplicadas en el área de la eficiencia energética, y la propuesta de técnicas de *Soft Computing* que permitan mejorar las técnicas actuales para modelado, predicción y explicación del consumo energético considerando, principalmente, problemas multivariantes donde existen múltiples edificios. Entre las técnicas que se asumen como más prometedoras, encontramos las siguientes:

- Metaheurísticas. Las metaheurísticas han tenido una amplia gama de problemas en las que han sido aplicadas. En particular, en el ámbito de la energía, podemos encontrar múltiples propuestas en la literatura, entre las que destacamos la resolución de problemas de clasificación, selección de características, agrupamiento multidimensional o detección de anomalías, entre otros. Por mencionar algunos ejemplos, en el trabajo [GKB17] desarrollaron una metaheurística híbrida (que combina el algoritmo k vecinos más cercanos junto con el algoritmo de escalada) que utiliza datos relacionados con el consumo eléctrico, información meteorológica y la ocupación en edificios para optimizar el uso de energía de los sistemas de calefacción, iluminación, ventilación y aire acondicionado. Por otro lado, en el trabajo [Zaf+17] utilizaron diferentes metaheurísticas (*Harmony Search Algorithm,*

Bacterial Foraging Optimization, Enhanced Deferential Evolution) para establecer un horario que optimizara el uso de los electrodomésticos en el hogar para minimizar el gasto de energía, atendiendo al consumo eléctrico de cada dispositivo, el precio de la energía a lo largo de las horas del día y las necesidades de los usuarios.

- Inferencia Gramatical [SC14]. La inferencia gramatical es uno de los problemas clásicos dentro del área del reconocimiento de patrones. En particular, ha sido una técnica ampliamente estudiada en múltiples aplicaciones, entre las que destacamos el modelado y reconocimiento de ADN, modelado de Series Temporales o Programación Genética. Es esta última técnica la que principalmente se desarrollará en esta tesis, para establecer dependencias y modelos del consumo energético que permitan identificar similitudes y diferencias entre diversos tipos de perfiles de consumo.

Finalmente, esta memoria consiste en el reagrupamiento de los trabajos de investigación publicados en medios científicos de alto impacto. Por esta razón, la presente tesis ha sido organizada en dos bloques principales: la tesis doctoral y las publicaciones. En esta primera parte, describimos una introducción del contexto general de este proyecto. En la Sección 1.3 analizamos en detalle los conceptos que respaldan esta tesis: Series Temporales, Regresión Simbólica, Programación Genética y *Straight Line Programs*. La metodología empleada para llevar a cabo el desarrollo de esta tesis doctoral se muestra en la Sección 1.4. En la Sección 1.5 introducimos una descripción de las principales publicaciones derivadas de esta tesis, finalizando con un análisis de los resultados obtenidos durante el período de investigación, en la Sección 1.6. Por último, la Sección 1.7 recoge las conclusiones generales obtenidas en esta tesis, así como las futuras líneas de trabajo que deja abiertas este proyecto.

La última parte de esta tesis se muestra en el Capítulo 2, y recoge las cuatro publicaciones realizadas en revistas de alto impacto:

- R.Rueda, M.P.Cuéllar, M.C.Pegalajar, M.Delgado (2019). Straight Line Programs for Energy Consumption Modelling. Applied Soft Computing, 80, 310-328

- R.Rueda, L.G.B.Ruiz, M.P.Cuéllar, M.C.Pegalajar (2020). An Ant Colony approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling. International Journal of Approximate Reasoning, 121, 23-38
- R.Rueda, M.P.Cuéllar, M.Molina-Solana, Y.Guo, M.C.Pegalajar (2019). Generalised Regression Hypothesis Induction for Energy Consumption Forecasting. Energies, 12, 1-22
- R.Rueda, M.P.Cuéllar, L.G.B.Ruiz, M.C.Pegalajar. A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming

Introduction

Due to global warming and the shortage of natural resources, energy efficiency is gaining special interest in the last decade for both governments and companies. As a result, the European Union has focused its research activities on a programme called Horizon Europe [Com], which is aimed exclusively at enabling researchers across Europe to find a solution to global warming and efficient energy management. According to Directive 2012/27/UE, the building sector accounts for 40 % of final energy consumption in the European Union, highlighting that high consumption in buildings comes mainly from HVAC systems (heating, ventilation, air-conditioning and lighting systems) [Age19a]. Moreover, according to the International Energy Agency (IEA) [Age], buildings and the building sector are responsible for about 40 % of the CO_2 emissions. In this sense, the International Renewable Energy Agency (IRENA) states in the report [Age19b] that the building sector remains strategic to achieve a complete decarbonization by 2050, focusing its interest on efficient building design to reduce its embedded emissions and improve its energy efficiency management. Consequently, the IEA is implementing a set of measures to guarantee that the global energy system meets the main objectives set out in the Paris Agreement to reduce pollutant emissions and ensure universal energy access [Cha16].

The solution to this problem is complex, and it requires that both public and private institutions work together to solve it. In Spain, Directive 2013/31/UE establishes that from 2020 all new buildings must be near-zero consumption buildings, supported by measures aimed at greater

efficiency and savings in energy consumption in new buildings, and promoting the improvement of energy efficiency in existing buildings [Tra14]. This is possible by providing buildings with the tools to improve their energy efficiency, providing them greater knowledge, and therefore control, of the events involved with the energy consumption that occurs in the facilities.

In this sense, Wireless Sensor Networks can be used as a tool to monitor useful events that may occur inside and outside of buildings [Sur+15]. This information can be analyzed to extract knowledge and improve energy efficiency. Taking into account the recent advances in computer technologies to process large amounts of data, new techniques of Artificial Intelligence (AI) and Data Mining have emerged to automatically discover non-trivial knowledge of the data. More specifically, in the field of building energy management, several techniques have emerged to solve problems related to the prediction of energy demand, to adapt its production and distribution, or energy consumption pattern recognition to detect fraud [Mol+17]. However, the techniques developed to solve this kind of problems have shown that, despite their potential, they are insufficient when the size of the energy consumption dataset is high or when the data comes from heterogenous sources. As a result, the development of advanced AI techniques to solve this problem has become a focus of interest for companies, institutions and researchers. In particular, the main advances are focused on the use of neural networks or Deep Learning techniques [Rui+16; Rui+18; KC19; Li+17; Liu+19], because they offer accurate solutions to large amounts of data, in a reasonable time. However, these methods are usually considered as black box models; which means that the solutions provided are difficult to understand by humans. As discussed in [Rud19], the lack of transparency of predictive models can trigger serious consequences. For example, models have been developed that have claimed that highly polluted air was safe to breathe [MCG18]. However, rather than creating interpretable models, researchers have focused on developing explainable ML where a second model is created to explain the first black box model [Gui+18]. However, as Rudin discusses [Rud19], if it is possible to develop an explanatory model to understand a black box model, then it is possible to create an inherently interpretable model that is capable of obtaining equally accurate results. Currently, this is a challenge for the research community. Since a lack of transparency in black box models of AI reduces the confidence of end users [VBC18; Gre18], it is important to design and develop

new interpretable techniques able to achieve results as good as black box models.

In this context, this thesis proposes the design and development of efficient and interpretable AI techniques to analyze information from various data sources, with the aim of helping the end user to understand how and when energy is consumed, providing a tool to help improve energy efficiency. More specifically, this thesis proposes the study of state-of-the-art techniques in Data Science applied to energy efficiency, and the proposal of Soft Computing methods to improve current techniques for modelling, forecasting and explaining energy consumption, mainly considering multivariate problems where there are multiple buildings. Among the techniques assumed to be the most promising, we highlight the following:

- Metaheuristics. Metaheuristics have had a wide range of problems in which they have been applied. In particular, in the field of energy, we can find multiple proposal in the literature, among which we highlight the resolution of classification problems, feature selection, clustering or anomaly detection, among others. To give some examples, in the work [GKB17] they developed a hybrid metaheuristic (combining k-nearest neighbors and Hill Climbing algorithms) that uses data related to electricity consumption, weather information and building occupancy to optimize the energy use of HVAC systems. On the other hand, the work [Zaf+17] used different metaheuristics (Harmony Search Algorithm, Bacterial Foraging Optimization and Enhanced Deferential Evolution) to optimize a schedule that determine the use of household appliances to minimize energy consumption, taking into account the energy consumption of each device, the energy price throughout the day and the needs of users.

- Grammatical inference [SC14]. Grammatical inference is one of the classic problems in pattern recognition problems. In particular, it has been a technique widely studied in multiple applications, such as the modelling and recognition of DNA, Time Series modelling or Genetic Programming. This last technique will be mainly developed in this thesis, in order to establish dependencies and models of energy consumption that allow detecting similarities and differences between different kinds of consumption profiles.

Finally, this report consists of a summary of research papers published in high-impact science journals. For this reason, this thesis has been organized in two main blocks: the doctoral thesis and the publications. In this first part, we describe an introduction to the general context of this project. Section 1.3 studies in detail the concepts that support this thesis: Time Series, Symbolic Regression, Genetic Programming and Straight Line Programs. The methodology used to carry out the development of this doctoral thesis is shown in Section 1.4. Section 1.5 introduces a description of the main publications derived from this thesis, including an analysis of the results obtained during the research period, in Section 1.6. Finally, Section 1.7 gathers the general conclusions obtained in this thesis and future work.

The last part of this thesis is shown in Chapter 2, and includes the publications made in high impact journals:

- R.Rueda, M.P.Cuéllar, M.C.Pegalajar, M.Delgado (2019). Straight Line Programs for Energy Consumption Modelling. Applied Soft Computing, 80, 310-328
- R.Rueda, L.G.B.Ruiz, M.P.Cuéllar, M.C.Pegalajar (2020). An Ant Colony approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling. International Journal of Approximate Reasoning, 121, 23-38
- R.Rueda, M.P.Cuéllar, M.Molina-Solana, Y.Guo, M.C.Pegalajar (2019). Generalised Regression Hypothesis Induction for Energy Consumption Forecasting. Energies, 12, 1-22
- R.Rueda, M.P.Cuéllar, L.G.B.Ruiz, M.C.Pegalajar. A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming

1.2. Objetivos

A modo de resumen, el **principal objetivo** de esta tesis consiste en el análisis, diseño y desarrollo de nuevas metodologías de Ciencias de Datos e IA para el análisis de Series Temporales procedentes del consumo energético, particularizando en el caso de estudio de la Universidad de

Granada, de modo que se mejore y enriquezca el conjunto de técnicas aplicables al soporte en la toma de decisiones que permitan la mejora de la eficiencia energética. Para lograrlo, hemos de cumplir los siguientes **objetivos específicos**:

- El primer objetivo consiste en el análisis, diseño, implementación y validación de algoritmos de preprocesamiento para el análisis de datos de consumo energético. Se tratará el problema de tratamiento de ruido, cambio de dimensionalidad y dominio de los datos, fusión de datos de bajo nivel y selección de características. La finalidad de este objetivo parcial consiste en obtener un conjunto de datos robusto y disponible para poder ser procesado por técnicas de Minería de Datos y *Machine Learning* de más alto nivel.

- El segundo objetivo versa sobre el estudio de las técnicas del estado del arte utilizadas para resolver los problemas de modelado y predicción de consumo energético. Posteriormente, procederemos al análisis, diseño y desarrollo de modelos de *Soft Computing* capaces de modelar el consumo energético de forma precisa. Además, se persigue el desarrollo de un modelo interpretable que ayude al usuario final a la toma de decisiones en la gestión del consumo energético. Este modelado, permitirá extraer conocimiento no trivial de forma automática y detectar los factores más relacionados con el consumo de un determinado edificio.

- En el tercer objetivo se procede al análisis, diseño y desarrollo de técnicas de minería de datos para el modelado de series temporales procedentes de datos de consumo energético multivariante. Se considerarán casos en los que existen varios edificios, para los que se dispondrán también de variables exógenas a los propios datos de consumo.

- Por último, el cuarto objetivo consiste en el diseño y desarrollo de mecanismos para mejorar la capacidad de exploración de los algoritmos de optimización previamente desarrollados, para modelar el consumo energético.

1.3. Antecedentes

Esta sección recoge los principales conceptos de las técnicas utilizadas en esta tesis. En primer lugar, la Sección 1.3.1 describe el concepto de Serie Temporal, su modelado y aplicación en el ámbito de la eficiencia energética. A continuación, introducimos las técnicas de Regresión Simbólica y Programación Genética en las Secciones 1.3.2 y 1.3.3 respectivamente. Por último, en la Sección 1.3.4 mostramos cómo la representación Straight Line Programs puede ser utilizada para representar expresiones algebraicas en el problema de Regresión Simbólica.

1.3.1. Series Temporales

Una Serie Temporal [BD16] es un conjunto ordenado de observaciones de uno o varios fenómenos registradas secuencialmente en el tiempo, usualmente a intervalos regulares. Las series temporales se caracterizan porque no sólo dependen de la variable tiempo, sino también de valores de la misma serie, registrados en instantes de tiempo anteriores al actual. Adicionalmente, una serie temporal también puede presentar dependencia con otras variables temporales externas. El análisis y predicción de series temporales es un problema ampliamente estudiado en múltiples disciplinas; por ejemplo, en meteorología observamos los cambios de temperatura diaria o las precipitaciones anuales en una determinada zona [Soa+18]. En epidemiología, el registro diario de infectados por *COVID-19* puede ser de utilidad para crear modelos de predicción que ayuden al gobierno y al personal médico a estar preparados en los sistemas de salud [Ben+20]. En medioambiente, se registra la evolución horaria de los niveles de dióxido de azufre y dióxido de nitrógeno en una ciudad para determinar los niveles de contaminación [Goc+14]. El propósito del análisis de series temporales consiste en modelar el proceso estocástico que da lugar a una determinada serie, así como predecir valores futuros basándose en registros históricos.

Formalmente, definimos una serie temporal como un conjunto de observaciones $X = \{x_{t_1}, x_{t_2}, \ldots, x_{t_n}\}$, $(t_1 < t_2 < \ldots < t_n)$, donde t_i representa el instante de tiempo en el que se registró la muestra x_{t_i}. Un ejemplo de serie temporal se muestra en la imagen 1.1, donde se muestran los

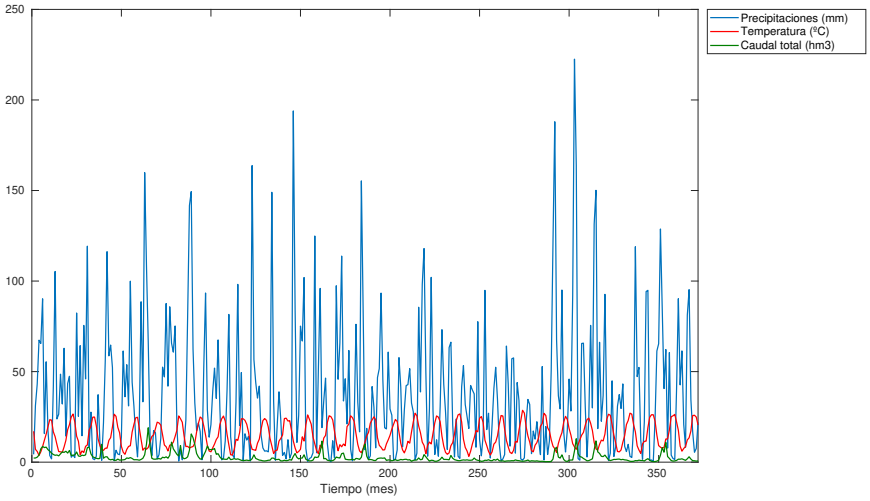


Figura 1.1: Ejemplo de series temporales. Variables de temperatura, precipitaciones y caudal del río Cubillas desde 1980 hasta 2008.

valores registrados de precipitaciones, temperatura y caudal para modelar el caudal total del río Cubillas.

Las herramientas para analizar una serie temporal se basan en la idea de descomponer la variación de una serie en varias componentes básicas. En el modelado clásico de series temporales con la metodología Box-Jenkins [BRJ94] con enfoque aditivo, las componentes que son analizadas en una serie son: (1) **Tendencia**, muestra la variabilidad en los datos a largo plazo. (2) **Estacionalidad**, refleja fluctuaciones periódicas en la serie temporal, es decir, datos afectados por un patrón estacional tales como el día del año o el día de la semana. (3) **Componente residual**. Una vez identificados y eliminados las componentes de tendencia y estacionalidad de la serie, persisten unos valores que son residuales. De las tres componentes de la serie, las dos primeras son determinísticas, mientras que la última es estocástica. De este modo, bajo un enfoque aditivo, podemos denotar una serie temporal como $X = T + E + I$, donde T es la tendencia, E refleja la estacionalidad e I hace referencia al residuo o componente aleatoria de la serie.

Los modelos clásicos para modelado y predicción de series temporales son resueltos utilizando técnicas de regresión y modelado de Box-Jenkins [BRJ94], donde tratan de estimar las componentes de la serie para conocer el comportamiento de la misma a largo plazo. Sin embargo, este tipo de técnicas presentan limitaciones; en el caso de los modelos de *Box-Jenkins*, no hacen

buenas predicciones si no cuentan con suficientes datos, algunos autores difieren en torno al número de observaciones necesarias, pero es recomendable que existan al menos 50 [YM00]. En el caso de las técnicas de regresión, estas presentan limitaciones si la función que modela los datos es desconocida y es difícil de aproximar. Por ello, es necesario recurrir a técnicas como Redes Neuronales Artificiales o Regresión Simbólica [KV98] para resolverlas. En esta tesis nos centraremos en el estudio de Regresión Simbólica como técnica para modelar y predecir series temporales de consumo energético.

1.3.2. Regresión Simbólica

El análisis de regresiones [CH12] es un método estadístico que permite encontrar relaciones entre dos o más variables. Por ejemplo, la relación entre la temperatura de un edificio y su ocupación con su consumo energético. El análisis de regresiones expresa esta relación mediante una ecuación o modelo que conecta la variable dependiente con una o más variables independientes. En el ejemplo anterior, la variable dependiente hace referencia al consumo energético, mientras que las variables independientes serían la temperatura y la ocupación. En este sentido, el análisis de regresiones trata de encontrar las relaciones entre las variables dependientes e independientes a través de una función.

Formalmente, el análisis de regresiones se compone por un modelo de hipótesis $f(\bar{x}, \bar{w}) + \epsilon$, un conjunto de datos de entrada o variables independientes $\bar{x} = \{x_1, x_2, \ldots, x_n\}$, un conjunto de datos de salida o variable dependiente $\bar{y} = \{y_1, y_2, \ldots, y_m\}$, un conjunto de parámetros $\bar{w} = \{w_1, w_2, \ldots, w_k\}$, y un error ϵ que representa la parte de los datos que el modelo $f(\bar{x}, \bar{w})$ no es capaz de modelar, debido a las propias condiciones del fenómeno a observar, o resolución de los sensores. El principal objetivo del análisis de regresiones es aproximar los mejores valores de los parámetros $\bar{w}$ tal que $\bar{y} \approx f(\bar{x}, \bar{w})$. Los parámetros $\bar{w}$ suelen ser estimados utilizando procesos numéricos, como optimización de mínimos cuadrados, que minimizan una medida de error $e(f, \bar{y}) = ||\bar{y} - f(\bar{x}, \bar{w})||$, como la suma de los errores al cuadrado entre el modelo estimado $f(\bar{x}, \bar{w})$ y la respuesta $\bar{y}$.

Sin embargo, la principal limitación del análisis de regresiones surge cuando los parámetros $\bar{w}$ y

el modelo de hipótesis f son desconocidos y difíciles de hallar por métodos tradicionales. Para dar solución a este problema, se hace uso de Regresión Simbólica (RS), que combina un conjunto predefinido de operadores atómicos (como por ejemplo $+, -, *, /$), variables independientes ($\bar{x}$) y parámetros ($\bar{w}$) para construir una expresión algebraica $\tilde{f}$ como aproximación de los valores de salida $\bar{y}$. El proceso para buscar el mejor modelo $\tilde{f}$ consiste en realizar todas las combinaciones posibles entre los los datos de entrada, parámetros y operadores atómicos para hallar aquella aproximación que minimice una medida de error, como $||\bar{y} - \tilde{f}(\bar{x}, \bar{w})||$. El principal inconveniente de RS es que el número de combinaciones posibles es elevado, además de que puede existir más de un modelo que se ajuste al conjunto de datos. Esto significa que RS, en comparación con técnicas clásicas de regresión, necesitará más tiempo de cómputo para encontrar una solución. Para facilitar esta tarea, se hace uso de algoritmos de optimización que ayudan a explorar el espacio de búsqueda para encontrar el mejor modelo de regresión. En este sentido, la resolución del problema de RS ha sido abordado por medio de diferentes técnicas de *Soft Computing*, desde métodos Bayesianos [Jin+19], hasta técnicas pertenecientes a la familia de los Algoritmos Evolutivos, como métodos de Inteligencia de Enjambre [Kar+12] o Programación Genética [Koz92]. Puesto que la Programación Genética ha sido la técnica más utilizada por la comunidad científica para resolver el problema de RS, obteniendo resultados prometedores, será la que utilizaremos en esta tesis como punto de partida de la investigación.

1.3.3. Programación Genética

La Programación Genética (PG) es un método de aprendizaje supervisado basado en los Algoritmos Genéticos, cuya principal diferencia reside en la representación de las soluciones. Los Algoritmos Genéticos (AG) son algoritmos de optimización, búsqueda y aprendizaje inspirados en los procesos de evolución natural y evolución genética, propuestos por *John Henry Holland* en 1975 [Hol75]. Como sucede en la naturaleza, un conjunto de individuos habitan en un determinado entorno con recursos limitados y compiten entre ellos para alcanzar un determinado objetivo: reproducirse y sobrevivir. La selección natural juega un papel esencial para determinar qué individuos sobrevivirán, consiguiéndolo solo aquellos que mejor se adapten al entorno.

Análogamente, en el ámbito de la IA, el conjunto de individuos son representados haciendo uso de estructuras de datos (como vectores, matrices, árboles binarios, etc) y la destreza de cada individuo se define por medio de una función matemática, comúnmente denominada como *fitness*. Este valor *fitness* refleja las oportunidades que tiene un determinado individuo a reproducirse y a sobrevivir en futuras generaciones.

La figura 1.2 muestra el esquema general de un algoritmo genético. El algoritmo comienza con una población inicial generada, normalmente de forma aleatoria. Esta población está formada por conjunto de individuos que representan una solución válida del problema, aunque no óptima. Posteriormente, se evalúan todos los individuos, en términos de un valor numérico que determina su calidad (*fitness*). Tras evaluar la población, se hace uso de un operador de selección para escoger de entre todos los individuos, aquellos que presentan unas características más prometedoras para combinar su material genético y dar lugar a nuevos individuos. Finalmente, con el objetivo de imitar el comportamiento de la evolución natural, la nueva población obtenida reemplazará la población anterior. Este procedimiento se repite hasta que se cumple un criterio de parada definido *a priori* por el usuario. Este criterio de parada puede establecerse atendiendo a un número limitado de evaluaciones o si se encuentra la solución óptima al problema.

Cabe destacar que existen dos factores principales que determinarán la calidad de la solución encontrada: por un lado, los mecanismos de reproducción para dar lugar a una nueva población de individuos, aumentan la diversidad necesaria para realizar una mejor exploración del espacio de búsqueda; mientras que los mecanismos de selección de padres y reemplazo, están diseñados para explotar las soluciones candidatas y por tanto aumentar la convergencia del algoritmo hacia una solución. Hallar la mejor combinación entre divergencia y convergencia garantiza una mejor exploración del espacio de búsqueda, evitando caer en óptimos locales.

Del esquema anterior, podemos destacar 2 componentes básicas que han de ser analizadas en detalle, pues determinarán el balance entre la exploración y explotación del espacio de soluciones [ES03]: la representación del problema y los operadores genéticos.

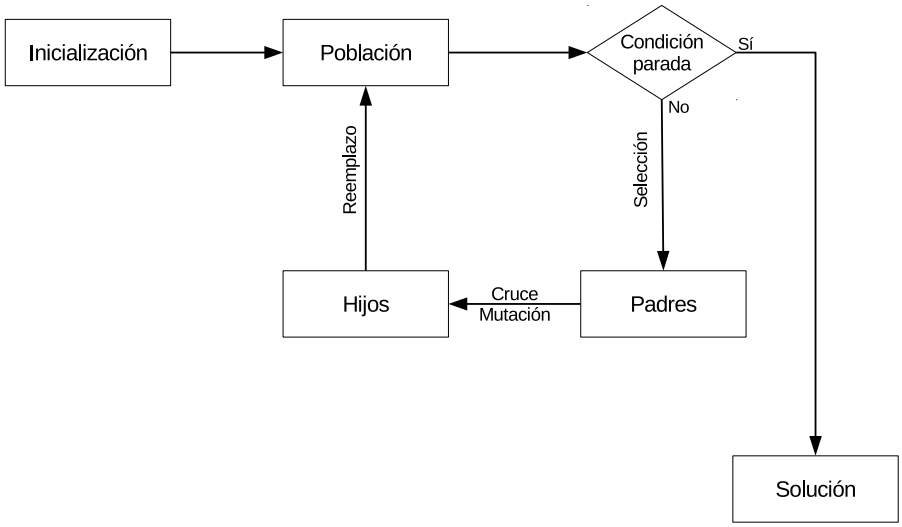


Figura 1.2: Esquema Algoritmo Genético.

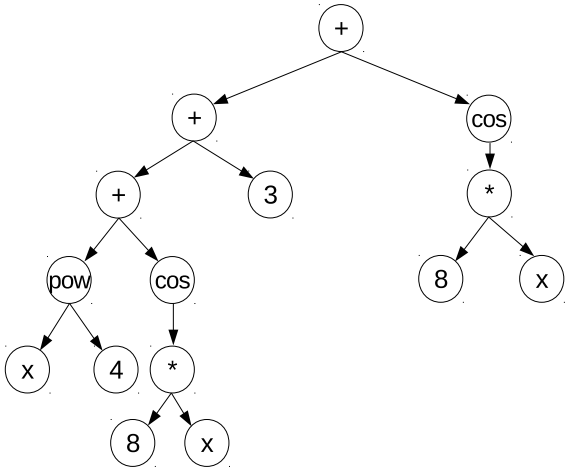


Figura 1.3: Ejemplo de expresión algebraica codificada con una estructura simbólica de tipo árbol.

1.3.3.1. Representación del problema

El primer paso para diseñar un algoritmo de PG consiste en encontrar una representación que conecte el contexto del problema real con el problema a resolver por PG. La estructura utilizada por PG se conoce como programas de ordenador, representados tradicionalmente por estructuras de tipo árbol. Cada nodo del árbol se compone de una función como operador (como por ejemplo, operadores aritméticos, funciones matemáticas, operaciones booleanas, etc.) y cada nodo hoja tiene un operando o terminal (variables, parámetros, etc). De este modo, esta estructura puede representar desde expresiones algebraicas (ver Figura 1.3), hasta programas de un determinado lenguaje de programación.

Un aspecto importante a considerar en PG es que el conjunto de funciones y terminales utilizados para representar un individuo, deben cumplir las propiedades de **clausura** y **suficiencia** [Koz92]. La propiedad de clausura requiere que las funciones utilizadas deben aceptar cualquier entrada del conjunto de terminales. Por ejemplo, en el caso particular de representar expresiones algebraicas, el algoritmo de PG debe estar provisto de algún mecanismo para garantizar operaciones como la división por 0. Por otro lado, el criterio de suficiencia hace referencia a que el conjunto de funciones y terminales deben ser capaces de expresar una solución al problema.

Una vez que los individuos de la población son representados con alguna estructura, por ejemplo, árboles, es necesario definir una función de evaluación para determinar la calidad de cada individuo. Es decir, queremos desarrollar una medida que nos indique cómo de bien resuelve un individuo el problema que tratamos de resolver. Este valor se utilizará como criterio para seleccionar los individuos más prometedores y determinará cuáles prevalecerán en la población en generaciones futuras. Esta medida es comúnmente conocida como *fitness*. Cabe destacar que si un individuo cumple la propiedad de clausura, la evaluación de los individuos funcionará correctamente.

Por último, definimos como población al conjunto de n individuos $I_t = \{I_{1_t}, I_{2_t}, \dots I_{n_t}\}$, donde cada individuo I_{k_t} es la representación k-ésima de una posible solución al problema en la generación t. Al igual que en la evolución natural, definimos el término generación como la época en la que vive un conjunto de individuos, en otras palabras, es el instante de tiempo en el que el conjunto de individuos de I son seleccionados, cruzados y mutados con el objetivo de mejorar el *fitness* de cada individuo. Cada vez que una nueva población sustituye a la población anterior, decimos que ha ocurrido un cambio generacional. El tamaño de la población n es un parámetro del algoritmo que debe ser fijado *a priori* por el usuario.

1.3.3.2. Operadores genéticos

Los operadores genéticos son la base del proceso de selección natural y supervivencia del más fuerte. Podemos clasificar los operadores genéticos en tres operadores principales: selección, reproducción y reemplazo. En primer lugar, el operador de **selección** se encarga de seleccionar

de entre todos los individuos de la población, aquellos más prometedores para ser candidatos a reproducirse y crear los individuos que formarán parte de las siguientes generaciones. Esta selección tiene en cuenta el *fitness* de los individuos candidatos a ser elegidos y suele realizarse de forma probabilística, de modo que los individuos con mejor *fitness* tendrán mas posibilidades de ser elegidos frente a los individuos de peor calidad. Sin embargo, los individuos con un *fitness* inferior tienen la oportunidad de ser elegidos, lo que beneficia a la exploración del espacio de búsqueda, aportando una mayor diversidad a la población.

Por otro lado, el operador de **reproducción** imita el comportamiento de la selección natural, de modo que los individuos escogidos por el operador de selección tomarán el rol de *padres*, con el propósito de generar una nueva población de nuevos individuos (hijos), como resultado de combinar o modificar el material genético de sus progenitores. Los operadores de reproducción se dividen en dos operadores: **mutación** y **cruce**. En primer lugar, el operador de mutación es un operador estocástico, que se aplica sobre un individuo para modificar uno de sus genes. Este operador es aplicado durante el ciclo genético con una baja probabilidad, con el propósito de imitar el comportamiento de la evolución biológica y aumentar la diversidad entre los individuos. En segundo lugar, el operador de cruce combina la información de dos individuos (padres) para generar uno o dos individuos nuevos (hijos). Los hijos generados competirán entre ellos (atendiendo a su valor *fitness*) para ocupar su lugar en la siguiente generación de individuos. Al igual que el operador de mutación, el operador de cruce es un operador estocástico en el sentido en que las partes de los padres que son seleccionadas para ser combinadas son seleccionadas de forma aleatoria.

Por último, el operador de **reemplazo** identifica los mejores individuos de la población en base a su calidad (*fitness*) para construir la siguiente generación de individuos. A diferencia del operador de selección, el operador de reemplazo es determinista y es utilizado sobre la nueva población de individuos, resultante de aplicar los operadores de reproducción, para seleccionar los candidatos a formar parte de la siguiente generación. Existen diversas estrategias de reemplazo, y su diseño podría influir en convergencias prematuras durante la exploración del espacio de soluciones. De entre todas las estrategias disponibles en la literatura, destacamos los enfoques *generacional* y *estacionario*, donde en el primero la población de hijos generada reemplazará la

generación anterior y en el segundo se permite que padres e hijos convivan simultáneamente en próximas generaciones, ya que un nuevo individuo optará a formar parte de la población si su calidad supera cierto umbral. En ambos enfoques puede incluirse además un criterio *elitista*, en el que se pretende conservar los mejores individuos encontrados en cada generación.

Cabe destacar que no existe una definición generalizada para cada operador que garantice encontrar una solución óptima, sino que existen diferentes diseños e implementaciones para cada componente, que han de ser diseñados de forma rigurosa atendiendo al problema a resolver. En concreto, la representación del problema delimitará el diseño de los operadores de cruce y mutación. Experimentalmente,la representación de tipo árbol presenta limitaciones, entre las que destacamos el problema de *bloating* [SC09; PLM08b], que consiste en el crecimiento descontrolado de los individuos sin mejorar el *fitness*. Para solucionar estas limitaciones, diferentes autores han propuesto diversas representaciones para codificar expresiones algebraicas, desde la representación de programas usando grafos codificados como *strings* lineales de enteros [MT00] o el uso de Matrices de Instrucciones para codificar los nodos de los árboles y sus subárboles de forma independiente [Li+08]. De entre todas las representaciones propuestas en la literatura, nosotros nos centraremos en el uso de gramáticas lineales, ya que creemos que una representación lineal provee de suficiente expresividad para representar una expresión algebraica sin reducir el espacio de búsqueda, además de facilitar un diseño simple y efectivo de los operadores genéticos en comparación con las representaciones de tipo árbol. En esta tesis doctoral estudiaremos el uso de estructuras alternativas a las de tipo árbol para codificar expresiones algebraicas y resolver el problema de RS, en concreto, nos centraremos en el uso de la gramática lineal *Straight Line Program* para representar expresiones algebraicas.

1.3.4. El uso de gramáticas en Programación Genética

Una gramática es una estructura lógico-matemática compuesta por un conjunto de reglas de formación que definen las cadenas de caracteres admisibles en un determinado lenguaje formal. Formalmente, definimos una gramática como una cuádrupla (V, T, P, S) en la que V es un alfabeto llamado variables o símbolos no terminales; T es un alfabeto llamado símbolos

terminales, P es un conjunto de pares (α, β) llamados reglas de producción. El par (α, β) se suele representar como $\alpha \to \beta$, donde α y $\beta \in (V \cup T)*$ y α contiene al menos un elemento de V, el símbolo $\to$ representa el proceso de derivación; S es un elemento de V llamado símbolo inicial. Una gramática sirve para determinar un lenguaje, como por ejemplo una oración en castellano o un programa de ordenador.

El uso de gramáticas ha sido ampliamente utilizado en Programación Genética desde la década de los 90 y han tenido un gran impacto en el desarrollo de nuevas aplicaciones [McK+10]. En concreto, el uso de gramáticas libres del contexto brindó nuevas oportunidades a PG, puesto que facilita el cumplimiento del criterio de clausura [Whi95]. Una gramática se dice que es independiente del contexto o de tipo 2 (atendiendo a la jerarquía de Chomsky [Cho56]) si y solo si todas las reglas de producción son de la forma $A \to \alpha$ donde $A \in V$ y $\alpha \in (V \cup T)^*$. En este trabajo, haremos uso de una herramienta con la capacidad expresiva de una gramática libre del contexto denominada Straight Line Program (SLP) propuesta por [BS84] y utilizada en [APM08] por primera vez en un algoritmo de PG. La analizaremos en detalle en la sección 1.3.4.1

1.3.4.1. Straight Line Programs

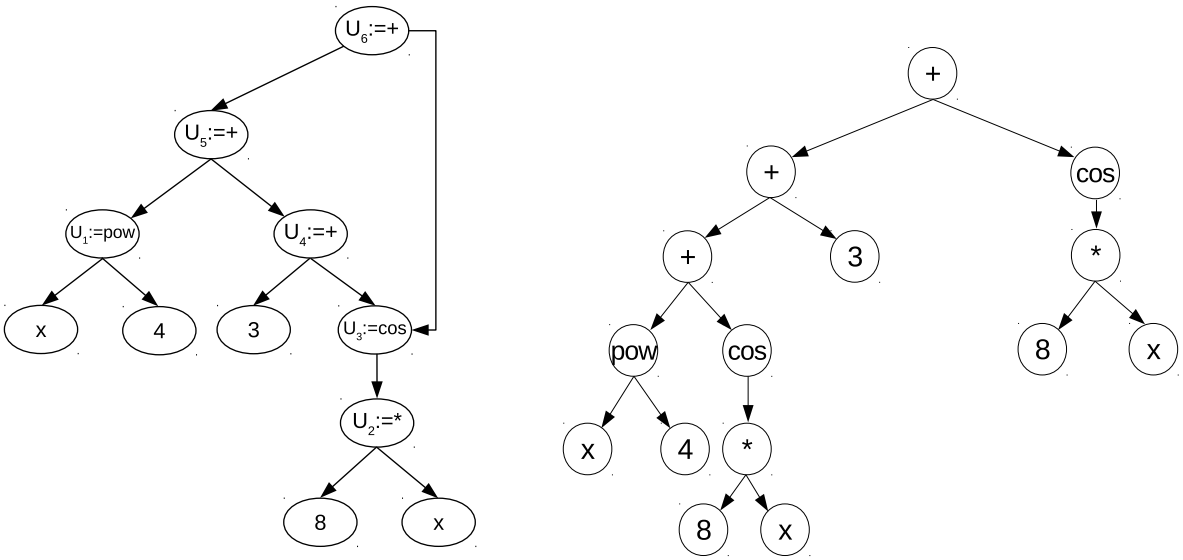
Una Straight Line Grammar (SLG) es una gramática no recursiva libre del contexto (V, T, P, S) capaz de generar un lenguaje de una sola palabra, donde V es el conjunto de símbolos no terminales de la gramática, T es el conjunto de símbolos terminales, S es un símbolo no terminal denominado símbolo inicial de la gramática y P es un conjunto finito de relaciones binarias de V a $(V \cup T)^*$. Un miembro de P es una regla de producción de la forma $A \to \alpha$, donde $A \in V$ es el antecedente de la regla y $\alpha \in (V \cup T)^*$ es el consecuente. Además, ya que un SLG es una gramática no recursiva, la generación de ciclos no está permitida y en consecuencia, solo se puede generar una única secuencia, lo que lo ha convertido en un área de interés para problemas como la complejidad de Kolmogorov o compresión de datos sin pérdida [BK13a].

Un SLP codifica un conjunto de reglas de producción de una SLG en Forma Normal de Chomsky, que puede ser utilizado en RS para generar una única expresión algebraica. En el problema

de RS abordado en esta tesis, el conjunto de símbolos terminales es $T = O_u \cup O_b \cup X \cup W$, donde O_u es el conjunto de operadores unarios, O_b es el conjunto de operadores binarios, X es el conjunto de variables de entrada $\{x_1, x_2, \ldots, x_n\}$, y W es el conjunto de parámetros $\{w_1, w_2, \ldots, w_k\}$. Un SLP tiene capacidad para representar N reglas de producción (P) de la forma $U_1, U_2, \ldots, U_N \in V$, donde U_N es el símbolo inicial de la gramática (S) y cada regla de producción es de la forma $U_i \to o_b r_{i1} r_{i2}$ o $U_i \to o_u r_{i1}$, donde $o_u \in O_u$ y $o_b \in O_b$ son operadores, y $r_{i1}, r_{i2} \in X \cup W \cup \{U_{i-1}, U_{i-2}, \ldots U_1\}$ son el primer y segundo operando respectivamente, que pueden ser un símbolo terminal (variable o parámetro) o un símbolo no terminal que referencia a reglas de producción inferiores para evitar recursividad.

Finalmente, la generación de la expresión algebraica codificada por un SLP se obtiene generando el símbolo inicial de la gramática U_N. Cada símbolo no terminal U_i es reemplazado iterativamente por su regla de producción asociada a $i = N - 1$ hasta $i = 1$. La fórmula 1.1 muestra un ejemplo de un SLP de tamaño $N = 10$ y parámetros $\bar{w} = w_1, w_2, w_3 = (4, 8, 3)$. Si aplicamos el procedimiento descrito anteriormente, la expresión algebraica es generada a través del símbolo inicial $\boldsymbol{U_{10}}$ como: $\tilde{f}(\bar{x}, \bar{w}) = \boldsymbol{U_{10}}$; $\boldsymbol{U_{10}} \Rightarrow U_9 + U_7 \Rightarrow (U_8 + U_3) + U_7 \Rightarrow ((U_5 + U_7) + U_3) + U_7 \Rightarrow ((U_5 + \cos(U_6)) + U_3) + \cos(U_6) \Rightarrow ((U_4^{U_1} + \cos(U_6)) + U_3) + \cos(U_6) \Rightarrow ((U_4^{U_1} + \cos(U_2 * U_4)) + U_3) + \cos(U_2 * U_4) \Rightarrow ((x^{w_1} + \cos(w_2 * x)) + w_3) + \cos(w_2 * x)$

$$
\begin{aligned}
U_1 &\to w_1 \\
U_2 &\to w_2 \\
U_3 &\to w_3 \\
U_4 &\to x \\
U_5 &\to pow\ U_4\ U_1 \\
U_6 &\to *\ U_2\ U_4 \\
U_7 &\to \cos\ U_6 \\
U_8 &\to +\ U_5\ U_7 \\
U_9 &\to +\ U_8\ U_3 \\
\boldsymbol{U_{10}} &\to +\ U_9\ U_7
\end{aligned}
\tag{1.1}
$$



(a) Representación en forma de DAG del SLP 1.1

(b) Representación tipo árbol

Figura 1.4: Comparativa entre las estructuras de tipo árbol y SLP para representar la expresión algebraica $((x^{w_1} + \cos(w_2 * x)) + w_3) + \cos(w_2 * x)$

Por otro lado, un SLP puede ser representado como un Grafo Dirigido Acíclico (DAG) (ver Figura 1.4a), lo que implica un gran potencial frente a las estructuras de tipo árbol. La figura 1.4 muestra la expresión $((x^{w_1} + \cos(w_2 * x)) + w_3) + \cos(w_2 * x)$ representada en forma de árbol y DAG. Aunque ambas aproximaciones tienen la misma capacidad expresiva en problemas de RS, la estructura lineal de un SLP frente a la no lineal del árbol de expresión, proporciona numerosas ventajas, como ha sido estudiado en [McK+10]. Esto implica una mejora en la función de evaluación, así como en el diseño de los operadores de cruce y mutación. Por ello, en esta tesis utilizamos la estructura SLP para representar expresiones algebraicas en PG para resolver el problema de RS.

1.4. Metodología

Para llevar a cabo esta tesis, es necesario seguir una metodología rigurosa dentro del método científico tradicional. Dado que los desarrollos de la investigación tienen una alta componente de experimentación y pruebas empíricas, se propone seguir los siguientes pasos:

1. **Observación.** Análisis del problema de la eficiencia energética y sus características específicas. Estudio y evaluación de las técnicas del estado del arte utilizadas para abordar el problema del modelado y predicción de series temporales de consumo energético.

2. **Formulación de hipótesis.** Teniendo en cuenta el análisis inicial realizado, se propone el diseño de nuevos métodos que sean de utilidad para el experto en la toma de decisiones para la mejora de la eficiencia energética. Los nuevos métodos desarrollados deben ajustarse a los objetivos descritos al inicio de este trabajo.

3. **Experimentación.** Análisis de los resultados obtenidos tras aplicar los modelos desarrollados sobre datos reales de consumo energético, procedentes de los edificios de la Universidad de Granada. En concreto, se analizará en detalle el equilibrio entre precisión e interpretabilidad de las soluciones obtenidas.

4. **Contraste de hipótesis.** Comparación y validación de los resultados obtenidos con los métodos propuestos frente a las técnicas del estado del arte utilizadas en problemas de modelado y predicción de consumo energético.

5. **Demostración o refutación de hipótesis.** Comprobar si las conclusiones obtenidas sobre cada uno de los métodos desarrollados se ajustan a los objetivos e hipótesis propuestas. Si los resultados obtenidos no satisfacen las expectativas iniciales, se retrocede al paso 2 y formulamos una nueva hipótesis.

6. **Extracción de tesis o teoría científica.** Formalizar las conclusiones obtenidas durante todo el proceso de investigación para crear una teoría que explique y justifique las técnicas desarrolladas en la experimentación. Cada uno de los modelos desarrollados a lo largo de esta investigación conformarán la presente tesis doctoral.

1.5. Resumen

Esta sección presenta las publicaciones realizadas durante la realización de esta tesis doctoral. En cada una de las siguientes subsecciones se muestra un resumen que describe la idea general del trabajo. A continuación se detallan los trabajos mencionados anteriormente.

- R.Rueda, M.P.Cuéllar, M.C.Pegalajar, M.Delgado (2019). Straight line programs for energy consumption modelling. Applied Soft Computing, 80, 310-328.
  DOI: 10.1016/j.asoc.2019.04.001.

- R.Rueda, L.G.B.Ruiz, M.P.Cuéllar, M.C.Pegalajar (2020). An Ant Colony approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling. International Journal of Approximate Reasoning, 121, 23-38.
  DOI: 10.1016/j.ijar.2020.03.005.

- R.Rueda, M.P.Cuéllar, M.Molina-Solana, Y.Guo, M.C.Pegalajar (2019). Generalised Regression Hypothesis Induction for Energy Consumption Forecasting. Energies, 12, 1-22.
  DOI: 10.3390/en12061069.

- R.Rueda, M.P.Cuéllar, L.G.B.Ruiz, M.C.Pegalajar. A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming.

El resto de la sección está organizado como sigue: En primer lugar, la Sección 1.5.1 muestra el estudio realizado para validar el potencial de la estructura SLP como representación de individuos en PG y modelar el consumo energético de la Universidad de Granada. En la Sección 1.5.2 discutimos la importancia de obtener un modelo interpretable que sirva de ayuda al experto en la toma de decisiones de alto nivel. La Sección 1.5.3 muestra cómo RS puede ser utilizada para resolver problemas de predicción de series temporales multivariante. Por último, la Sección 1.5.4 presenta una métrica desarrollada para comparar SLPs y mejorar la exploración del espacio de búsqueda en PG.

1.5.1. Regresión Simbólica y el modelado de consumo

Debido al aumento del consumo energético a nivel mundial, la comunidad investigadora ha tratado de comprender y modelar el consumo energético, con el propósito de conseguir un consumo eficiente y reducir el impacto medioambiental. Para lograrlo, se han desarrollado diferentes aplicaciones para comprender el consumo energético, con el objetivo de establecer una serie de medidas que ayuden a reducirlo. De entre las aplicaciones desarrolladas para la mejora de eficiencia energética en edificios, destacamos técnicas para la detección de anomalías, modelado de consumo energético, creación de perfiles de consumo o predicción de la demanda energética. Aunque todas ellas han demostrado buenos resultados en la gestión eficiente de energía, en este trabajo, nos centramos en el estudio de técnicas de modelado de consumo energético. En concreto, nuestra hipótesis inicial persigue el desarrollo de un método para encontrar de forma automática las relaciones entre los datos de consumo energético, y de esta manera poder proporcionar un modelo preciso e interpretable que explique dicho consumo y que sirva de ayuda para los analistas y CEOs en la toma de decisiones.

Para encontrar relaciones en el consumo energético, hemos de ser conscientes de que estas pueden no ser lineales, y que por tanto, las técnicas clásicas como regresiones lineales o métodos de *Box-Jenkins* no pueden ser utilizadas. En su lugar, haremos uso de Regresión Simbólica y Programación Genética. Como discutimos en la Sección 1.3, uno de los aspectos a considerar a la hora de implementar un algoritmo de Programación Genética, es el diseño de la estructura de datos que permitirá representar a los individuos. En este artículo, profundizamos en el uso de diferentes estructuras de datos para codificar expresiones algebraicas en PG: árbol de expresión, *Linear Genetic Programming* (LGP) y SLP, con el objetivo de validar el potencial de esta última en la exploración del espacio de soluciones y resolver el problema de modelado de consumo energético. Además, construimos un algoritmo de PG híbrido, que incluye una herramienta de optimización por mínimos cuadrados para la estimación de parámetros, con el objetivo de mejorar la exploración del espacio de soluciones. Finalmente, se ha llevado a cabo una experimentación sobre datos reales de consumo energético, para analizar el potencial de SLP sobre representaciones tradicionales. La publicación asociada a este estudio es:

- R.Rueda, M.P.Cuéllar, M.C.Pegalajar, M.Delgado (2019). Straight line programs for energy consumption modelling. Applied Soft Computing, 80, 310-328. DOI: 10.1016/j.asoc.2019.04.001.

1.5.2. El problema de la interpretabilidad

El uso de técnicas alternativas a modelos de cajas negras han cobrado especial interés recientemente, ya que el desarrollo de modelos interpretables a la vez que precisos ofrecen una mayor confianza al usuario final, facilitando el uso de la IA como herramienta de soporte en la toma de decisiones. En el problema de la eficiencia energética, Regresión Simbólica puede ser utilizada para encontrar un modelo que explique el consumo energético en términos de una expresión algebraica interpretable.

Tradicionalmente, el problema de RS ha sido resuelto por medio de algoritmos de PG, como describimos en la Sección 1.5.1. Sin embargo, la interpretabilidad de las soluciones encontradas pueden verse afectadas, debido a problemas clásicos presentes en PG, como el *bloating*, que implica un crecimiento descontrolado de las soluciones sin mejorar la calidad de las mismas. Por ello, en esta investigación, proponemos el diseño de un Algoritmo de Colonias de Hormigas (ACO) para resolver el problema de RS. En nuestra propuesta, utilizamos la estructura SLP para codificar expresiones algebraicas, de modo que podemos utilizar su estructura lineal de tipo grafo para abordar el problema como un problema de recorrido de grafos, en el que se pretende encontrar el SLP de menor tamaño que minimice una medida de error.

Por último, en esta investigación abordamos el problema del modelado de consumo energético, haciendo uso de datos de consumo procedentes de las facultades de Ciencias y Psicología, y los centros de investigación CITIC (Centro de Investigación en Tecnologías de la Información y de las Comunicaciones) y Mente, Cerebro y Comportamiento de la Universidad de Granada. Para validar nuestra propuesta, realizamos una comparativa con un algoritmo Ant Colony Optimization (ACO) clásico que utiliza la representación de tipo árbol y con el algoritmo de Programación Genética descrito en la sección anterior. La publicación asociada a este trabajo es:

- R.Rueda, L.G.B.Ruiz, M.P.Cuéllar, M.C.Pegalajar (2020). An Ant Colony approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling. International Journal of Approximate Reasoning, 121, 23-38.
  DOI: 10.1016/j.ijar.2020.03.005.

1.5.3. Predicción de series temporales multivariantes

Los algoritmos de predicción son una de las técnicas más utilizadas en ML e IA para resolver problemas como el de la presente tesis. Tradicionalmente, la predicción de consumo energético en edificios se ha llevado a cabo haciendo uso de una serie temporal que recoge el consumo histórico de un edificio, y es utilizada como entrada de modelos predictivos, como ARIMA o redes neuronales, habiendo demostrado un alto potencial. La Universidad de Granada se compone por un total de 25 centros docentes, distribuidos en diferentes zonas geográficas. Por tanto, para predecir el consumo energético de cada edificio, tradicionalmente se construye un modelo predictivo para cada uno de ellos. Sin embargo, en esta investigación, nuestra hipótesis de partida es que si las series temporales de consumo de un conjunto de edificios muestran una correlación media/alta, entonces podremos diseñar un único modelo capaz de aprender su comportamiento común, y construir un modelo de predicción general capaz de aprender las relaciones entre todas las series temporales, y parametrizar dicho modelo para adaptarlo a cada caso específico, obteniendo un modelo de predicción preciso que explique el consumo global del conjunto de edificios. Es decir, queremos modelar, si existe, el comportamiento común en el consumo energético de varios edificios, y expresarlo en términos de una fórmula matemática. Por ejemplo, supongamos que las Facultades de Psicología e Informática muestran un consumo energético correlacionado, y que puede ser expresado en términos de una expresión algebraica como $d_5 = d_1 * log(d_3)$, donde d_1, d_3 y d_5 son los días de la semana lunes, miércoles y viernes, respectivamente. La expresión algebraica anterior nos indica que el consumo del viernes viene dado por el consumo del lunes, multiplicado por logaritmo del consumo del miércoles. Esta ecuación puede ser utilizada para predecir el consumo de las facultades de Psicología e Informática, bastaría con aplicar dicha fórmula sobre los datos de consumo de ambas facultades,

respectivamente.

Para abordar este problema hacemos uso de RS, algoritmos genéticos y la representación SLP. En primer lugar, formulamos nuestro problema como un problema multi-objetivo, donde debemos encontrar una expresión algebraica que satisfaga la predicción del consumo de varios edificios (cada edificio es un objetivo), utilizando el algoritmo genético multi-objetivo NSGA-II (del inglés *Non-dominated Sorting Genetic Algorithm II*). En segundo lugar, formulamos el problema como un problema mono-objetivo, donde utilizamos el algoritmo genético desarrollado en el primer trabajo de esta tesis, con una pequeña modificación: el *fitness* utilizado para evaluar la calidad de un individuo se calcula como el promedio de los errores parciales de la estimación del consumo de cada edificio y su correspondiente consumo real. Para comprobar qué enfoque es mejor, diseñamos dos experimentos: uno sobre datos sintéticos generados por medio de funciones *benchmark*, y otro sobre datos de consumo energético de la UGR. La publicación asociada a este trabajo es:

- R.Rueda, M.P.Cuéllar, M.Molina-Solana, Y.Guo, M.C.Pegalajar (2019). Generalised Regression Hypothesis Induction for Energy Consumption Forecasting. Energies, 12, 1-22. DOI: 10.3390/en12061069.

1.5.4. Mejora de la exploración del espacio de búsqueda

Uno de los principales problemas de los algoritmos de PG durante el proceso de exploración del espacio de soluciones es el balance entre convergencia y divergencia. Mientras que una convergencia prematura provoca una caída en óptimos locales, una alta divergencia implica una reducción de la exploración del espacio de búsqueda. Como estudiamos al principio de esta tesis, el balance entre convergencia y divergencia puede ser logrado a través de los operadores de selección y recombinación. En la literatura podemos encontrar diferentes aproximaciones para abordar este problema, desde el desarrollo de nuevos operadores de cruce y mutación para preservar la diversidad en la población, hasta el diseño de estrategias de selección basadas en el uso de una distancia que determine cómo de diferentes son entre sí los individuos de la población,

y usarlo como criterio de selección para controlar la diversidad. Las medidas adoptadas por la comunidad investigadora para resolver este problema pueden ser clasificadas en tres categorías: Diversidad de fenotipo, que tiene en cuenta el valor del *fintess* de cada individuo. Diversidad de genotipo, que considera las diferencias estructurales entre los individuos y una combinación de ambas; Diversidad fenotipo y genotipo.

Atendiendo a la clasificación anterior, podemos encontrar diferentes criterios para controlar la diversidad de una población, desde el uso de medidas para cuantificar diferencias o similitudes entre individuos de la población (diversidad genotipo), hasta técnicas para usar información semántica como criterio de selección de individuos (diversidad fenotipo). En nuestra investigación, nos centramos en el desarrollo de una medida basada en la distancia de *Levenshtein* que nos ayude a cuantificar cómo de diferentes son dos SLPs. La distancia de *Levenshtein* se utiliza para determinar el número mínimo de operaciones necesarias para transformar una cadena de caracteres en otra. Las operaciones permitidas son la inserción, eliminación y sustitución de un carácter. Dicha distancia aplicada en nuestra investigación pretende obtener el número mínimo de operaciones requeridas para convertir un SLP en otro. De esta manera, proponemos utilizar esta medida para determinar la diversidad de una población de SLPs y alcanzar un equilibrio entre divergencia y convergencia. En concreto, en este trabajo demostramos que la medida desarrollada es una métrica y que puede ser utilizada en un algoritmo CHC (*Cross generational elitist selection Heterogeneous recombination Cataclysmic mutation algorithm*), para encontrar un balance entre exploración y explotación del espacio de búsqueda. Como no hemos encontrado ningún trabajo previo que utilice la mencionada métrica para cuantificar cómo de diferentes son dos SLPs, comparamos nuestra propuesta con métodos clásicos basados en árboles. En concreto, validamos nuestra propuesta sobre un conjunto de datos sintéticos y sobre datos reales de consumo energético procedentes de la UGR. La publicación asociada a este trabajo es:

- R.Rueda, M.P.Cuéllar, L.G.B.Ruiz, M.C.Pegalajar. A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming.

1.6. Resultados

En esta sección discutimos los resultados obtenidos en las publicaciones realizadas durante este proyecto de tesis. La Sección 1.6.1 muestra los resultados obtenidos con las diferentes representaciones utilizadas para codificar expresiones algebraicas en un algoritmo de PG para resolver el problema del modelado de consumo energético. Posteriormente, en la Sección 1.6.2 discutimos la importancia del tamaño de los resultados obtenidos en RS, mostrando como afecta en la interpretabilidad de las soluciones. Además, mostramos los beneficios de utilizar un algoritmo de colonia de hormigas frente algoritmos de PG. En la Sección 1.6.3 mostramos los resultados obtenidos con algoritmos mono- y multi-objetivo para resolver el problema de predicción de consumo energético procedente de varios edificios. Finalmente, la Sección 1.6.4 muestra el potencial de la métrica desarrollada para controlar la diversidad en una población de SLPs, realizando una comparativa con algunas de las métricas usadas en representaciones clásicas de tipo árbol.

1.6.1. Regresión Simbólica y el modelado de consumo

La representación utilizada para codificar individuos en PG, así como el diseño de sus operadores genéticos, ha generado un amplio debate en la comunidad investigadora, llegando a la conclusión de que éstos están estrechamente ligados con la capacidad de exploración del espacio de soluciones. Es por ello que la principal motivación de este trabajo versa sobre el estudio de la estructura más conveniente para codificar expresiones algebraicas, junto con el diseño de sus correspondientes operadores genéticos para resolver el problema de Regresión Simbólica. Para validar esta investigación, abordamos el problema del modelado del consumo energético de la Universidad de Granada.

Para analizar el potencial de la estructura SLP, hemos seleccionado dos representaciones alternativas para establecer una comparativa. En concreto, hemos utilizado la estructura no lineal árbol de expresión y la estructura lineal *Linear Genetic Programming* (LGP). Adicionalmente,

hemos implementado un algoritmo de PG híbrido, donde incluimos un procedimiento de estimación por mínimos cuadrados para estimar el valor de los parámetros $\bar{w}$ en un SLP, con el objetivo de comprobar si dicha hibridación aumenta la capacidad de exploración del espacio de soluciones, evitando caer en óptimos locales. A dicha implementación, la denominamos SLP-GA.

De este estudio surge un análisis relacionado con la capacidad de cada representación en la exploración del espacio de búsqueda en PG, su habilidad para encontrar la expresión algebraica de menor tamaño y el tiempo computacional requerido para el entrenamiento cada aproximación. Comenzando por la capacidad de exploración, utilizamos el Error Cuadrático Medio (ECM) como criterio para determinar la calidad de las soluciones encontradas. En este sentido, concluimos que SLP-GA, encuentra la mejor expresión algebraica capaz de minimizar el ECM, en un 75 % y un 80 % de los experimentos realizados, en comparación con las representaciones de tipo árbol y LGP, respectivamente. Cabe destacar que, SLP-GA ha realizado una mejor exploración del espacio de búsqueda a costa de un incremento del tiempo computacional.

Por último, además del buen ajuste obtenido con SLP-GA, queremos señalar que las expresiones algebraicas encontradas han demostrado tener un tamaño de expresión menor que las encontradas por árboles y LGP. Este hecho sugiere que el algoritmo implementado permite seleccionar las variables más importantes de forma automática, ya que expresiones de menor tamaño han sido capaz de modelar el consumo energético de forma más precisa. Esto sirve de motivación, ya que aunque el algoritmo no ha sido diseñado como técnica de selección de características, ha sido capaz no solo de encontrar mejores expresiones algebraicas en términos de precisión (en comparación con árboles y LGP), sino también en términos de selección de características, permitiendo ser una herramienta útil para la ayuda de toma de decisiones en el problema de la gestión de eficiencia energética.

1.6.2. El problema de la interpretabilidad

Tras verificar que la estructura SLP tiene potencial sobre estructuras clásicas de tipo árbol o estructuras lineales alternativas, centramos nuestros esfuerzos en mejorar la interpretabilidad de nuestras soluciones, para ayudar al experto en la toma de decisiones de alto nivel. Aunque en el

primer trabajo concluimos que la estructura SLP era capaz de encontrar expresiones algebraicas de menor tamaño, seleccionando las variables más relevantes del problema, detectamos que su tamaño seguía siendo elevado. Con el propósito de reducir el tamaño de las expresiones algebraicas resultantes, diseñamos un algoritmo de optimización de colonias de hormigas (ACO), al que denominamos SLP-ACO. Esta técnica, a diferencia de los algoritmos de Programación Genética, trata de encontrar no solo una solución precisa, sino también la de menor tamaño.

Los ACO son un tipo de metaheurística que pertenece a los métodos de inteligencia de enjambre, y se utilizan en problemas de optimización donde la formulación del problema puede diseñarse como la búsqueda del camino más corto en el recorrido de un grafo. Para ello, ACO se basa en el comportamiento de las hormigas a la hora de buscar alimento y transportarlo hasta el hormiguero, siempre por el camino más corto. En nuestro problema a resolver, el homólogo al hormiguero es el símbolo inicial de la gramática SLP y todas las posibles combinaciones para construir un SLP son los caminos a explorar por las hormigas, seleccionando siempre el camino más corto y que a su vez encuentre una solución precisa.

Para comprobar el potencial de nuestra propuesta, realizamos una comparativa con un algoritmo de colonias de hormigas clásico denominado *Dynamic Ant Programming* (DAP) para resolver el problema de RS, utilizando árboles de expresión. Adicionalmente, establecimos una comparativa frente al algoritmo SLP-GA desarrollado en el trabajo anterior. Las conclusiones alcanzadas en este trabajo fueron que nuestra propuesta, en comparación con la aproximación DAP, demostró encontrar mejores soluciones en términos de precisión, demostrando el potencial de la estructura SLP frente a estructuras clásicas de tipo árbol. En relación a la comparativa con el algoritmo de PG, llegamos a la conclusión de que SLP-ACO es capaz de encontrar soluciones tan precisas como SLP-GA, con la ventaja de encontrar una expresión algebraica de menor tamaño en todos los casos.

1.6.3. Predicción de series temporales multivariantes

En este trabajo abordamos el problema de la predicción de consumo energético en edificios. En concreto, nos centramos en el caso particular de edificios distribuidos, donde un conjunto

de edificios de la misma institución se encuentran ubicados en zonas geográficas diferentes, tal y como ocurre con las instalaciones de la Universidad de Granada. En este caso particular, es interesante estudiar si el consumo del conjunto de edificios mantienen un comportamiento similar, hecho que podría ser de ayuda para el experto en la toma de decisiones. En este sentido, si encontramos similitudes en su consumo, podremos desarrollar un único modelo capaz de aprender dicho comportamiento común y utilizarlo como herramienta para predecir el consumo energético de cada edificio.

Para corroborar nuestra hipótesis inicial, realizamos un primer estudio para analizar si existe alguna relación en el consumo energético entre los diferentes edificios de la UGR. Tras este primer análisis, encontramos que el consumo energético de un conjunto de edificios mostraba una correlación media/alta. Motivados por este hecho, abordamos el problema de predicción utilizando RS y algoritmos généticos para encontrar una expresión algebraica general que explicara el comportamiento común entre las diferentes series de consumo procedentes de varios edificios, y usarla para predecir su consumo energético.

Desarrollamos dos algoritmos genéticos para este fin: un algoritmo multi-objetivo y un algoritmo mono-objetivo. La primera aproximación se basa en el algoritmo de optimización NSGA-II, mientras que el algoritmo mono-objetivo se diseñó igual que la aproximación SLP-GA, con la diferencia de que el valor *fitness* de cada individuo fue calculado como el promedio de errores de cada objetivo.

Para validar nuestra propuesta, establecimos una comparativa entre ambas aproximaciones; mono- y multi-objetivo sobre dos escenarios: primero sobre un conjunto de datos sintéticos y el segundo sobre datos reales de consumo energético de la UGR. Los resultados mostraron que la aproximación mono-objetivo tiene potencial sobre el algoritmo multi-objetivo, siendo capaz de encontrar mejores resultados en todos los casos. Adicionalmente, comprobamos que la calidad de los resultados obtenidos por el algoritmo multi-objetivo es inversamente proporcional al número de objetivos del problema, es decir, a mayor número de objetivos, menor calidad de soluciones. Por último, concluimos que el algoritmo multi-objetivo se ve afectado por sobreajuste, ya que mostró mejores resultados en experimentos sobre datos de entrenamiento y peores resultados

sobre datos de validación.

1.6.4. Mejora de la exploración del espacio de búsqueda

Con el objetivo de mejorar la exploración del espacio de búsqueda y encontrar un balance entre convergencia y divergencia, en este último trabajo desarrollamos una medida basada en la distancia de *Levenshtein*, para determinar cómo de diferentes son dos individuos representados por SLPs. Esta medida es utilizada en combinación con el algoritmo CHC para controlar la diversidad de la población, y evitar caer en óptimos locales a la hora de resolver el problema de RS. A dicha aproximación, la denominamos *SLP-CHC*. Para validar nuestra propuesta, hemos realizado una comparativa frente a métodos clásicos de Programación Genética basados en árboles. En concreto, hemos utilizado como algoritmos de línea base, dos algoritmos genéticos basados en nichos, que implementan la distancia de *Levenshtein* para calcular las distancias entre individuos representados con estructuras de tipo árbol. Adicionalmente, incluimos en la comparativa el algoritmo genético SLP-GA, para comprobar si la medida desarrollada para SLPs incrementa la diversidad de la población. Por otro lado, dicha comparativa se realizó sobre dos escenarios diferentes: el primero de ellos, sobre un conjunto de datos sintéticos generados a partir de funciones *benchmark*, y el segundo sobre datos de consumo energético procedentes de 4 edificios de la UGR.

En la comparativa con los algoritmos de línea base, teniendo en cuenta los resultados obtenidos en términos de *fitness*, concluimos que nuestra métrica propuesta en combinación con el algoritmo CHC ayudó a realizar una mejor exploración del espacio de búsqueda, alcanzando mejores resultados que los obtenidos por los algoritmos de línea base. Por otro lado, analizamos la diversidad de la población a lo largo de las generaciones del ciclo genético de SLP-CHC y SLP-GA. De este estudio, concluimos que nuestra propuesta SLP-CHC permite incrementar la diversidad de la población, y consecuentemente posibilita una mejor exploración del espacio de búsqueda.

1.7. Conclusiones y trabajo futuro

Esta sección reúne las conclusiones obtenidas a lo largo del trabajo desarrollado en la presente tesis doctoral. La sección concluye arrojando algunas posibles líneas de investigación que podrían ser continuación de este proyecto.

En esta tesis abordamos el problema de modelado y predicción de series temporales de consumo energético procedente de edificios de la UGR. Para ello, nos hemos centrado en el estudio de técnicas precisas e interpretables que sean de utilidad para el experto en la toma de decisiones. En concreto, hacemos uso de Regresión Simbólica para encontrar una fórmula matemática que explique cómo se produce el consumo energético.

En primer lugar, hemos estudiado cómo afecta la representación de individuos en la capacidad de exploración del espacio de búsqueda en un algoritmo de PG, centrándonos en diferentes tipos de representaciones, lineales y no lineales, para codificar expresiones algebraicas en el problema de RS. De este estudio concluimos que la representación SLP tiene potencial sobre estructuras clásicas de tipo árbol y LGP. Además, la hibridación del algoritmo de PG junto con la técnica de mínimos cuadrados para la estimación de parámetros, ha ayudado a realizar una mejor exploración del espacio de búsqueda, permitiendo encontrar las variables dependientes que definen un modelo de regresión más preciso. Este hecho podría ser de ayuda para el experto en problemas de toma de decisiones y detección de anomalías. Sin embargo, detectamos que el tamaño de las expresiones algebraicas resultantes podría ser reducido, siendo posible aumentar su interpretabilidad. Por ello, en el segundo objetivo de esta tesis plantemos el estudio de métodos alternativos a PG para encontrar soluciones más interpretables y precisas. De este trabajo, surgió el desarrollo de una técnica basada en ACO como alternativa a PG para resolver el problema de RS, demostrando su potencial para encontrar soluciones tan precisas como los algoritmos de PG, con la ventaja de encontrar modelos más interpretables.

En otro de los objetivos, describimos una nueva formulación para abordar el problema de predicción de series temporales de consumo procedentes de varios edificios. Partimos de la hipótesis de que si un conjunto de edificios mostraba patrones de consumo similar (es decir,

analizamos la existencia de una correlación media/alta en el consumo energético), entonces podríamos construir un modelo general capaz de encontrar las relaciones en el consumo, y expresarlo en términos de una expresión algebraica que podría ser utilizada para predecir el consumo energético del conjunto edificios. Los resultados mostraron que nuestra propuesta tenía el potencial de encontrar una única fórmula matemática capaz de predecir el consumo energético de hasta 5 facultades.

En el último objetivo de esta tesis tratamos de mejorar la exploración del espacio de soluciones de un algoritmo de PG, usando la representación SLP. Basándonos en la distancia de *Levenshtein*, desarrollamos una medida capaz de cuantificar cómo de diferentes son dos SLPs, y usarla en un algoritmo CHC. En este estudio llegamos a la conclusión de que la medida propuesta es una métrica y que en combinación con el algoritmo CHC, es capaz de realizar una exploración del espacio de búsqueda más exhaustiva que el algoritmo de PG desarrollado en el primer objetivo de esta tesis, encontrando soluciones más precisas en todos los casos.

De las conclusiones extraídas en esta tesis, podemos proponer nuevas y prometedoras líneas de investigación, entre las que destacamos:

- Definición de propiedades de los operadores aritméticos para dotar de información semántica a los algoritmos desarrollados. Con esta información permitiremos una mayor reducción del espacio de búsqueda, permitiendo identificar expresiones algebraicas sintácticamente diferentes pero semánticamente equivalentes.

- Análisis y prevención de eventos. Construir un sistema que combine las técnicas desarrolladas en esta tesis, junto con técnicas de análisis semántico para poder identificar eventos o alertas en relación al consumo energético, tales como la previsión de picos de demanda.

- Uno de los principales inconvenientes de los métodos desarrollados en esta tesis es el elevado tiempo computacional empleado para entrenar cada modelo. Por ello, sería interesante hacer uso de herramientas de computación de alto rendimiento en GPUs para la paralelización de los algoritmos desarrollados y reducir su coste computacional.

Conclusions and future work

This section summarises the conclusions obtained throughout the work developed in this doctoral thesis, and concludes with some future works.

In this thesis, we address the problem of modelling and forecasting energy consumption time series, applied to the UGR buildings. To do so, we studied accurate and interpretable techniques that are useful for the expert in decision-making problems. More specifically, we use Symbolic Regression (SR) to find a mathematical formula that explains the dynamics of energy consumption time series.

Firstly, we studied how the solution representation affects the ability to explore the search space in a Genetic Programming (GP) algorithm, focusing on different types of representations, i.e. linear and non-linear, to encode algebraic expressions in Symbolic Regression problems. From this study we concluded that the SLP representation has potential over classical tree and LGP structures. Besides, the hybridization of the GP algorithm with the local search for parameter estimation helps make a better exploration of the search space, allowing the dependent variables that define a more precise regression model to be found. This fact could be helpful for the expert in both decision-making and anomaly detection problems. However, the size of the resulting algebraic expressions is high, and they could be reduced, being possible to increase their interpretability. Therefore, in the second goal of this thesis we proposed the study of alternative methods to GP in order to find not only more interpretable but also accurate solutions. From this work, we developed a technique based on ACO to solve the problem of SR, proving its potential to find as accurate solutions as the provided by GP algorithms, with the advantage of finding more interpretable models.

Thirdly, we described a new formulation to address the problem of energy consumption time series forecasting from several buildings. We assumed that if a set of buildings shown similar patterns in their consumption (i.e. we analyzed the existence of a medium/high correlation in energy consumption), then we could build a general model able to find the relationships in their consumption, and explain it in terms of an algebraic expression that could be used to predict

the energy consumption of the set of buildings. The results show that our proposal was able to find a single mathematical formula able to predict the energy consumption of up to 5 faculties in the dataset used for experimentation.

In the last objective of this PhD, we attempted to improve the exploration of the search space in a GP algorithm, using the SLP representation. Based on the *Levenshtein* distance, we developed a measure able to determine how different are two SLPs, and used it in a CHC algorithm. In this study we concluded that the proposed measure is a metric and that in combination with the CHC algorithm, it is capable of performing a better exploration of the search space than the GP algorithm developed in the first objective of this thesis, finding more accurate solutions in all cases.

From the conclusions drawn in this thesis, we can propose new and promising lines of research, among which we highlight:

- Definition of arithmetic operators properties to provide semantic information to the developed algorithms. With this information we will allow a greater reduction of the search space, identifying syntactically different but semantically equivalent algebraic expressions.
- Analysis and prevention of events. To build a system that combines the developed techniques, together with semantic analysis techniques to be able to identify events or alerts in relation to energy consumption, such as forecasting peak demand.
- One of the main drawbacks of the developed methods in this thesis is the high time needed to train each model. Therefore, it would be interesting to make use of high performance computing tools, such as GPUs, to develop parallel algorithms that help to reduce their computational cost.

Capítulo 2

PUBLICACIONES

2.1. Straight Line Programs for Energy Consumption Modelling

- **Referencia**: R.Rueda, M.P.Cuéllar, M.C.Pegalajar, M.Delgado (2019). Straight Line Programs for Energy Consumption Modelling. Applied Soft Computing, 80, 310-328
- **Estado**: Publicado
- **Factor de impacto** (JCR 2019): 5.472
- **Categoría**: Posición 20/136 en el área "Computer Science, Artificial Intelligence". Q1
- **DOI**: 10.1016/j.asoc.2019.04.001
- **Revista/Editorial**: Applied Soft Computing / Elsevier

Straight Line Programs for Energy Consumption Modelling

R.Rueda, M.P.Cuéllar, M.C.Pegalajar, M.Delgado

Department of Computer Science and Artificial Intelligence, University of Granada, Spain.

C/. Pdta. Daniel Saucedo Aranda s.n., 18071 Granada (Spain)

Abstract

Energy consumption has increased in recent decades at a rate ranging from 1.5 % to 10 % per year in the developed world. As a consequence, several efforts have been made to model energy consumption in order to achieve a better use of energy and to minimize environmental impact. Open problems in this area range from energy consumption forecasting to user profile mining, energy source planning, to transportation, among others. To address these problems, it is important to have suitable tools to model energy consumption data series, so that the analysts and CEOs can have knowledge about the underlying properties of the power demand in order to make high-level decisions. In this paper, we focus on the problem of energy consumption modelling, and provide a solution from the perspective of symbolic regression. More specifically, we develop hybrid genetic programming algorithms to find the algebraic expression that best models daily energy consumption in public buildings at the University of Granada as a testbed, and compare the benefits of Straight Line Programs with the classic tree representation used in symbolic regression. Regarding algorithm design, the outcomes of our experimentation suggest that Straight Line Programs outperform other representation models in the symbolic regression problems studied, and also that the hybridation with local search methods can improve the quality of the resulting algebraic expression. On the other hand, with regards to energy consumption modelling, our approach empirically demonstrates that symbolic regression can be a powerful tool to find underlying relationships between multivariate energy consumption data series.

Keywords: energy modelling, Symbolic Regression, Straight Line Programs

1 Introduction

Energy efficiency is gaining special interest in recent years due to the remarkable increase in energy consumption that has been happening for decades [Cas+15]. As it is shown in [Yu+10], energy consumption in buildings has increased by between 1.5 % and 1.9 % per year in Europe and North America from 1994 to 2004, 10 % per year during the past 20 years in China [Cai+09], and by 1.54 times in Iran [Sad+11]. The increase in the price of energy and the high demand from citizens and companies have encouraged governments to consider energy saving policies, trying to avoid irresponsible energy consumption and increase social welfare [ZK13][YZ11]. For all of the aforementioned reasons, researchers carry out several studies in order to reduce energy consumption and to use energy efficiently [PJP14].

If we focus on the case of residential and public buildings, nowadays we can set up a Building Automation System (BAS) [Sal05], and deploy multiple sensors to perform energy consumption monitoring, occupancy, lighting, temperature, etc., for online or a later offline analysis [Ekw+13]. While energy consumption forecasting is the problem that has been studied most [WS17][FB15], sensor-based technologies have provided the possibility of studying further applications of computer science in the area of energy efficiency research, such as anomaly detection [CT14][CW17a], energy consumption modelling [CFW01][AZN94], consumer profile mining [CPB17][Fig+05], systems control [Bal+13][Sha+13], or energy demand planning [GNC16], among others. The techniques used to solve each of these problems vary depending not only on the nature of the problem, but also on the requirements of the desired outputs. For instance, in the case of consumer profile mining, in [CPB17] a Fuzzy C-Means algorithm is used to classify consumer patterns assuming there are pre-selected clusters, while in [Fig+05] it is assumed that the consumer patterns are unknown in advance and the authors propose carrying out a cluster analysis prior to the consumer profile classification procedure. As another example, in reference [CT14] an anomaly detection system for energy consumption that works in real-time as prerequisite is developed, comparing different prediction methods such as neural networks or ARIMA for a later classification with K-Means. On the other hand, the work [CW17a] also addresses anomaly

detection but it focuses on data visualization and model selection to improve output information and assessment for facility managers.

The use of a BAS in a single building or a compound eases data collection and control over the automation systems with which a building is equipped, but it also enables the integration of energy consumption data with other information coming from external sources (climate, occupancy, etc.). Thus, we can use this new information to build more accurate prediction systems of energy consumption, or to include new knowledge in the system. As an example of both situations, we cite the study [Kho+16], in which the authors address the problem of energy consumption forecasting using neural networks considering exogenous input data such as the temperature, time of day, solar radiation, or wind speed, among others; or the work [Bal+13] that develops a control system that uses the WiFi network traffic in a building to calculate occupancy and then uses this new information to control the HVAC. However, using several information sources also implies an increase in the complexity of the monitoring system, due to the potential heterogeneity of the data [Kho+16] and it is necessary to use machine learning techniques in order to process and extract knowledge from large amount of data sources. We can find different studies focused on the use of machine learning to reduce or manage the energy consumption in buildings [Ber+10]. More specifically, the proposal in [ENP12] uses neural networks and support vector machines to predict the power consumption in residential buildings as a support for decision making.

Another aspect to be considered for a preliminary study of a problem regarding energy consumption analysis is to know if the input energy consumption data are either univariate or multivariate. Traditionally, forecasting methods used to predict energy consumption, and also time series in general, assume that the consumption data series is univariate and comes from a single source (i.e., the energy consumption sensor of a building or room, etc.), as for instance in [WS17][Kho+16], although they could use additional exogenous data as in [Kho+16]. However, other applications, such as energy consumption modelling problems, may consider energy consumption data from different sources as a single multivariate data series, where each dimension of the data could come from different energy consumption sensors. One example of this situation is the work in the reference [Rue+17a], where the authors tackle the problem of finding the

relationship between the energy consumption of similar buildings in the same compound.

Finally, another relevant aspect to be considered in the design of machine learning techniques for energy consumption analysis is the trade-off between accuracy and interpretability [BF16]. There are plenty of forecasting approaches with an average or high accuracy of prediction, such as, for instance neural networks [Kho+13], support vector machines or ensemble models [WS17]. In our opinion, these techniques are useful if their role is to be used as a black box system that takes input data from sources and provides output data that can be used for decision-making or as input for another system. The interpretability of the prediction model itself is not relevant in these types of applications. However, there are applications in which the interpretability of the model is a key requirement. One example is the work [Cia+14], where it is developed a system to model household energetic behaviour for high-level information gathering and modelling. The authors selected Mamdani fuzzy rules to model the energy consumption behaviour, so that the resulting models could be assessed by experts for a later analysis of energy plant sizing management.

After this short summary about energy efficiency, energy consumption analysis and related problems and their characteristics, we are ready to formulate the principal problem addressed in this study. In this manuscript, we tackle the problem of energy consumption modelling. Unlike forecasting, where the goal is to predict future values of energy consumption data, energy consumption modelling focuses on data mining and targets at developing models that can discover new knowledge, or explain the behaviour of energy consumption considering either univariate or multivariate energy consumption data series, plus additional exogenous data in some cases. References [Ang07][CFW01][AZN94][Sad09] are examples of previous approaches to energy consumption modelling. In [Ang07], the authors study the relationships between pollutant emissions and energy consumption in France, using co-integration and vector error-correction modelling techniques, and conclude about the high correlation of the studied variables. The reference [CFW01] proposes a vector autoregressive model (VAR) to model the relationship between energy consumption, employment and output for Taiwan, concluding that these three variables are co-integrated with one vector. Al-Garni et al. [AZN94] show how the energy consumption in Eastern Saudi Arabia could be modelled as a function of climate data, solar

radiation and population, using a regression model. Conversely, Perry Sadorsky [Sad09] shows some models of renewable energy consumption using *panel co-integration techniques* that explain how the economic growth of a country and the demand of energy creates opportunities for increasing the use of renewable energy.

To be more specific, the problem addressed in this work is the development of a method that can automatically find the inter-relationships between data in an energy consumption data series, and provide an accurate interpretable model that explains energy consumption considering these relationships. As these relationships might not be linear, and the problem is not targeted at forecasting but at knowledge discovery, classic time series analysis such as autocorrelation, linear regression or Box-Jenkins methodology [GH06] cannot answer both questions. Finding these relationships is a data mining problem that could not only provide information that explains the data series behaviour, but it could also be a powerful tool for an accurate estimation of energy consumption data. We tackle this problem from the perspective of symbolic regression (see Section 2), and the main contributions of this work are: The formulation of an energy consumption modelling problem from the perspective of the symbolic regression paradigm, for both data approximation and feature selection in energy consumption data; the proposal of a suitable representation model for symbolic regression, being compared experimentally with other classic models; and the methodology for data acquisition and treatment for energy consumption data modelling, including an algorithm with dynamic parameter settings estimation during the genetic algorithm iteration. Further applications such as time series prediction, anomaly detection, or higher-level decision making, might benefit from the outputs of our approach, as we suggest in Section 5.

In our experimentation, we provide a proof of concept of the proposed method applied to energy consumption data of public buildings at the University of Granada. The problem formulation is to know if the energy consumption of a working day can be explained over time with the energy consumption of the remaining working days in the same week and, if so, which days are related and how. The methodology that we propose to achieve our goal formulates the problem under the symbolic regression paradigm [MWB95], since symbolic regression can be applied to a dataset of numeric data series, it can find the relationships between dependent and independent

data, and it provides an algebraic expression as output which explains the relationships between dependent (output) and independent (input) data accurately. Despite the potential benefits of symbolic regression, classic genetic programming techniques [MWB95] have limitations and they return local optima easily. In this research work, we make a study of different representation techniques for symbolic regression and provide a hybrid algorithm to solve some limitations of the state of the art. Due to we are aware that the reader might not be familiar with related concepts about Symbolic Regression, Straight Line Programs and genetic programming, Section 2 provides an additional background in these topics so that the remaining of the article can be read fluently. After that, Section 3 explains how the problem of energy consumption modelling can be formulated as a symbolic regression problem, and the proposed search algorithm. Section 4 shows the experimentation and discusses the outcomes and limitations of the approach, and finally conclusions and future research work are shown in section 5.

2 Additional background

2.1 Fundamentals of Symbolic Regression

Regression analysis [Fra15] is a mathematical methodology used to fit a functional model between independent and dependent variables. In the literature, we can find that regression analysis is a widely used methodology in research for prediction [TY07] or data modelling [MPV12]. The components of regression analysis are: a function or model hypothesis $f(\bar{x}, \bar{w})$, a set of input data $\bar{x} = (x_1, x_2, ..., x_n)$, a set of output data $\bar{y} = (y_1, y_2, ..., y_m)$, and a set of constant parameters that depend on the model hypothesis f, named as $\bar{w} = (w_1, w_2, ..., w_k)$. The problem of regression analysis is to find the best values for the model parameters $\bar{w}$ so that $\bar{y} \approx f(\bar{x}, \bar{w})$. For this purpose, an error measurement function is minimized, such as, for instance $e(f, \bar{y}) = ||f(\bar{x}, \bar{w}) - \bar{y}||$, the sum of squared errors between the expected model hypothesis response $\bar{y}$, and its actual output $f(\bar{x}, \bar{w})$. The literature offers a wide variety of techniques for carrying out regression analysis, such as: linear regression [MPV12] or ordinary least squares

regression for polynomials [DG49]. Nevertheless, all these methods share the same limitation: the function f must be known in advance. If this function is unknown, sometimes it is difficult to provide a formulation for the model hypothesis f unless strong assumptions are assumed (linearity, logarithmic relationships, etc.), and, thus, it becomes necessary to find an alternative method to provide a model to approximate the desired dependent variables $\bar{y}$.

Despite these limitations, regression analysis has been successfully applied to the problem of energy consumption modelling. Just to cite a few research works, for instance the article [KR13] compares the use of regression analysis and neural network to predict energy demand in the residential sector in the USA. The article [BAB14] also uses regression analysis to predict the energy consumption in a supermarket. The study in [FB15] proposes simple and multiple linear regression analysis to predict residential energy consumption, and [AZN94] performs a study that relates climate data, solar radiation and population to model energy consumption.

All of these studies rely on an initial regression hypothesis f found by the researchers. However, when the hypothesis f is unknown or difficult to formulate, alternative techniques can be used to approximate this function, such as neural networks [PM05][BC07] or symbolic regression [MWB95][AB00]. Nevertheless, although neural networks are able to provide very accurate results to model the output data $\bar{y}$ [TAB16], their main limitation is the lack of interpretability, since they have been studied traditionally as black box models [EK99].

On the other hand, symbolic regression [BD02a] allows us to find a balance between the quality (accuracy) and the interpretability of the solution found. Symbolic regression generalizes the process of classical regression analysis, assuming that both the parameters $\bar{w}$ and the regression hypothesis f are unknown. Thus, the main goal of symbolic regression techniques is to find both f and $\bar{w}$ simultaneously, under the assumption that f is an algebraic expression. Symbolic regression attempts to build an approximation $\tilde{f}$ for the function f, making a combination of atomic operators that are known in advance, for instance: addition, subtraction, logarithm operators, etc, with the objective that $\tilde{f}(\bar{x}, \bar{w}) \approx f(\bar{x}, \bar{w})$. For this reason, symbolic regression uses an optimization algorithm to find the best candidate $\tilde{f}$ and $\bar{w}$ that minimize an error measure such as $||\bar{y} - \tilde{f}(\bar{x}, \bar{w})||$. Such an algorithm is usually a global search procedure such as a

genetic algorithm [MWB95], and the problem is known as *Genetic Programming (GP)*. Genetic programming [Lan98] is a supervised learning method based on genetic biological evolution. It uses a genetic algorithm to evolve a population of candidate algebraic expressions $\tilde{f}(\bar{x}, \bar{w})$, traditionally encoded into a binary tree.

Beyond the classic proposal of GP, the literature offers a wide selection of procedures to solve a genetic programming problem. For example, reference [MKJ12] describes how to include semantics into geometric semantic genetic programming (GSGP), to reduce the search space. Zhong et al. [ZOC16] explored a new representation to encode algebraic expressions as *main programs* with an additional set of automatically defined functions to encapsulate more complex operations. These approaches help us to overcome the *bloat* problem of genetic programming [PC13], and the second of these also helps to speed up convergence, obtaining accurate solutions in a more efficient way. Also, as traditional algebraic expression representation is a binary tree, the problem of symbolic regression has been formulated as a graph traverse problem, and it has been solved using ant colony optimization [BC02a]. In this case, both accuracy and the resulting expression size improve classic GP approaches, although the same limitations regarding the inability to prove global convergence of the algorithm remains, as it happens in all bio-inspired algorithms and metaheuristics [BLS13]. Further reading about genetic programming and algorithm proposals can be found in [Pol+07] and [VCS14].

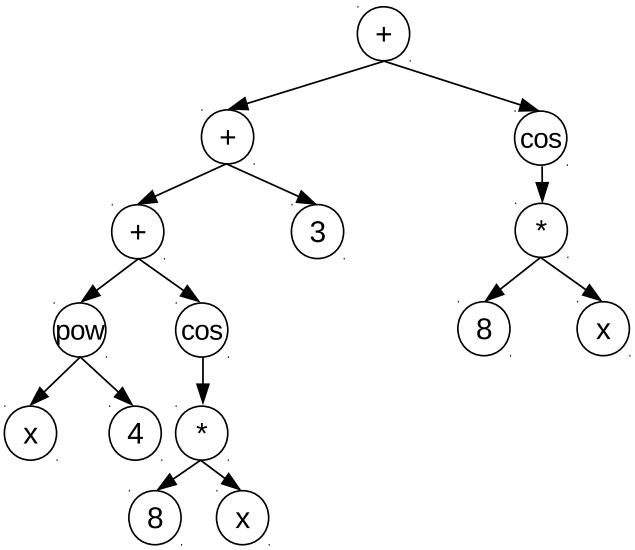


Figura 2.1: Tree structure representation of an algebraic expression

2.2 Symbolic Regression Representation

One of the main decisions to consider when solving a problem with metaheuristics is the selection of solution representation. The representation used may influence the size of the search space, and other components such as genetic recombination or mutation in genetic algorithms [McC05]. Therefore, it could have an impact in the final efficiency and effectiveness of the algorithm.

The literature provides different models of representation for the symbolic regression problem, although they can be classified into two main categories: graphs [Pol+07] and grammars [McK+10]. As was previously mentioned, the traditional representation used in GP is a binary tree that is consistent with a tree grammar, where the root and intermediate nodes are linked to operators that are applied over their child branches, and the leafs are data from $\bar{x}$ or parameters from $\bar{w}$. Figure 2.1 shows an example of the binary tree representation that encodes the algebraic expression $\tilde{f}(x, \bar{w}) = ((x^{w_1} + cos(w_2 * x)) + w_3) + cos(w_2 * x)$, where $\bar{w} = (w_1, w_2, w_3) = (4, 8, 3)$. Experimentally, the tree representation has some limitations. The most important of these is the bloat problem, as well as the difficulty of exploring the search space in order to find a replicate sub-expression in different branches of a single tree (see the subexpression $cos(8 * x)$ in Figure 2.1), the difficulty of comparing two different trees that represent the same algebraic expression, and also finding suitable genetic operators for recombination and mutation.

To overcome these limitations, other research efforts in GP have focused on the representation problem of the algebraic expression. For example, Miller et al. [MT00] represent a program using an indexed graph encoded as a linear string of integers. In [Li+08], Li et al. use an Instruction Matrix (IM) to evolve tree nodes and subtrees separately.

From all the existing representations in the literature about GP, in this article we focus on linear grammars. We believe that a linear representation could provide enough expressiveness to represent an algebraic expression without shrinking the search space, it could provide benefits for a better exploration of the search space, and it could also ease the design of simple, efficient and effective genetic operators in contrast to tree-based representations.

An example of linear representation is the Linear Genetic Programming (LGP) approach proposed in [BB10]. The structure used by LGP to encode algebraic expressions is based on a sequence of sentences that operate over a memory equipped with a set of registers. These sentences are mainly composed of an unary/binary operator that is applied over one/two input/s, and the result is stored into a register. The operation of a linear program is simple: The instructions are executed in a sequence from the first to the last sentence as a linear program, modifying the memory registers after the execution of each. The output is obtained from the calculations of the last sentence. In LGP, the search space is bounded by the maximum number of sentences allowed in a linear program.

Equation 2.1 shows an example of a linear program that encodes the same expression provided in Figure 2.1, where $r(i)$ stands for the value of the i-th memory register. Please note that the register values are modified during the program. As compared to the traditional tree representation, the linear genetic programming approach gives us the possibility to store calculations in memory and reuse this calculated information later in the program. Also, as the representation has a linear structure, it could improve evaluation, recombination and mutation time complexity in practice.

$$\begin{aligned} r(1) &:= pow(x, 4) \\ r(2) &:= 8 * x \\ r(2) &:= cos(r(2)) \\ r(3) &:= r(1) + r(2) \\ r(1) &:= r(3) + 3 \\ r(1) &:= r(1) + r(2) \end{aligned} \tag{2.1}$$

Another type of linear grammars are Straight Line Grammars (SLG) [BK13b]. SLGs are a type of non-recurrent grammars that allow the generation of a unique expression and can be as computationally powerful as free-context grammars. When applied to symbolic regression problems, SLGs are based on a set of production rules that generate an algebraic expression.

Straight Line Programs (SLP) [Alo+09] are based on Straight Line Grammars. They are a computational model that can be used for symbolic regression representation. As LGP, a SLP can be represented as a sequence of sentences indexed in a table, where each row represents a production rule of the SLG. Each of these production rules are of the form shown in equation 2.2, where U_i is the i-th entry in the table, $O_{U_i} \in \{o_1, o_2, ..., o_n\}$ is an arithmetic operator from a known set (for instance, the arithmetic operators $\{+, -, *, /, \log, \exp\}$), $T = \{t_1, t_2, ..., t_m\}$ is a set of terminal symbols (for instance, a parameter $w_a \in \{w_1, w_2, ..., w_k\}$ or an independent variable $x_b \in \{x_1, x_2, ..., x_n\}$), and $R_{U_i,1}, R_{U_i,2} \in \{T \cup \{U_1, U_2, ..., U_{i-1}\}\}$. If a SLP table contains N entries, then the output is provided by the evaluation of the N-th entry. Also, we remark that each non-terminal symbol appears on the left-hand side of the rule and can be converted into a terminal symbol or into the concatenation of two terminal or non-terminal symbols together with an operator symbol. We may observe that the rule references that can appear in the consequent must be references to previous rules, to prevent recursion. Thus, a SLP avoids cycles in the generation of algebraic expressions.

$$\begin{cases} U_i \rightarrow t_i \\ U_i \rightarrow R_{U_i,1} O_{U_i} R_{U_i,2} \end{cases} \tag{2.2}$$

Figure 2.2 shows an example of a SLP table with the algebraic expression encoded $\tilde{f}(x, \bar{w}) = ((x^{w_1} + cos(w_2 * x)) + w_3) + cos(w_2 * x)$, and its graph representation.

If we compare SLPs with tree-based representation, we may notice that SLPs are able to represent graphs instead of trees, therefore allowing us to reuse sub-expressions in the same algebraic formula, as happens with the term $cos(8 * x)$ in both Figure 2.1 and Figure 2.2. Although both approaches have the same power of expressiveness in symbolic regression problems, SLPs provide additional benefits, such as its linear structure in contrast to the non-linear tree representation. This eases not only the algebraic expressions evaluation, but also their evolution and the design of crossover and mutation operators in GP. Benefits of linear representations as compared to tree-based grammars were studied previously in [McK+10].

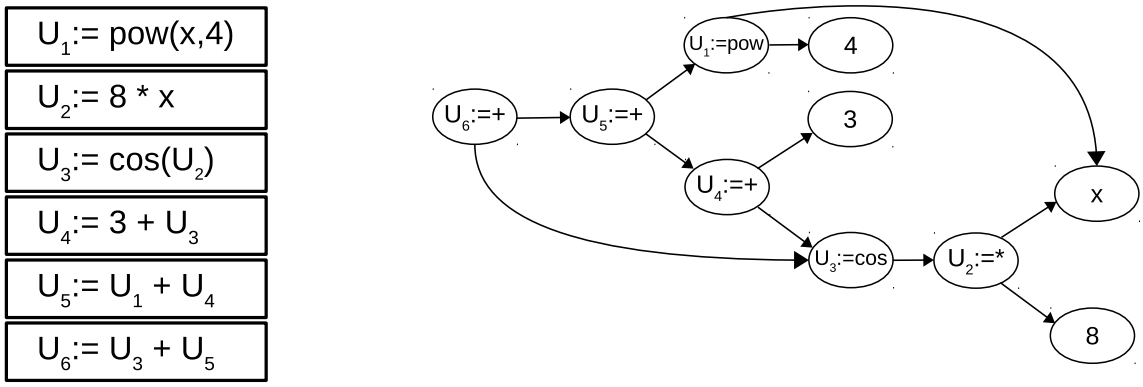


Figura 2.2: SLP Representation of an algebraic expression (left) and its non-cyclic directed graph associated (right)

On the other hand, if we compare LGP with SLP, we may notice that both representations also have the same power of expressiveness for symbolic regression problems since LGP and SLP are able to represent any algebraic expression. Moreover, both techniques assume the same underlying model to evaluate an algebraic expression, as a noncyclic directed graph. Although the computational representation of LGP and SLP are equivalent, the former, however, is a more general form of the latter with regards to representation. SLPs do not have memory registers to store partial computations. In our opinion, the generalization of LGP is a powerful tool that can be of great interest in problems such as automatic program generation, and in symbolic regression to store partial operations for a later reuse. However, in the case of symbolic regression, including register selection and assignments in every sentence produces a larger search space that could be more difficult to explore rather than using SLPs, with the consequence that the probability of finding the optimal solution decreases using LGP. Since SLP production rules may include references to previous calculations, the model does not need additional memory registers to store partial computations. Thus, we believe that this feature reduces the search space in SLPs.

Finally, if we look into applications of genetic programming and symbolic regression to energy efficiency and energy consumption problems, we may find some previous work in the literature, almost all focused on forecasting. Behera et al. [Beh+12] use a genetic algorithm in order to develop an effective planning system able to estimate demand and energy consumption. They

use tree representation to fit energy consumption for forecasting, and conclude that genetic programming provide more accurate results than classic regression analysis techniques. In the article [Huo+08], the authors propose gene expression algorithms to discover mathematical models to predict energy consumption using meteorological factors. In this case, the authors use a string linear model of fixed length to represent algebraic expressions. The work in [BAN02] uses LGP to perform consumer electricity demand forecasting. As compared to fuzzy systems and artificial neural networks, the authors suggest that the results obtained with LGP were the best when considering both accuracy and training time. Finally, the study in [Alo+09] proposed SLPs for symbolic regression, and the results obtained suggest that SLP may outperform classic representation limitations of the studied problem. Later, the work [Rue+17a] used Straight Line Programs to perform a preliminary study of SLP and tree representation performance for energy consumption modelling, to measure the contribution of a set of buildings to the overall amount of energy consumption in a compound. The authors conclude that using SLP could provide an improvement in training computational time due to their linear structure, which facilitates the reuse of algebraic subexpressions, and that SLP can also be used in future research works as a feature selection method. The literature offers different applications of SLPs, to represent algebraic operations [Ber84], solving geometry problems [Giu+98], polynomial equations [Kri02], or document clustering [SCS12].

In this article, we extend the preliminary tests performed in the conference paper [Rue+17a]. In this approach, our main hypothesis is that the use of SLPs cannot only be used to find a more accurate regression model than using other classic representations, but also that they can be used as a simultaneous feature selection method. In our research, we develop a hybrid genetic algorithm to train SLPs and simultaneously optimize the parameters of energy consumption models. Thus, feature selection is performed automatically by the algorithm itself, under the assumption that using the correct features will improve accuracy, while the inclusion of incorrect features will provide worse solutions that will be discarded during the evolutionary process. The experimental section shows in a real case study that the use of SLPs reduces the search space, and can provide more accurate regression functions in the datasets used. Moreover, the training procedure can automatically select the input variables that best model the output data

in the problems studied. Experimental analysis comparing SLPs with traditional tree-based representation and LGP give support to our hypothesis in energy consumption data modelling problems.

3 Methodology

3.1 Problem statement

As described in the introduction, energy consumption modelling is a general research topic that can be tackled in different ways, depending of the objectives pursued and the output requirements. In this piece of research, our input is an energy consumption data series measured in kW/h, coming from a BAS installed in a building. Our goal is to find inter-relationships between the daily energy consumption of the building, which help to approximate the energy consumption of working days in the same week. As an example, an output of the desired system could be interpreted by an analyst, CEO, or manager, as *"Wednesday's energy consumption is mainly related to that of Tuesdays and Fridays. The energy consumption of Mondays and Thursdays ommited to model Wednesday's. Moreover, if I need to approximate the energy consumption of Wednesdays, I can use the formula* $f(x_{Tuesdays}, x_{Fridays})$ *that the system provided"*, where $x_{Tuesdays}$ and $x_{Fridays}$ stand for the energy consumption data of those days in the same week.

Thus, the main purpose of this study is to find, if these exists, dependencies between the days of the week in order to model and estimate the energy consumption of another (different) working day. Assuming we name the working days as d_1, d_2, d_3, d_4, d_5, equation 2.3 shows that we want to approximate the energy consumption of day i considering the remaining days j_1, j_2, j_3, j_4, where $j_k \neq i \forall k$, and $\bar{w}$ and f are unknown.

$$d_i = f(d_{j_1}, d_{j_2}, d_{j_3}, d_{j_4}, \bar{w}) \tag{2.3}$$

The initial hypothesis in this study is that the energy consumption of all working days are

related. If so, then an algorithm to solve symbolic regression could be able to find both $\bar{w}$ and f. We also hypothesize that, if not all days are related, then the symbolic regression algorithm could select the best variables in $\{d_{j_1}, d_{j_2}, d_{j_3}, d_{j_4}\}$ to approximate d_i as accurately as possible, therefore performing a feature selection. We emphasize that any of the hypotheses assumed in this study does not mean causality, but correlation in data values.

To achieve our goals, the objective of the problem solution is to find the function f and parameters $\bar{w}$ that minimizes the error $||d_i - f(d_{j_1}, d_{j_2}, d_{j_3}, d_{j_4}, \bar{w})||$. The experimental results in this article empirically demonstrate that it is possible to find the best candidate regression hypothesis f that minimizes the approximation error, its parameters $\bar{w}$, and also the best subset of days $\{d_{j1}, d_{j2}, d_{j3}, d_{j4}\}$ to model the energy consumption for the i-th day in all the buildings studied. The hypothesis regarding the feature selection capability relies on the fact that using unrelated variables to find function f will provide algebraic expressions with a greater error rate than those solutions that use the correct working days.

From the problem statement described in the previous paragraph, we can design a system that fulfills all requirements regarding interpretability, numeric approximation and feature selection using symbolic regression. The following subsections describe the complete methodology followed in our research, from data acquisition to system design.

3.2 Data acquisition

We use a dataset containing the energy consumption of four buildings at the University of Granada, measured hourly from March 2013 to October 2015. To acquire the energy consumption data, each building is equipped with a set of sensors whose purpose is to monitor the energy consumption per hour (kW/h). A Building Automation System is responsible for monitoring the sensors, providing data from energy consumption of heating, ventilation, air conditioning and lighting systems in each building. The BAS stores the raw sensed data in a Database.

The raw data cannot be used directly in our experimentation due to misalignments in measurement times and missing data, so that we applied a preliminary preprocessing stage before

the experimentation. Thus, the first step in this preprocessing phase consists of seeking missing values (around 5 % of the data are missing) and interpolating each value. After that, a time alignment between sensor data measurements is necessary to obtain the data in the same temporal range. Each sensor data has a timestamp with a precision of an hour, so that we used these values to align the sensor data. The final step in the preprocessing stage is *Data aggregation*. Due to we want to model daily energy consumption, we calculated the total energy consumption for each day to obtain a daily energy consumption data series of each building. Finally, we organized these univariate data into a multivariate data series with 5 dimensions, each one for a working day. The results, for each building, were stored in tables with 5 columns (one for each working day), where each row in the table is the building energy consumption in a week, from Mondays (column 1) to Fridays (column 5). Figure 2.3 shows a scheme of the whole preprocessing stage from data acquisition to aggregation and the creation of the final dataset. This dataset is then used as input/output data for supervised learning of algebraic expressions using symbolic regression.

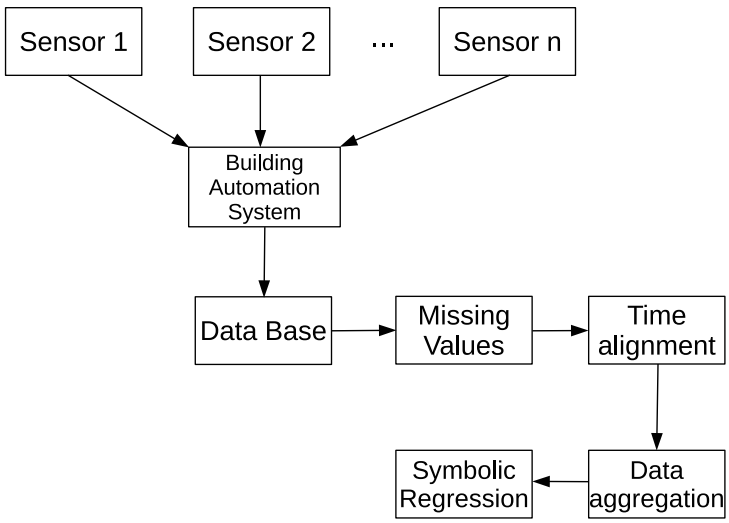


Figura 2.3: BAS system and data preprocessing

3.3 Straight Line Program representation and optimization algorithm

As we have previously described, we use SLPs to encode algebraic expressions, although our experimental study compares the outputs of SLPs with tree representation and LGP. We also emphasize that we have hybridized the genetic algorithm with a local search optimization procedure in order to achieve a better approximation of the parameters $\bar{w}$. Next, we describe the adaptation of SLP for symbolic regression, and the genetic crossover and mutation operators used. These operators were first proposed in [Alo+09].

Regarding SLP representation, equation 2.2 describes two rules to generate a table entry in a SLP. To solve the problem of symbolic regression, we can discard rule $U_i \rightarrow t_i$, which only generates a terminal symbol, since it increases the search space and it only provides trivial solutions. Moreover, we can also consider two cases for the rule $U_i \rightarrow R_{U_i,1} O_{U_i} R_{U_i,2}$, depending on the structure of the operator O_{U_i}. If the operator is binary, then $R_{U_i,1}$ will be considered as the first operand and $R_{U_i,2}$ as the second one. On the other hand, if the operator is unary, it will be applied to the first operand $R_{U_i,1}$, and the second operand will be omitted from the algebraic expression, and considered as to be empty (for the notation in following paragraphs, we use the symbol $\emptyset$ for the second operand when this situation occurs).

With these considerations, Algorithm 1 describes the main genetic algorithm (GA) used to train SLPs. Firstly, a population P is generated with a set of S_p random SLPs, where each SLP represents a candidate algebraic function $\tilde{f}(\bar{x}, \bar{w})$. Then, the tournament selection is used as the selection operator, to select the individuals that will be used as parents. After that, a crossover operator is applied with probability P_m from an uniform distribution $U(0, 1)$ to generate new offspring individuals C_1, C_2 from the recombination of the parents. Once the crossover has finished, a mutation operator is applied to the offspring. Then, the offspring are evaluated according to a fitness measure to be minimized, i.e. the Mean Square Error in our experimentation. Finally, the offspring replace population P for the next generation.

We would like to highlight that although classic implementations of genetic programming used a fixed value for the parameters $\bar{w}$, in this work we have constructed a hybrid algorithm that does not only find the best candidate algebraic expression $\tilde{f}(\bar{x}, \bar{w})$, but also the values of $\bar{w}$ simultaneously. More specifically, we propose a hybridation where a non-linear least-squares method [MK89] is applied during the evaluation of each candidate solution $\tilde{f}$, to approximate the parameters $\bar{w}$ that minimize the MSE. This operation is performed in the procedure *optimize* in Algorithm 1. With this hybridation, we ensure that the values w_i used in the candidate SLP are the optimal ones to ensure finding the best algebraic expression that can approximate output data.

Finally, we also emphasize that we have included an elitism component in the evolutionary genetic cycle. Then, when the replacement operator is applied to overwrite the initial population with the children, we ensure that the best individual found in the evolutionary process remains in the population. By doing so, we ensure that a potentially good candidate location in the solution space is not lost during the search process.

Since a GA is an almost standardized procedure that does not depend on solution representation, the specific components that must be designed to use GAs for SLP optimization in symbolic regression are: a) The generation of random SLPs; b) the recombination procedure; c) the mutation operator; and d) the evaluation process. Algorithm 2 explains the random generation of a SLP with size N, considering that the output algebraic expression is obtained from rule U_N. The operation of the algorithm is simple: We go through the rows of the SLP and, for each row, we randomly select the operator and the first operand. After that, if the operator is binary, the second operand is selected. Operators are randomly chosen from a fixed set of known operators O, while the operands may be a link to data of an input variable from $\bar{x}$, a link to a parameter from $\bar{w}$, or a reference to the calculations of a previous table entry in the SLP.

Once the SLP grammar is constructed, we can build the algebraic expression from the SLP starting at the N-th row of the table as the first production rule. The procedure to extract the algebraic expression from rule U_N consists in replacing each non-terminal symbol U_i with the consequent of their rule, iteratively from symbol U_N down to U_1, or until the expression

contains terminal symbols only. As an example, the algebraic expression of the SLP encoded in Figure 2.2 is obtained as follows: $f(\bar{x}, \bar{w}) = U_6$; $U_6 \Rightarrow U_3 + U_5 \Rightarrow U_3 + (U_1 + U_4) \Rightarrow U_3 + (U_1 + (w_3 + U_3)) \Rightarrow cos(U_2) + (U_1 + (w_3 + cos(U_2))) \Rightarrow cos(w_2 * x) + (U_1 + (w_3 + cos(w_2 * x))) \Rightarrow cos(w_2 * x) + (x^{w_1} + (w_3 + cos(w_2 * x)))$. Assuming the values for parameters $\bar{w} = (w_1, w_2, w_3) = (4, 8, 3)$ in the example of Figure 2.2, the evaluation of the expression for a concrete value of x would also require substituting the link to the parameters with their actual value, therefore providing the expression $f(\bar{x}) = cos(8 * x) + (x^4 + (3 + cos(8 * x)))$.

Regarding the crossover and mutation operators, in our study we used those proposed in [Alo+09]. However, to make this article self-contained, we describe the operators in the following paragraphs.

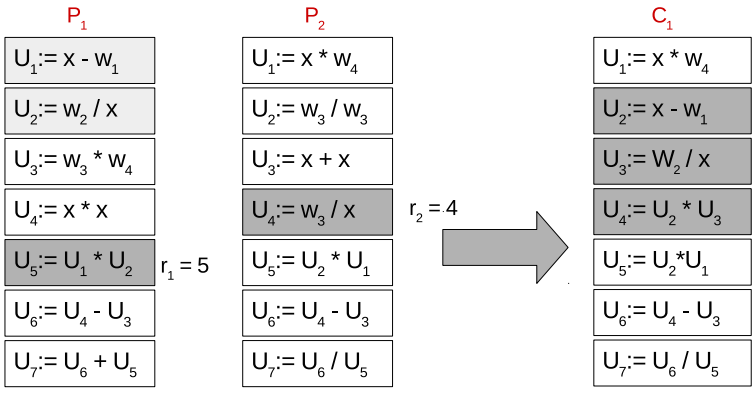


Figura 2.4: Example of crossover between parents P_1 and P_2

- **Crossover**. In Algorithm 1, two parents P_1 and P_2 are used in the *recombination* procedure to generate two children C_1, C_2. The crossover operates as follows: First, a random rule $U_i \in \{U_1, U_2, ..., U_{N-1}\}$ from P_1 is selected. Then, we build the ordered set of rules $R = \{U_i\} \cup \{U_j : U_i \stackrel{*}{\Rightarrow} U_j\}$, where $U_i \stackrel{*}{\Rightarrow} U_j$ means that the non-terminal symbol U_j can be reached from rule U_i with one or more production rules. The ordering criterion is performed considering the position of the entry of each rule in the SLP table. After that, a rule $U_k \in \{U_1, U_2, ..., U_{N-|R|+1}\}$ from P_2 is selected, where $|R|$ is the cardinal of R. Then the offspring C_1 is created as a copy of P_2, and the rules in R are copied into C_1 and renamed from $U_{k-|R|+1}$ to U_k. The offstring C_2 is generated with the same procedure, but exchanging the role of parents P_1 and P_2.

Figure 2.4 outlines an example of the crossover operation. In this example, rule $r_1 = 5$ is randomly selected from P_1. Then, the set R is calculated as $R = \{U_1, U_2, U_5\}$ since U_2 and U_1 can be derived from U_5. Please note that $R = \{U_i\}$ is ordered by index i. After that, since $|R| = 3$, we randomly select a production rule from P_2 from 1 to $N - |R| + 1 = 5$, assuming the size of the SLPs is $N = 7$. In this case, we select the random position $r_2 = 4$. Finally, the set R was copied into C_1 from position $r_2 - |R| + 1$ to position r_2 and renamed. Thus, rule $U_1 := x - w_1$ from P_1 is copied and renamed as rule $U_2 := x - w_1$, and rules U_2 and U_5 are copied and renamed as $U_3 := w_2/x$ and $U_4 := U_2 + U_3$, respectively, to preserve the information from parent P_1 in the child solution.

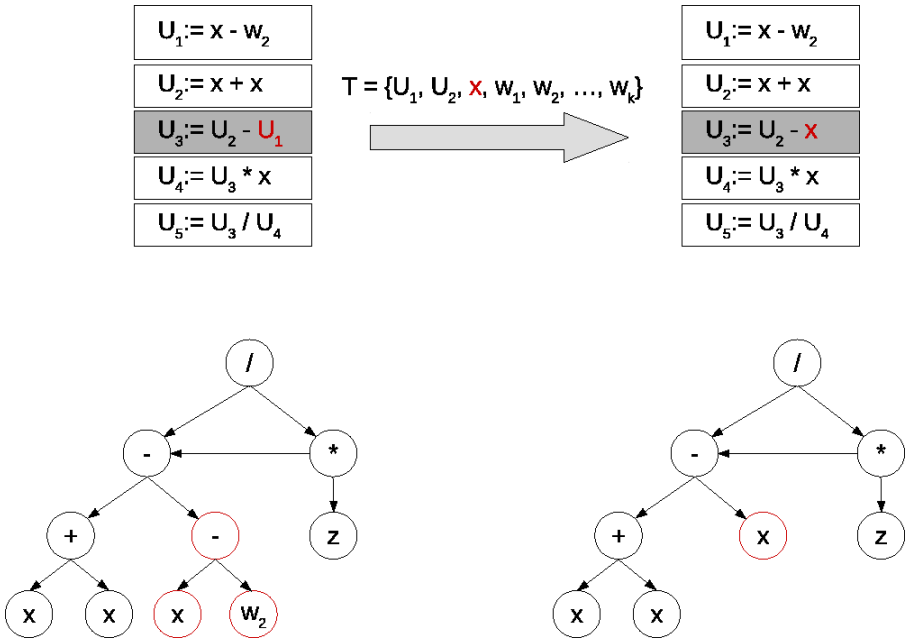


Figura 2.5: Example of mutation of a SLP

- **Mutation**. The mutation operator in Algorithm 1 selects a random element of the consequent of a random rule in the SLP table and exchanges it for another valid symbol (another operator if the selected symbol is an operator, or a terminal/link to a production rule if it is an operand). In the case of mutation of a binary operator to a unary operator, the second operand is left to the value $\emptyset$. On the other hand, for a mutation from a unary operator to a binary operator, then the second operand is randomly generated from the set of valid operands for the production rule that is affected by mutation. Figure 2.5 shows an example of the mutation operator.

Once we have described the components of the proposed algorithm to find algebraic expressions that help us to achieve the goals of our research, the next section describes the experimental settings and discusses the results obtained.

4 Experimentation

With this experimentation, we pursue to validate experimentally the hypotheses described the following questions:

1. Is it possible to model the energy consumption of a working day (target) considering the remaining days in the week (sources) using symbolic regression?

2. If so, is it possible to know which source days have influence to predict the energy consumption of the target day, and which ones do not influence in the model?

The answer to these two questions, formulated as a symbolic regression problem, would result in an algebraic expression where the energy consumption of a specific day (Monday, Tuesday, Wednesday, Thursday or Friday) can be obtained as a function whose inputs are the energy consumption of a subset of the remaining days. With regards to theoretical aspects, the experimentation tests how the hybridation of a genetic algorithm procedure with a local search method can help to obtain more accurate algebraic expressions, and also that linear representations such as SLPs may overcome traditional limitations of classic non-linear representations such as trees. On the other hand, regarding the energy consumption modelling problem addressed, this experimentation helps to prove experimentally that symbolic regression is a suitable method to perform both feature selection and algebraic expression search to provide an interpretable model of the energy consumption behaviour. This section is organized as follows: Firstly, subsection 4.1 introduces the dataset. After that, subsection 4.2 describes the experimental settings used, and finally subsection 4.3 show the experimental results and the discussion

4.1 Data description

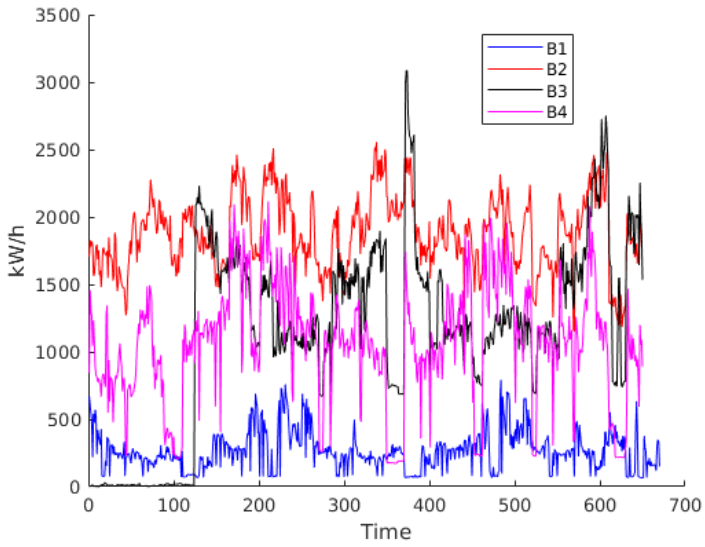
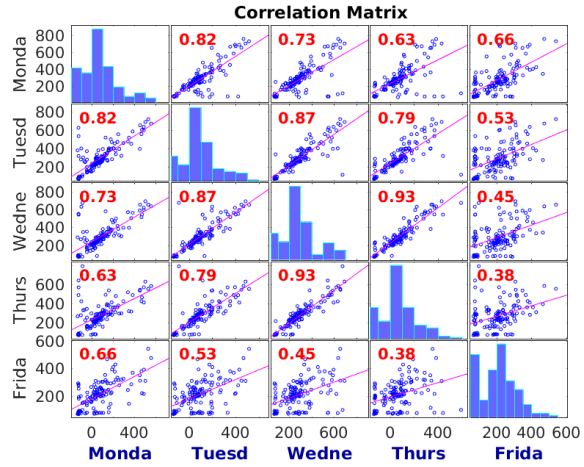


Figura 2.6: Energy consumption data series of buildings *B1, B2, B3, and B4*

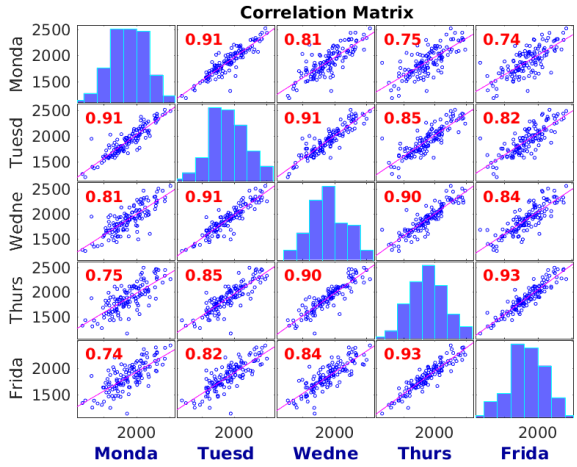
In this experimentation, we start from the dataset obtained after data acquisition and pre-processing explained in Section 3.2, for four buildings of the University of Granada equipped with a BAS. For confidenciality reasons, we are not allowed to provide the dataset, so that in this experimentation we name the buildings as buildings *B1, B2, B3, B4*. The buildings were selected with the criterion of having different energy consumption behaviour, so that the experimentation could be applied in different cases. The buildings are two research centers, a big faculty and a small faculty. The energy consumption of these buildings are shown in Figure 2.6, where the axis x stands for the time (in days) and the axis y for the energy consumption in kW/h, during a period of 130 weeks from Mondays to Fridays.

Since our initial assumption is that the energy consumption of a working day can be approximated using the energy consumption of the remaining working days in the same week, the first step in the experimentation is to know how working days are related each other, regarding the energy consumption. To that end, Figures 2.7a, 2.7b, 2.7c and 2.7d show the correlation plot matrices for each working day and building. The diagonal of the plot matrices shows the histogram, to know how the energy consumption is distributed for each working day. Finally, the text in red in the remaining correlation plots stands for the correlation coefficient R, ranging from -1 to 1,

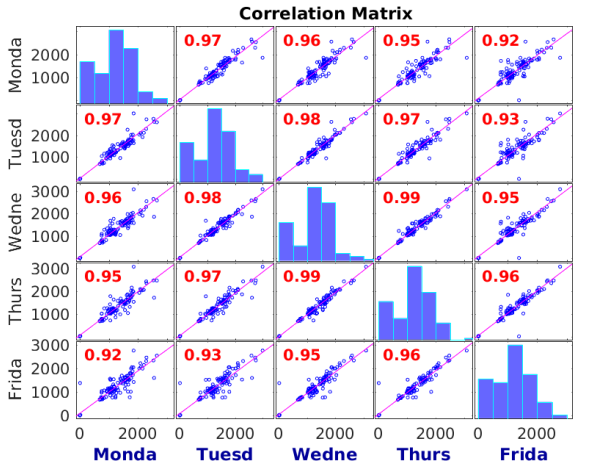
between the days of the corresponding row and column of the plot matrices.



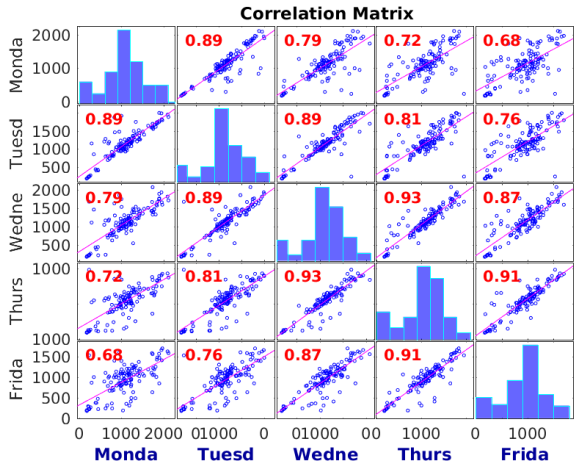
(a) Energy consumption correlation between working days for building *B1*



(b) Energy consumption correlation between working days for building *B2*



(c) Energy consumption correlation between working days for building *B3*



(d) Energy consumption correlation between working days for building *B4*

Figura 2.7: Correlation matrices of energy consumption for buildings B_1 to B_4, from Monday to Friday.

As it is expected, the information provided from Figures 2.7a, 2.7b, 2.7c and 2.7d asserts that there is a high ($R > 0{,}7$) or medium ($0{,}3 \leq R < 0{,}7$) correlation between the working days. This information cannot be interpreted in the way that the energy consumption of a working day must be approximated using the energy consumption of all the remaining working days, since many energy consumption data series could be related each other and provide the same partial information. Similarly, it also cannot be interpreted as that the energy consumption of a working day can be approximated using the data from a single (but different) working day, since we cannot ensure full dependence between two working days. Thus, the datasets contain

different variables that are partially related each other, and our goal is to find inter-relationships between the data and provide an interpretable model that uses the minimum number of variables necessary to provide an accurate estimation of the energy consumption. In the experimental results, we run the proposal and the algorithms GA and LGP with tree and linear program representation as baseline methods for comparison, to model energy consumption of all working days, for each building, considering the energy consumption coming from all the remaining working days as inputs. Then, we will obtain a model for each working day and building, together with the automatic feature selection performed by the algorithm.

4.2 Experimental settings

For the experimentation, we allowed 13 operators for the algebraic expression design, including the most usual operators such as $+, -, *, /, sin, cos, tan, log, exp, pow, sqrt, min, max$. We performed a preliminary extensive experimentation to tune the parameters of the genetic algorithm, considering both SLP and tree representation approaches. We also tried different number of parameters $\bar{w} = (w_1, w_2, ..., w_k)$ to be optimized, and we selected the value $k = 5$, since a lower value for k did not provide suitable results in the experimentation and a greater value increased the local search method to find $\bar{w}$ substantially with no significant improvements in accuracy. The parameters selected for the algorithms are shown in Table 2.1. The number of memory registers for LGP were obtained also after a trial-and-error procedure with tests ranging from 2 to 32 registers.

	Value
Number of samples	130
Population size	180
SLP size / Tree nodes	32
Crossover rate	0.9
Mutation rate	0.1
Function set F	$\{+, -, *, /, sin, cos, tan, log, exp, pow, sqrt, min, max\}$
Parameters w	$\{w_1, w_2, ..., w_5\}$
Dependent (input) variables	$\{d_1, d_2, d_3, d_4, d_5\}$ (excluding the target day)
Number of evaluations	40000

Tabla 2.1: Experimental Settings

It is necessary to emphasize that, during the experimentation, we have applied the optimization of parameters $\bar{w}$ of the resulting algebraic expressions within the genetic algorithm for SLPs in our proposal, but we also compare the approach with the traditional method, where $\bar{w}$ are known in advance and have fixed values. In these experiments, we established the values for the parameters $w_1 = 1, w_2 = 2, ..., w_i = i, ..., w_k = k$, since these values are widely used in the literature.

The algorithms included in the experimentation are described as follows:

- We use the acronym **SLP** as a reference to the proposed method. It implements the algorithm described in Section 3.3.
- We name **Classic SLP** as the genetic algorithm that trains SLPs with no parameter $\bar{w}$ optimization. Instead, $\bar{w}$ takes the values described in the previous paragraph.
- We use the term **Tree** to name the classic genetic programming approach, using tree representation and no parameter $\bar{w}$ optimization.
- The acronym **LGP** is used to name the linear program genetic approach, also with no parameter $\bar{w}$ optimization.

We run 30 executions for each algorithm, building and working day, so that we could perform a statistical analysis of the results. Each dataset was randomly divided into training (70 %) and test (30 %), to prevent overtraining, and we performed the analysis of results over the test set. Next section discusses the results obtained after this experimentation was performed.

4.3 Results and discussion

The results of the experimentation are organized in Tables 2.2, 2.3, 2.4 and 2.5, for buildings *B1*, *B2*, *B3*, and *B4*, respectively. Columns 2, 3, 4, 5 and 6 show the working days from Mondays to Fridays, and Column 1 describes the evaluated items. The rows in each table are organized in

groups of six, for each algorithm analyzed (*Tree*, *SLP*, *Classic SLP*, and *LGP*). In each group, we focus on the measurements *Average fitness*, which contains the average MSE of the 30 runnings of each algorithm; the *Best Fitness* and *Worst Fitness* with the minimum and maximum MSE obtained in the 30 runnings, respectively; the *average time* spent by the algorithm (in seconds) in the 30 runnings; the *expression size* with the average size of the algebraic expressions returned by the 30 runnings; and the *dependent variables*, which show the working days provided by the best solution found by each algorithm to estimate the energy consumption, using the notation introduced in Section 3.1. The size of an algebraic expression is calculated as the number of operators it contains, i.e. the number of non-leaf nodes in the tree representation, and the number of table entries and valid operations in SLP and LGP, respectively. Finally, the column labeled as *Parameter Estimation (s)* in the *SLP* algorithm contains the computational time (in seconds) used by the least squares estimation method during the search.

Moreover, for a better analysis of the results in Tables from 2.2 to 2.5, we have also included the boxplots of the MSE distribution in all experiments in Tables 2.8 to 2.11. Each table contains the results for a building from $B1$ to $B4$, respectively. Each picture in a table contains the boxplots of the MSE for the algorithms being compared, i.e. SLP, Classic SLP, LGP and Tree, for the same building and week day. Finally, a box plot of an algorithm in a picture must be interpreted as follows: A boxplot contains a visualization of the MSE distribution of the corresponding algorithm, using the quartiles of the MSE from Q1 to Q3, where Q2 (the median value) is highlighted with a red line. The whiskers plot the lowest MSE data of the error distribution still within 1.5 of the interquartile range (IQR) of the lower quartile, and the highest MSE data of the error distribution still within 1.5 IQR of the upper quartile. Finally, data highlighted with red symbol *'+'* represent all data from the MSE error distribution outside the limits of the IQR ranges described previously.

If we focus on the modelling accuracy of the resulting algebraic expressions of the algorithms being compared, according to the results in Tables from 2.2 to 2.5, and also from the boxplots in Tables 2.8 to 2.11, we can conclude that the proposed *SLP* algorithm has been able to find the solution with best fitness (minimum MSE) for all modelled working days and buildings. However, this algorithm was also able to provide the worst solution in 4 of the 20 case studies.

	Measures	Monday (d_1)	Tuesday (d_2)	Wednesday (d_3)	Thursday (d_4)	Friday (d_5)
		Building B_1				
Tree	Average Fitness	$5{,}74 * 10^3$	$9{,}86 * 10^3$	$6{,}09 * 10^3$	$3{,}9 * 10^3$	$5{,}86 * 10^3$
	Best Fitness	$4{,}15 * 10^3$	$6{,}7 * 10^3$	$3{,}9 * 10^3$	$3{,}75 * 10^3$	$4{,}8 * 10^3$
	Worst Fitness	$7{,}84 * 10^3$	$1{,}27 * 10^4$	$7{,}17 * 10^3$	$3{,}99 * 10^3$	$8{,}12 * 10^3$
	Average time	179.6	175.04	174.24	174.15	175.99
	Expression size	10.23	10.73	11.47	10.87	11.2
	Dependent variables	d_2, d_3, d_4, d_5	d_3, d_4, d_5	d_2, d_5	d_1, d_3	d_1, d_4
SLP	Average Fitness	$6{,}92 * 10^3$	$6{,}15 * 10^3$	$3{,}21 * 10^3$	$7{,}04 * 10^3$	$4{,}97 * 10^3$
	Best Fitness	$3{,}19 * 10^3$	$3{,}27 * 10^3$	$2{,}31 * 10^3$	$2{,}76 * 10^3$	$4{,}09 * 10^3$
	Worst Fitness	$4{,}04 * 10^4$	$1{,}26 * 10^4$	$6{,}43 * 10^3$	$1{,}06 * 10^5$	$6{,}13 * 10^3$
	Average time	8.08	7.87	7.77	7.75	7.78
	Parameter Estimation	892.88	935.85	909.03	1113.55	1041.82
	Expression size	10.2	10.4	10.23	10.33	10
	Dependent variables	d_2, d_3, d_4	d_1, d_4, d_5	d_1, d_2, d_5	d_2, d_3, d_5	d_1, d_2
Classic SLP	Average Fitness	$5{,}77 * 10^3$	$8{,}9 * 10^3$	$4{,}24 * 10^3$	$3{,}96 * 10^3$	$5{,}35 * 10^3$
	Best Fitness	$4{,}2 * 10^3$	$5{,}23 * 10^3$	$2{,}87 * 10^3$	$3{,}63 * 10^3$	$4{,}93 * 10^3$
	Worst Fitness	$8{,}34 * 10^3$	$1{,}2 * 10^4$	$6{,}91 * 10^3$	$4{,}4 * 10^3$	$7{,}19 * 10^3$
	Average time	8.09	7.87	7.77	7.75	7.78
	Expression size	9.37	11.46	10.6	10	10.23
	Dependent variables	d_2, d_3, d_4	d_1, d_3, d_5	d_1, d_4, d_5	d_2, d_3	d_1
LGP	Average Fitness	$8{,}18 * 10^3$	$1{,}27 * 10^4$	$4{,}17 * 10^3$	$4{,}38 * 10^3$	$8{,}39 * 10^3$
	Best Fitness	$7{,}28 * 10^3$	$1{,}22 * 10^4$	$3{,}47 * 10^3$	$3{,}88 * 10^3$	$6{,}74 * 10^3$
	Worst Fitness	$9{,}29 * 10^3$	$1{,}29 * 10^4$	$7{,}17 * 10^3$	$4{,}84 * 10^3$	$1{,}43 * 10^4$
	Average time	5.5	5.26	5.45	5.35	5.34
	Expression size	11.07	13.87	10.27	13.6	13.17
	Dependent variables	d_2, d_3, d_4	d_3, d_4	d_2	d_2	-

Tabla 2.2: Results for building B_1.

Considering the average fitness, *SLP* obtained the best results in 14 of the 20 case studies, while *Tree* and *Classic SLP* obtained the best scores in the remaining 5 and 1 cases, respectively. Finally, according to the boxplots, both *LGP* and *Classic SLP* algorithms performed with an intermediate average error in the problems studied.

We have performed statistical tests in order to verify these results. Since not all error distributions follow a normal distribution, a non-parametric Kruskal-Wallis test with 95 % of confidence level was applied to compare the results of *Tree*, *LGP*, *SLP* and *Classic SLP* representations. The test results are shown in Table 2.6, where Columns from 2 to 7 show the results of the test applied over pairs of algorithms, and Column 1 the target day to be modelled in the problem. The table rows are also organized by groups of 5, one group for each building from *B1* to *B4*. Finally, each cell contains the resulting p-value of the Kruskal-Wallis test. A value $< 0{,}05$ means that there are significant differences between the results of the compared algorithms in the corresponding dataset, and a value $> 0{,}05$ means that there are no significant differences in the results. In order to ease readability, we have also marked with symbol (+) if the left-hand

	Measures	Monday (d_1)	Tuesday (d_2)	Wednesday (d_3)	Thursday (d_4)	Friday (d_5)
		Building B_2				
Tree	Average Fitness	$1{,}05 * 10^4$	$7{,}47 * 10^3$	$8{,}77 * 10^3$	$5{,}82 * 10^3$	$9{,}03 * 10^3$
	Best Fitness	$9{,}52 * 10^3$	$7 * 10^3$	$8{,}12 * 10^3$	$5{,}7 * 10^3$	$8{,}74 * 10^3$
	Worst Fitness	$1{,}17 * 10^4$	$8{,}87 * 10^3$	$9{,}88 * 10^3$	$5{,}96 * 10^2$	$9{,}15 * 10^3$
	Average time	181.76	176.08	174.14	172.89	175.53
	Expression size	10.4	9.73	10.37	11.4	10.3
	Dependent variables	d_3	d_1, d_4, d_5	d_1, d_2, d_4	d_3, d_5	d_3, d_4
SLP	Average Fitness	$9{,}82 * 10^3$	$7{,}12 * 10^3$	$8{,}06 * 10^3$	$1{,}54 * 10^4$	$8{,}66 * 10^3$
	Best Fitness	$8{,}67 * 10^3$	$6{,}15 * 10^3$	$6{,}79 * 10^3$	$5{,}06 * 10^3$	$7{,}01 * 10^3$
	Worst Fitness	$1{,}12 * 10^4$	$9{,}29 * 10^3$	$1 * 10^4$	$2{,}93 * 10^5$	$1{,}07 * 10^4$
	Average time	7.88	7.89	7.95	7.85	7.87
	Parameter Estimation	1015.31	924.99	1044.45	1005.15	1095.13
	Expression size	9.47	11.37	9.37	10.53	10.63
	Dependent variables	d_2, d_3, d_4	d_1, d_4	d_1, d_2, d_4	d_2, d_3, d_5	d_1, d_2,d_3,d_4
Classic SLP	Average Fitness	$1{,}04 * 10^4$	$7{,}15 * 10^3$	$1{,}2 * 10^4$	$7{,}13 * 10^3$	$9{,}29 * 10^3$
	Best Fitness	$9{,}29 * 10^3$	$6{,}83 * 10^3$	$8{,}12 * 10^3$	$5{,}73 * 10^3$	$8{,}44 * 10^3$
	Worst Fitness	$1{,}21 * 10^4$	$9{,}29 * 10^3$	$2{,}03 * 10^4$	$1{,}17 * 10^4$	$1{,}19 * 10^4$
	Average time	7.88	7.89	7.95	7.85	7.87
	Expression size	11.7	11.1	10.97	10.03	11.23
	Dependent variables	d_3, d_4, d_5	d_3, d_5	d_1, d_5	d_2, d_3, d_5	d_4
LGP	Average Fitness	$1{,}09 * 10^4$	$1{,}04 * 10^4$	$1{,}82 * 10^4$	$1{,}11 * 10^4$	$9{,}4 * 10^3$
	Best Fitness	$1{,}03 * 10^4$	$7{,}17 * 10^3$	$1{,}01 * 10^4$	$9{,}02 * 10^3$	$8{,}97 * 10^3$
	Worst Fitness	$1{,}24 * 10^4$	$1{,}75 * 10^4$	$2{,}28 * 10^4$	$1{,}63 * 10^4$	$1{,}11 * 10^4$
	Average time	5.16	5.12	5.22	5.65	5.63
	Expression size	11.27	13.17	11.73	10.7	9.9
	Dependent variables	d_5	$d1$	d_1	-	d_1, d_4

Tabla 2.3: Results for building B_2.

side algorithm is better, with symbol $(-)$ if the best algorithm is in the right-hand side of the comparison, and with symbol (x) if both are equivalent.

Similar conclusions to the ones obtained from the preliminary analysis of results were thrown after the statistical test analysis was carried out. In this case, regarding the results of *Tree* and *SLP*, Table 2.6 shows that *SLP* obtained the best results in 15 of 20 experiments, equivalent in 2 problems and worse in the remaining 3. Then, with regards of the results of *SLP* and *LGP*, *SLP* achieved better results in 16 of 20 problems, worse solutions in 3 problems and it was equivalent in 1 problem. These results help us to conclude that the *SLP* proposal can help to improve the search of the best algebraic expression in most of the problems studied.

On the other hand, if we focus our attention in the possible benefits of the proposal of hybrid genetic programming with least square estimation of parameters $\bar{w}$, then we should compare *SLP* and *Classic SLP* both in performance and complexity. According to Tables 2.5 to 2.6, *SLP* performs better than *Classic SLP* in 15 of 20 problems, and they are equivalent in 1 problem.

	Measures	Monday (d_1)	Tuesday (d_2)	Wednesday (d_3)	Thursday (d_4)	Friday (d_5)
		Building B_3				
Tree	Average Fitness	$2{,}71 * 10^4$	$2{,}69 * 10^4$	$9{,}96 * 10^3$	$9{,}86 * 10^3$	$2{,}85 * 10^4$
	Best Fitness	$2{,}44 * 10^4$	$2{,}52 * 10^4$	$8{,}22 * 10^3$	$8{,}89 * 10^3$	$2{,}33 * 10^4$
	Worst Fitness	$3{,}59 * 10^4$	$3{,}41 * 10^4$	$1{,}24 * 10^4$	$1{,}87 * 10^4$	$3{,}04 * 10^4$
	Average time	178.55	178.11	173.67	174.27	172.41
	Expression size	10.4	9.23	11.3	11.3	9.533
	Dependent variables	d_1, d_3	d_1, d_4	d_2, d_4, d_5	d_1, d_3, d_5	d_1, d_2, d_4
SLP	Average Fitness	$2{,}25 * 10^{11}$	$2{,}47 * 10^4$	$9{,}14 * 10^3$	$1{,}02 * 10^4$	$2{,}32 * 10^4$
	Better Fitness	$1{,}86 * 10^4$	$2{,}06 * 10^4$	$5{,}45 * 10^3$	$6{,}89 * 10^3$	$2 * 10^4$
	Worse Fitness	$6{,}75 * 10^{12}$	$2{,}77 * 10^4$	$1{,}37 * 10^4$	$2{,}76 * 10^4$	$2{,}98 * 10^4$
	Average time	7.94	7.89	7.71	7.94	7.79
	Parameter Estimation	847.83	996.11	893.58	928.08	954.9
	Expression size	9.4	11.17	9.97	10.1	10.6
	Dependent variables	d_2, d_3, d_4	d_1, d_4	d_2, d_4, d_5	d_3	d_1, d_4
Classic SLP	Average Fitness	$2{,}91 * 10^4$	$2{,}53 * 10^4$	$1{,}21 * 10^4$	$1{,}13 * 10^4$	$2{,}52 * 10^4$
	Best Fitness	$2{,}39 * 10^4$	$2{,}47 * 10^4$	$7{,}22 * 10^3$	$8{,}01 * 10^3$	$2{,}23 * 10^4$
	Worst Fitness	$7{,}75 * 10^4$	$2{,}57 * 10^4$	$2{,}08 * 10^4$	$1{,}75 * 10^4$	$3{,}01 * 10^4$
	Average time	7.94	7.89	7.71	7.94	7.79
	Expression size	10.43	10.8	11.2	11.1	11.1
	Dependent variables	d_3	d_3, d_5	d_2, d_4, d_5	d_2, d_3	d_4
LGP	Average Fitness	$2{,}94 * 10^4$	$2{,}83 * 10^4$	$1{,}39 * 10^4$	$1{,}62 * 10^4$	$2{,}76 * 10^4$
	Best Fitness	$2{,}55 * 10^4$	$2{,}55 * 10^4$	$1{,}02 * 10^4$	$1{,}37 * 10^4$	$2{,}44 * 10^4$
	Worst Fitness	$3{,}63 * 10^4$	$4{,}18 * 10^4$	$1{,}52 * 10^4$	$3{,}53 * 10^4$	$3{,}63 * 10^4$
	Average time	5.31	5.17	5.26	5.34	5.26
	Expression size	12.33	7.07	7.7	11.27	8.67
	Dependent variables	d_3, d_5	-	d_2	d_1, d_2, d_3	d_4

Tabla 2.4: Results for building B_3.

However, time complexity increases in *SLP* in a rate of more than 100 times being compared to *Classic SLP*. Row *Parametter Estimation* verifies that the computational time for estimating $\bar{w}$ before any SLP evaluation is very computationally expensive. According to boxplots in Tables 2.8 to 2.11, this increase in time complexity could be worthy in some cases, when the *Classic SLP* gets trapped into local optima, since the optimization of $\bar{w}$ could improve the solution performance substantially.

If we compare *Classic SLP* with *Tree*, we also observe a suitable behaviour of the former method, since it provides better results in 8 of 20 problems, and they are equivalent in 9 problems. Thus, our experimentation results suggest that using linear models such as SLPs can help to overcome the problems regarding tree representation, which were known in advance in previous research works [McK+10][Alo+09][Rue+17a]. In addition, computational time decreases substantially from *Tree* to *Classic SLP*, since implementation of SLP crossover and mutation operators is much more simple than crossover and mutation operators over non-linear structures such as trees. Finally, if we compare *Classic SLP* with *LGP*, we may observe that *LGP* is the most

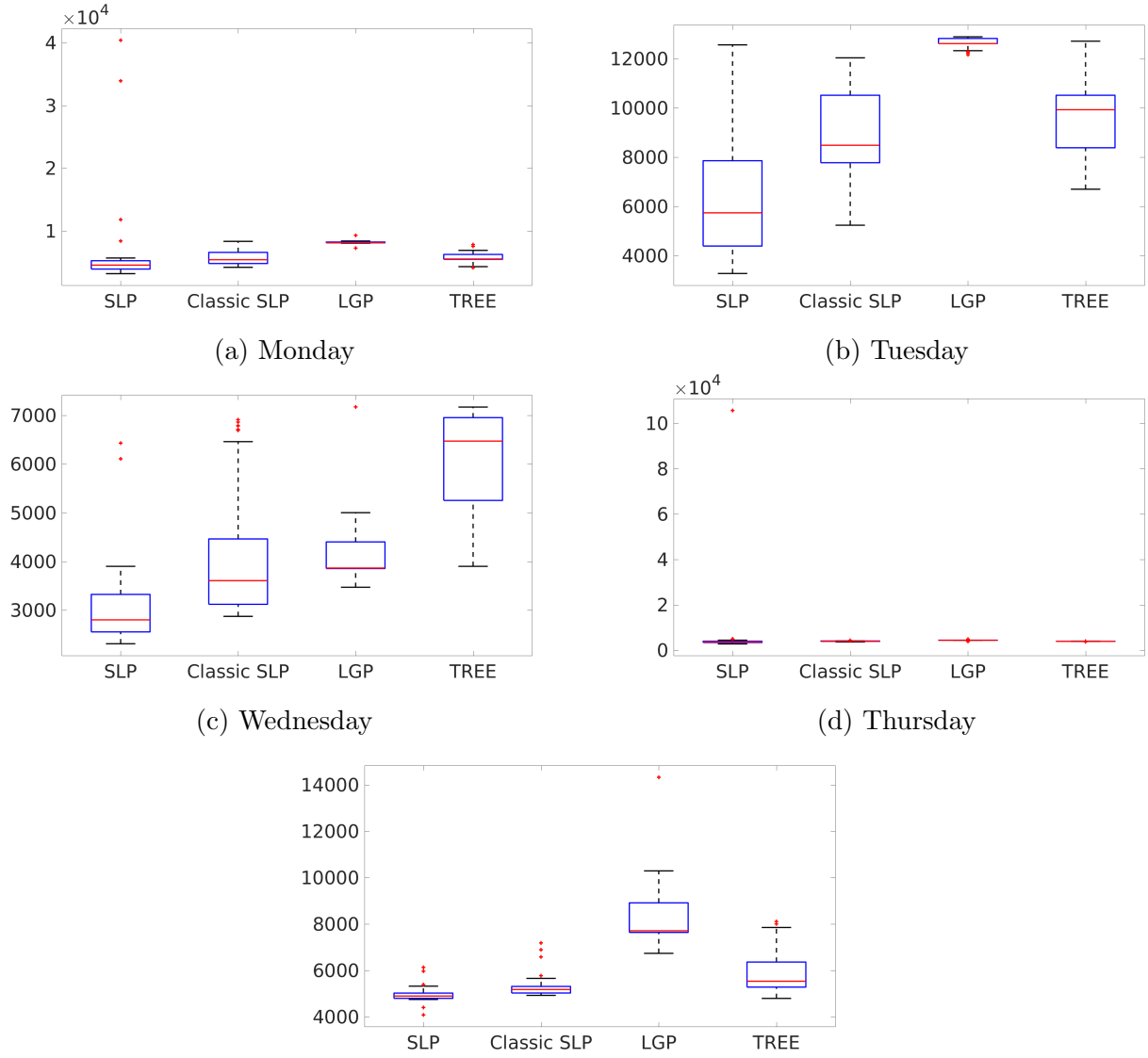
	Measures	Monday (d_1)	Tuesday (d_2)	Wednesday (d_3)	Thursday (d_4)	Friday (d_5)
		Building B_4				
Tree	Average Fitness	$4{,}17 * 10^4$	$6{,}32 * 10^4$	$2{,}37 * 10^4$	$1{,}74 * 10^4$	$4{,}17 * 10^4$
	Better Fitness	$4{,}04 * 10^4$	$4{,}2 * 10^4$	$1{,}35 * 10^4$	$1{,}32 * 10^4$	$4{,}04 * 10^4$
	Worse Fitness	$4{,}58 * 10^4$	$7{,}96 * 10^4$	$4{,}16 * 10^4$	$1{,}99 * 10^4$	$4{,}58 * 10^4$
	Average time	179.06	178.33	173.86	173.57	179.06
	Expression size	10.8	9.6	10.5	11.57	11.6
	Dependent variables	d_3, d_4	d_1, d_3	d_3, d_4, d_5	d_1, d_3, d_5	d_2, d_4
SLP	Average Fitness	$3{,}98 * 10^4$	$4{,}58 * 10^4$	$1{,}5 * 10^4$	$1{,}29 * 10^4$	$1{,}83 * 10^4$
	Better Fitness	$2{,}86 * 10^4$	$1{,}46 * 10^4$	$1{,}11 * 10^4$	$1{,}12 * 10^4$	$1{,}52 * 10^4$
	Worse Fitness	$4{,}32 * 10^4$	$7{,}34 * 10^4$	$4{,}49 * 10^4$	$1{,}7 * 10^4$	$2{,}29 * 10^4$
	Average time	7.78	7.72	7.67	7.8	7.56
	Parameter Estimation	1197.52	995.27	989.33	979.32	973.32
	Expression size	9.9	10.47	11.57	11.93	10.17
	Dependent variables	d_3, d_5	d_1, d_4	d_1, d_2, d_5	d_3, d_5	d_1, d_4
Classic SLP	Average Fitness	$1{,}56 * 10^5$	$5{,}67 * 10^4$	$1{,}76 * 10^4$	$1{,}65 * 10^4$	$1{,}82 * 10^4$
	Best Fitness	$4{,}02 * 10^4$	$4{,}18 * 10^4$	$1{,}4 * 10^4$	$1{,}32 * 10^4$	$1{,}77 * 10^4$
	Worst Fitness	$2{,}01 * 10^6$	$7{,}34 * 10^4$	$4{,}13 * 10^4$	$2{,}08 * 10^4$	$1{,}88 * 10^4$
	Average time	7.78	7.73	7.67	7.8	7.56
	Expression size	10.73	11.43	10.1	11.17	12
	Dependent variables	d_3	d_1, d_3	d_1, d_2, d_4, d_5	d_2, d_3	d_1, d_2, d_3, d_4
LGP	Average Fitness	$4{,}07 * 10^4$	$8{,}11 * 10^4$	$4{,}58 * 10^4$	$2 * 10^4$	$2{,}54 * 10^4$
	Best Fitness	$4{,}03 * 10^4$	$7{,}24 * 10^4$	$1{,}43 * 10^4$	$1{,}83 * 10^4$	$1{,}82 * 10^4$
	Worst Fitness	$4{,}21 * 10^4$	$9{,}49 * 10^4$	$8{,}95 * 10^4$	$2{,}07 * 10^4$	$7{,}9 * 10^4$
	Average time	5.25	5.35	5.4	5.4	5.35
	Expression size	14.57	11.83	8.83	11.57	12.67
	Dependent variables	d_3, d_4, d_5	-	-	d_2, d_3, d_5	d_1, d_3, d_4

Tabla 2.5: Results for building B_4.

efficient method regarding time complexity, although this method gets also trapped into local optima. We hypothesize that this behaviour could be due to the use of the memory registers in the representation to store the partial information, which make the search space of this algorithm larger than the search space of SLPs.

From this facts we may conclude that SLPs representation and the hybrid training algorithm used in this work are able to overcome local optima and provide better algebraic expressions than using classic genetic programming approaches with trees or LGP.

On the other hand, if we focus on the size of the expressions returned by the algorithms, we notice that *SLP* has returned the smallest algebraic expressions in 9 of 20 problems, while *Classic SLP*, *Tree* and *LGP* returned the smallest algebraic expressions in the remaining 4, 2 and 6 problems, respectively. This fact also suggests that using SLP representation helps to minimize the impact of the bloat problem in genetic programming, since 13 of the smallest algebraic expressions used this representation in the experiments. According to this, we may conclude



(a) Monday

(b) Tuesday

(c) Wednesday

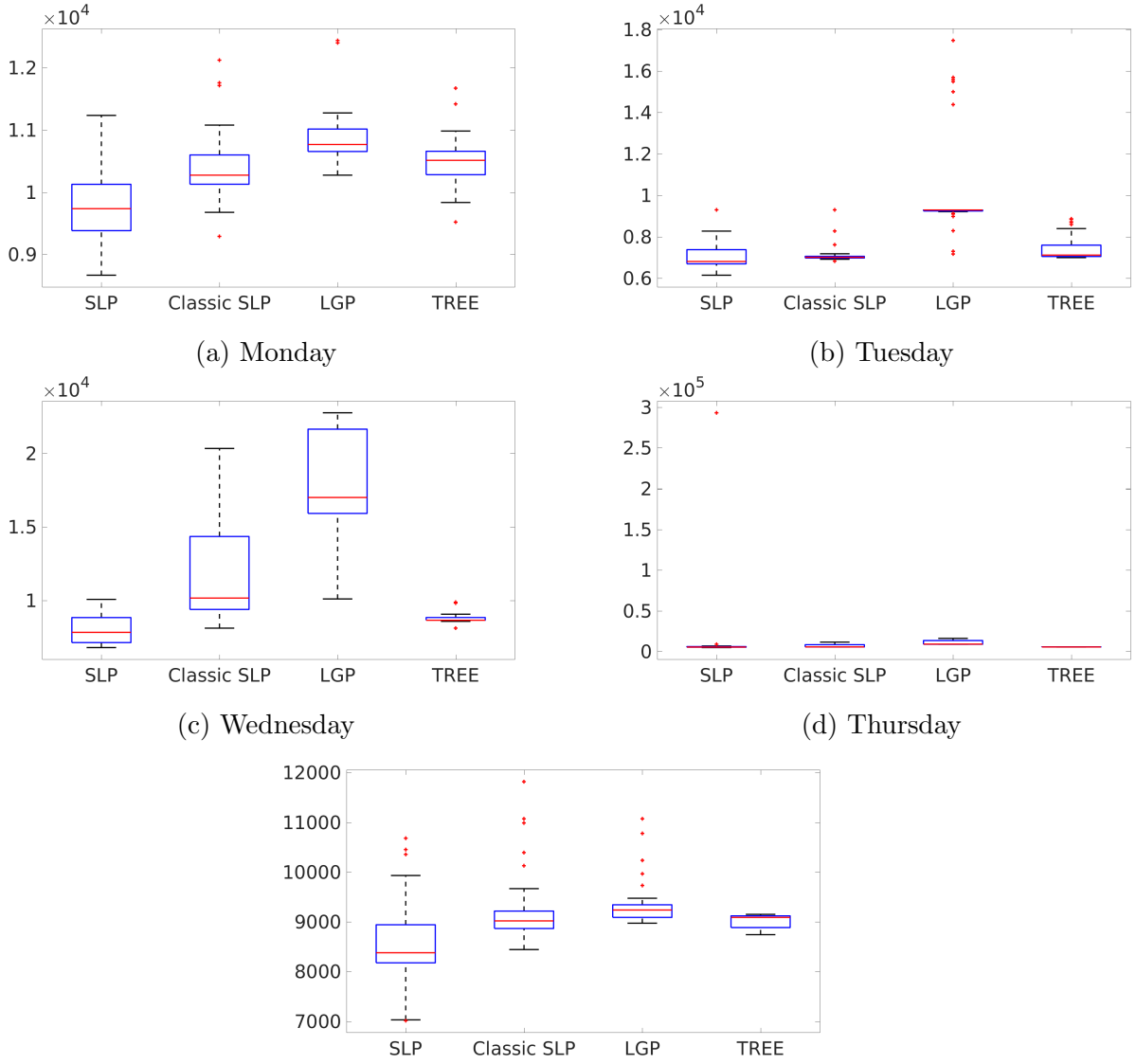
(d) Thursday

(e) Friday

Figura 2.8: Boxplots of the error rate (MSE) for building B1

that linear representations of algebraic expressions such as SLPs could not only improve the quality of the results obtained considering accuracy, but also to obtain simpler solutions.

Regarding the problem of feature selection, we can observe that both the SLP and tree algorithms are able to perform an automatic feature selection of the dependent variables simultaneously to the optimization process. However, we also may notice that *SLP* performs a better feature selection since it can overcome local optima better than the other algorithms, and therefore it is able to select the most appropriate inputs to provide the minimum MSE. If we compare the input working days selected to perform the energy consumption modelling with the correlation matrices in Figures from 2.7a to 2.7d, we can observe that the results obtained are consistent



(a) Monday

(b) Tuesday

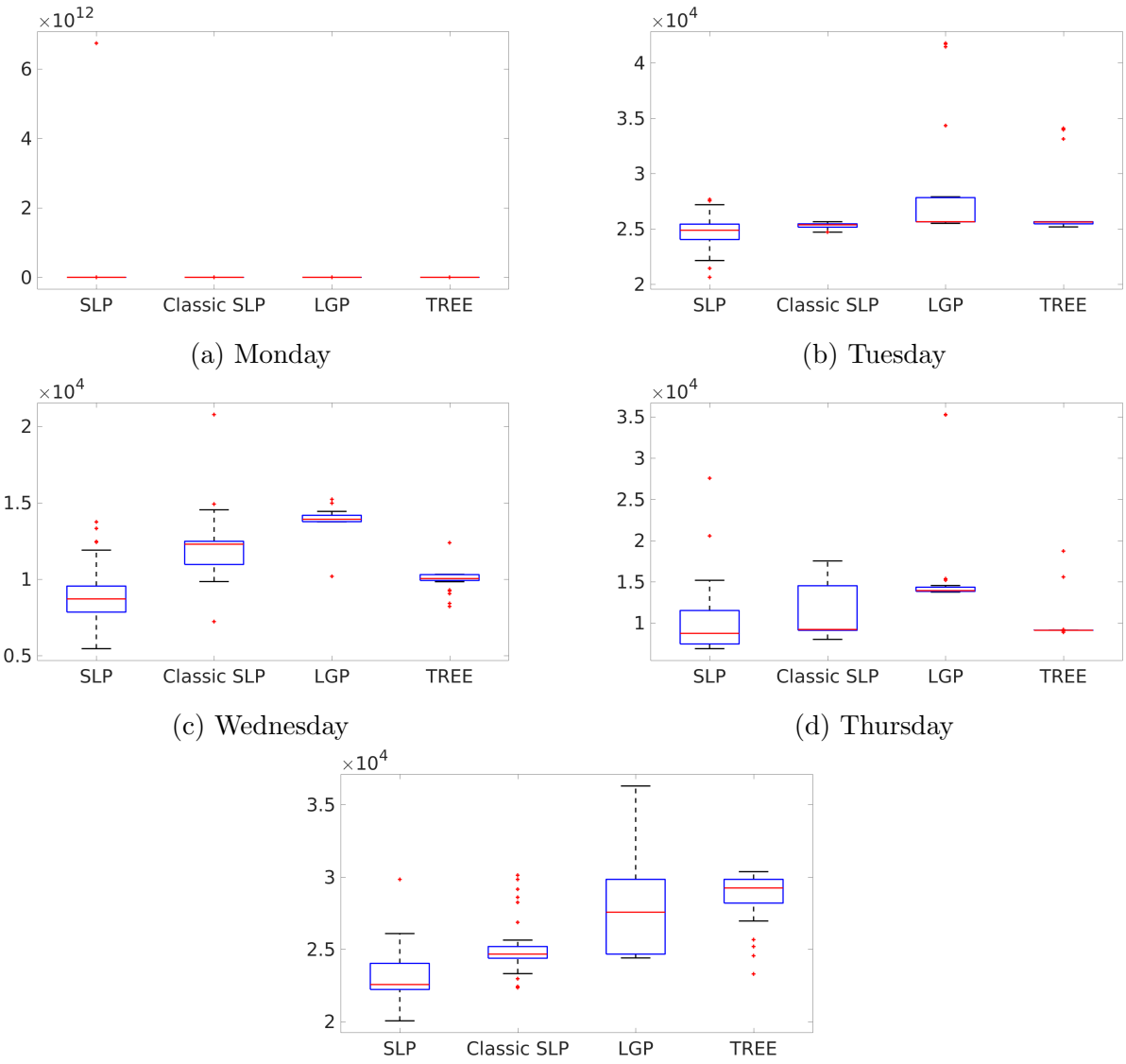
(c) Wednesday

(d) Thursday

(e) Friday

Figura 2.9: Boxplots of the error rate (MSE) for building B2

with these matrices. For example, the *SLP* algorithm selected *Mondays* and *Tuesdays* to model *Friday*'s energy consumption for building *B1*, and Figure 2.7a shows the highest correlation for these days. As another example, algorithm *SLP* was able to select *Wednesdays* only to model *Thursday*'s energy consumption for building *B3*. This is consistent with the information from Figure 2.7c, where it is shown a correlation coefficient R=0.99 between Wednesdays and Thursdays. Thus, our experimentation suggests that symbolic regression could not only find the best algebraic expression that models output data from a multivariate set of input data, but also perform a feature selection over the input data, in real problems where the number of features is not high. We are aware that the symbolic regression algorithms used in this work are not



(a) Monday

(b) Tuesday

(c) Wednesday

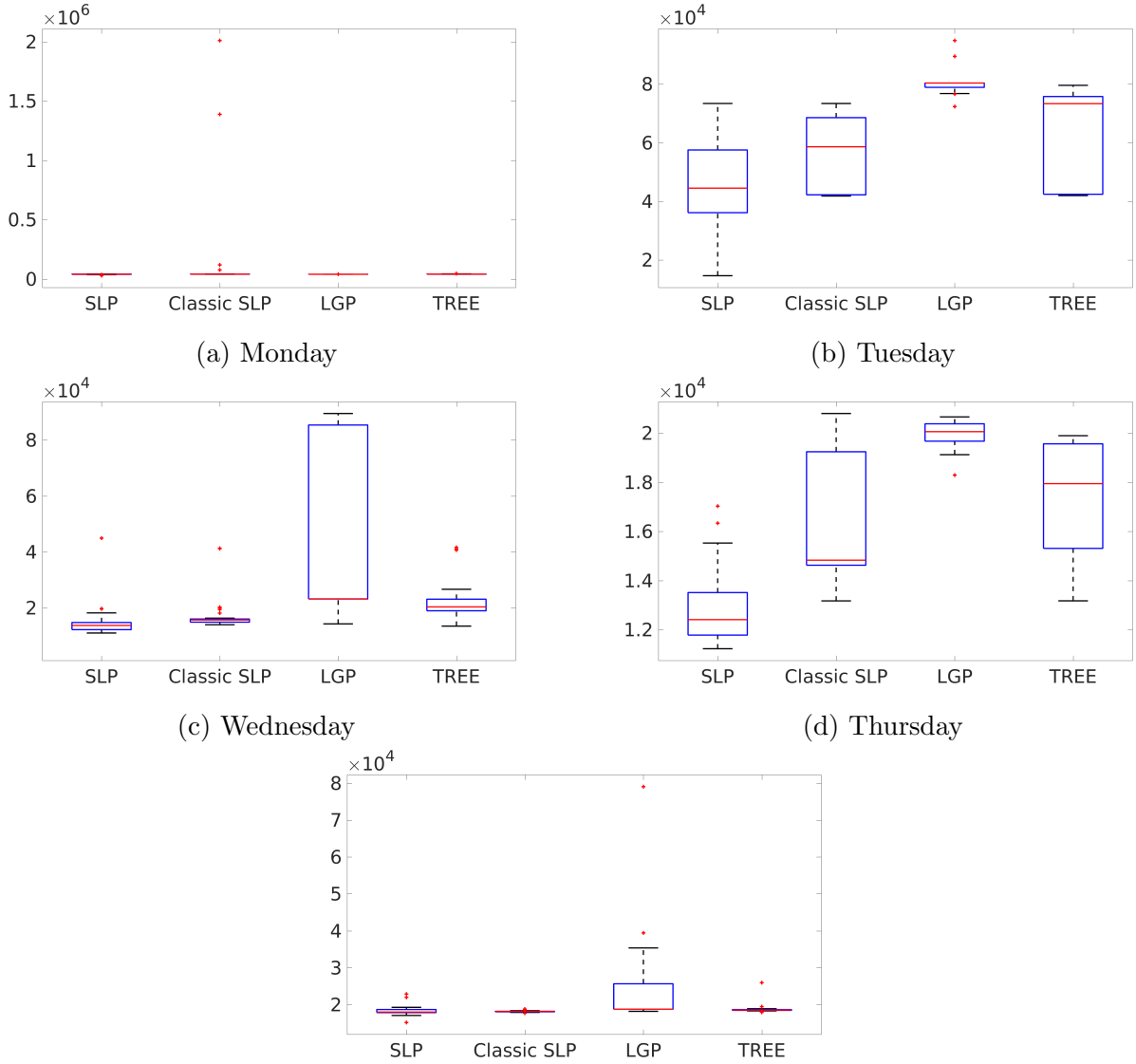
(d) Thursday

(e) Friday

Figura 2.10: Boxplots of the error rate (MSE) for building B3

designed for feature selection, but for finding the most accurate algebraic expressions. However, after these results were obtained, we believed that studying this feature could be worthy, and the analysis of results obtained promising outcomes. Nevertheless, the feature selection capability must be tested in high-dimensional problems in future works, that should consider how the search space grows as input data dimensions do, and also how to include feature selection capability into the algorithm objectives.

Finally, we analyze the interpretability of the algebraic expressions returned by the symbolic regression algorithms. Equation 2.4 shows as an example of the solution found by *SLP* to model *Thursdays*'s (d_4) energy consumption considering the energy consumption of *Mondays* (d_1),



(a) Monday

(b) Tuesday

(c) Wednesday

(d) Thursday

(e) Friday

Figura 2.11: Boxplots of the error rate (MSE) for building B4

Tuesdays (d_2) and *Wednesdays* (d_3), for building *B2*. This algebraic expression is a suitable example to ilustrate the interpretability of the results. However, although the formula can be understood by any analyst, or manager, we may notice that the algorithms do not return the simplified formula. From our point of view, this is not a limitation since there are algorithms that could be used after the algorithm execution to perform such task. However, we would like to remark that the solutions returned have a balance between complexity and interpretability, and provide accurate solutions that could be used for decision-making in higher levels analyses.

$$d_4 = min(2359{,}15, (((382 * d_1) + ((d_3 + min(d_3, -626{,}72)) * d_2))^{0,5})) \tag{2.4}$$

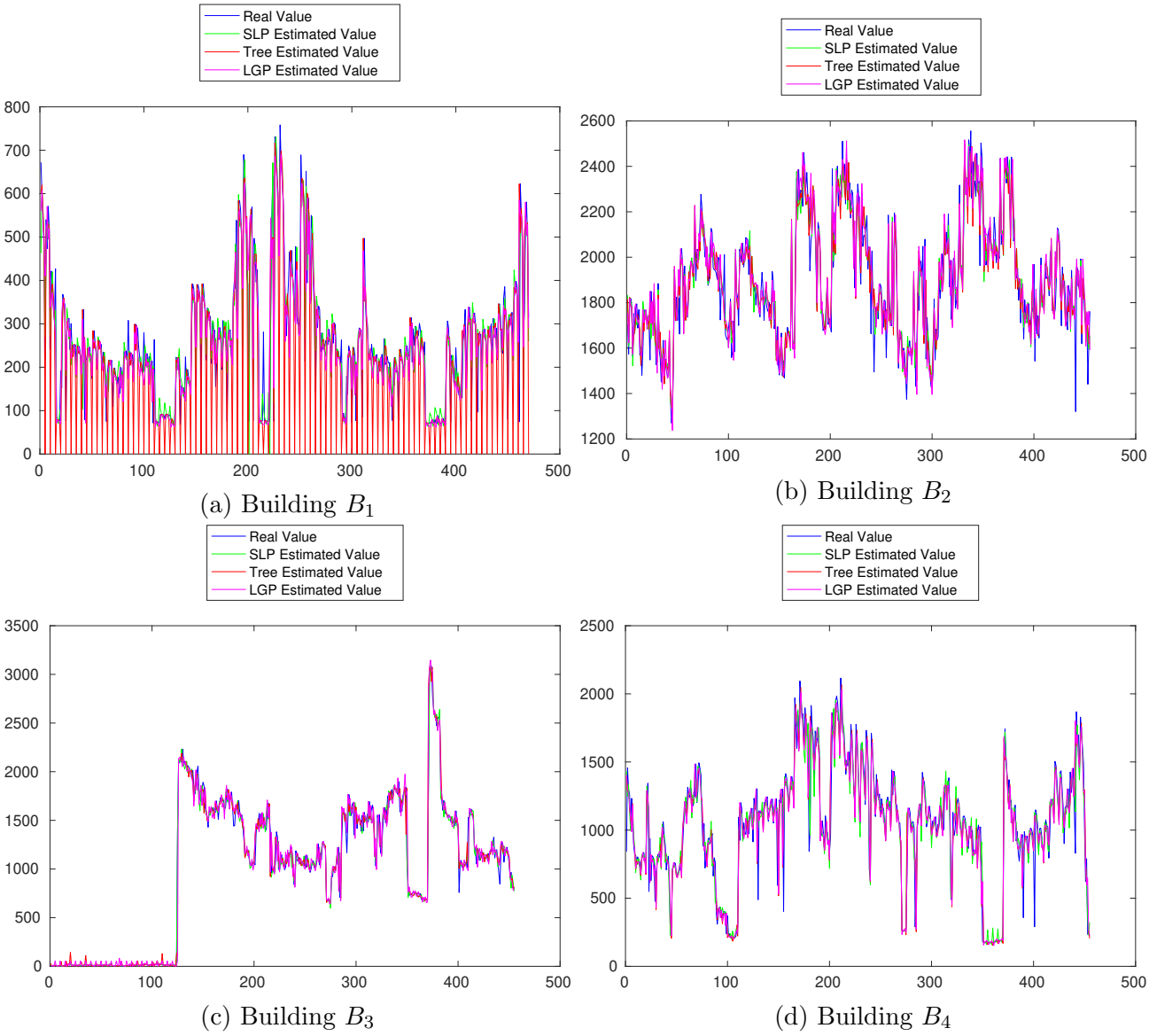
	B1					
	SLP *vs* Tree	**SLP *vs* LGP**	**LGP *vs* Tree**	**SLP *vs* Classic SLP**	**Classic SLP *vs* Tree**	**Classic SLP *vs* LGP**
Monday	$3{,}8 * 10^{-4}$ (-)	$9{,}86 * 10^{-7}$ (+)	$4{,}23 * 10^{-11}$ (-)	$2{,}69 * 10^{-3}$ (-)	0.46 (x)	$3{,}63 * 10^{-6}$ (+)
Tuesday	$2{,}57 * 10^{-9}$ (+)	$4{,}27 * 10^{-11}$ (+)	$3{,}21 * 10^{-10}$ (-)	$1{,}54 * 10^{-6}$ (+)	0.02 (+)	0.03(-)
Wednesday	$1{,}4 * 10^{-9}$ (+)	$5{,}22 * 10^{-7}$ (+)	$4{,}34 * 10^{-9}$ (+)	$1{,}8 * 10^{-5}$ (+)	$1{,}66 * 10^{-6}$ (+)	0.03 (-)
Thursday	$5{,}4 * 10^{-3}$ (-)	$1{,}5 * 10^{-7}$ (-)	$1{,}21 * 10^{-8}$ (-)	$8{,}32 * 10^{-4}$ (-)	0,09 (x)	$5{,}48 * 10^{-8}$ (+)
Friday	$9{,}17 * 10^{-7}$ (+)	$2{,}82 * 10^{-11}$ (+)	$1{,}29 * 10^{-8}$ (-)	$2{,}51 * 10^{-5}$ (+)	$3{,}75 * 10^{-3}$ (+)	$3{,}45 * 10^{-11}$ (+)
	B2					
Monday	$4{,}91 * 10^{-6}$ (+)	$7{,}14 * 10^{-9}$ (+)	$2 * 10^{-4}$ (-)	$1{,}62 * 10^{-4}$ (+)	0.07 (x)	$1{,}38 * 10^{-5}$ (+)
Tuesday	$8{,}3 * 10^{-4}$ (+)	$1{,}9 * 10^{-9}$ (+)	$2{,}94 * 10^{-10}$ (-)	0.02 (+)	$1{,}24 * 10^{-4}$ (+)	$2{,}2 * 10^{-10}$ (+)
Wednesday	$3{,}3 * 10^{-3}$ (+)	$2{,}77 * 10^{-11}$ (+)	$2{,}19 * 10^{-11}$ (-)	$1{,}93 * 10^{-8}$ (+)	$9{,}78 * 10^{-7}$ (-)	$6{,}15 * 10^{-7}$ (+)
Thursday	0.08 (x)	$1{,}2 * 10^{-9}$ (-)	$2{,}68 * 10^{-11}$ (-)	$1{,}13 * 10^{-3}$ (-)	0.02 (-)	$4{,}81 * 10^{-6}$ (+)
Friday	$3 * 10^{-4}$ (+)	$2{,}35 * 10^{-5}$ (+)	$4{,}73 * 10^{-5}$ (-)	$3{,}09 * 10^{-4}$ (+)	0.59 (x)	$2{,}43 * 10^{-3}$ (+)
	B3					
Monday	$1{,}12 * 10^{-6}$ (-)	$1{,}32 * 10^{-8}$ (-)	$6{,}55 * 10^{-5}$ (-)	$1{,}4 * 10^{-06}$ (-)	0.17 (x)	$1{,}2 * 10^{-3}$ (+)
Tuesday	$2{,}81 * 10^{-5}$ (+)	$4{,}38 * 10^{-7}$ (+)	$5{,}4 * 10^{-3}$ (-)	$6{,}07 * 10^{-3}$ (+)	$1{,}15 * 10^{-3}$ (+)	$4{,}92 * 10^{-8}$ (+)
Wednesday	$3 * 10^{-4}$ (+)	$4{,}55 * 10^{-11}$ (+)	$9{,}9 * 10^{-11}$ (-)	$2{,}4 * 10^{-6}$ (+)	$1{,}47 * 10^{-5}$ (-)	$8{,}33 * 10^{-7}$ (+)
Thursday	0.14 (x)	$3{,}12 * 10^{-7}$ (+)	$2{,}94 * 10^{-8}$ (-)	0.01 (+)	$5{,}71 * 10^{-2}$ (x)	$3{,}64 * 10^{-4}$ (+)
Friday	$1{,}15 * 10^{-8}$ (+)	$9{,}59 * 10^{-8}$ (+)	0.24 (x)	$2{,}76 * 10^{-5}$ (+)	$1{,}59 * 10^{-6}$ (+)	$9{,}1 * 10^{-4}$ (+)
	B4					
Monday	$3{,}5 * 10^{-3}$ (+)	0.92 (x)	$2{,}48 * 10^{-5}$ (+)	0.01 (+)	0.68 (x)	$1{,}73 * 10^{-5}$ (-)
Tuesday	$4{,}8 * 10^{-4}$ (+)	$2{,}63 * 10^{-11}$ (+)	$1{,}32 * 10^{-9}$ (-)	0.02 (+)	0.11 (x)	$3{,}43 * 10^{-11}$ (+)
Wednesday	$4{,}87 * 10^{-8}$ (+)	$1{,}03 * 10^{-9}$ (+)	$7{,}3 * 10^{-5}$ (-)	$1{,}29 * 10^{-5}$ (+)	$1{,}1 * 10^{-4}$ (+)	$8{,}46 * 10^{-8}$ (+)
Thursday	$4{,}35 * 10^{-9}$ (+)	$2{,}74 * 10^{-11}$ (+)	$4{,}33 * 10^{-8}$ (-)	$3{,}08 * 10^{-8}$ (+)	0.24 (x)	$3{,}13 * 10^{-6}$ (+)
Friday	$1{,}5 * 10^{-3}$ (+)	$9{,}11 * 10^{-7}$ (+)	$1{,}5 * 10^{-4}$ (+)	0.16 (x)	$4{,}39 * 10^{-6}$ (+)	$1{,}43 * 10^{-9}$ (+)

Tabla 2.6: Statistical tests to compare algorithms in the results of all working days of buildings B1 to B4

To conclude with the analysis of results, Figures 2.12a to 2.12d show the original datasets and the results of the modelled data in the complete energy consumption data series (both training and test sets) for each building. As it can be observed, the algebraic expressions found by the algorithms are able to fit the data correctly under a visual analysis. This fact suggests that the search algorithms perform a feature selection capabilities automatically that can approximate the real data suitably, even when energy consumption peaks take place for each building. Although the results shown in tables 2.2 to 2.5 show a high MSE value, we can verify in the figures that the modelled energy consumption fits correctly the real values. As a result, we conclude that SLPs are a promising alternative for real applications of symbolic regression.

5 Conclusions

In this paper, we have used symbolic regression to model the energy consumption of the working days in different public buildings of the University of Granada. The results suggest that symbolic regression can be used to find algebraic expressions that model energy consumption accurately, using different representation models such as trees, Straight Line Programs or Linear



(a) Building B_1

(b) Building B_2

(c) Building B_3

(d) Building B_4

Figura 2.12: Plots of real data (blue), SLP estimated data (green), Tree estimated data (red), and LGP estimated data (magenta) for buildings B_1 to B_4

Programs. The outcomes of our experimentation shows that modelling energy consumption can be performed accurately, and return interpretable results that can be used for decision-making. The SLP representation allowed to model the energy consumption of the working days, and simultaneously obtaining the best subset of dependent variables that allow to find the most accurate regression hypothesis. This fact may help the experts and managers in decision-making processes and also to detect energy consumption anomalies.

Our experimentation considers 3 types of representation: classic tree, Linear Programs, and Straight Line Programs. If we compare all the representations studied, we can conclude that

SLP is the most suitable representation technique in the problems addressed, because of its simplicity, the efficiency for algebraic expression evaluation, and genetic operators design. SLP representation is also the solution that best experimental results has provided, considering accuracy.

On the other hand, our study does not only suggest that the use of SLP helps to improve the accuracy of the resulting regression models being compared with classic tree and LGP representations, but also shows that symbolic regression is able to perform an automatic feature selection of dependent variables simultaneously to the model training.

We have also studied the benefits and drawbacks of estimating the algebraic expression parameters online during the algorithm execution. In this case, our results suggest that performing such automatic optimization can help to overcome local optima, and to obtain more accurate results than using classic methodology. However, the hybridation with least square estimation makes the time complexity of the algorithm increase by a factor of 100. In addition to finding a good model fitting, the size of algebraic expressions takes on a large size, being compared with simplified algebraic expressions. However, SLPs have proven to outperform the algebraic expression size of traditional representations such as trees, specially if the estimation of the algebraic expression parameters is performed.

Despite these advantages, we have also found limitations and interests for future works. Firstly, although SLPs have been able to overcome local optima being compared with traditional techniques, the optimization of algebraic expression parameters is computationally expensive. In future works, we will attempt to develop strategies to reduce this complexity, while optimizing the algebraic expression parameters. Secondly, the ability to perform automatic feature extraction is consistent with the initial hypothesis of the work, which states that the optimization algorithm should select the input variables that minimize the error of the resulting algebraic expression. Although we have discussed that this strategy has been working in our practical problems, the optimization algorithms are not designed to the purpose of feature selection. Future works will also be addressed to design algorithms that can handle the accuracy optimization and feature selection in the design within, specially to address problems with high-dimensional

data. We believe that multi-objective optimization could be a suitable start for this piece of research. Finally, even if SLPs have been able to overcome local optima, they still have a representation problem: The same solution can have multiple representations, due to properties such as transitivity and associativity of mathematical operators. In future works, we will address the problem of reducing the search space considering these properties. Here, we believe that the inclusion of semantics or shrinking the representation itself could help to achieve our goal, and therefore to improve accuracy.

Acknowledgements

This work has been supported by the project TIN201564776-C3-1-R.

Algorithm 1 Genetic procedure to train SLP

Require: Input: S_p, the size of the population
Require: Input: P_c, the crossover probability
Require: Input: P_m, the mutation probability
Require: Input $\bar{x} = (x_1, x_2, ..., x_n)$, input data for SLP evaluation
Require: Input $\bar{y} = (y_1)$, output data for SLP evaluation
Ensure: Output: $SLP(1...N)$, a sequence of rules that encode the algebraic expression

```
{Initialization of population}
for counter i=1 to S_p do
    Initialize P(i):= Random SLP
    w̄:= Optimize(x̄, ȳ, P(i))
    Evaluate(P(i), w̄, x̄, ȳ)
end for
{Evolutionary process}
while No stopping criterion is fulfilled do
    C:={∅} {Offspring population initialization to empty}
    for counter i=1 to S_p/2 do
        Set {P_1, P_2}:= Select two different elements from P with tournament selection [FL10] {Parent selection}
        {Crossover }
        if U(0,1) < P_c then
            Set {C_1, C_2}:= recombination(P_1, P_2)
        else
            Set C_1:=P_1, C_2:=P_2
        end if
        {Mutation}
        for counter j=1 to 2 do
            if U(0,1) < P_m then
                Update C_j := mutation(C_j)
            end if
            w̄:= Optimize(x̄, ȳ, C_j)
            Evaluate(C_j, w̄, x̄, ȳ)
            Update C:=C ∪ {C_j}
        end for
    end for
    {Elitism}
    if best solution of C is worse than best solution of P then
        Replace worst solution of C with best solution of P
    end if
    {Replacement}
    Update P:= C
end while
return Best solution of P
```

Algorithm 2 Procedure to create a random SLP

Require: Input: N, the size of the SLP
Require: Input: O, the set of operators
Require: Input: $\bar{x} = \{x_1, x_2, ..., x_n\}$, the set of dependent variables, inputs to f
Require: Input: $\bar{w} = \{w_1, w_2, ..., w_k\}$, the set of parameters of f
Ensure: Output: $SLP(1...N)$, a sequence of rules that encode the algebraic expression
  for counter i=1 to N **do**
    O_{U_i}:= element from O randomly selected
    $R_{U_i,1}$:= element from $\{\bar{x} \cup \bar{w} \cup \{U_1, ..., U_{i-1}\}\}$ randomly selected
    if O_{U_i} is unary **then**
      Create rule $U_i \rightarrow R_{U_i,1} O_{U_i} \emptyset$
    else
      $R_{U_i,2}$:= element from $\{\bar{x} \cup \bar{w} \cup \{U_1, ..., U_{i-1}\}\}$ randomly selected
      Create rule $U_i \rightarrow R_{U_i,1} O_{U_i} R_{U_i,1}$
    end if
    $SLP(i) := U_i$
  end for
  return SLP

References

[AB00] D. A. Augusto y H. J. C. Barbosa. «Symbolic regression via genetic programming». En: *6th Brazilian Symposium on Neural Networks, Rio de Janeio, Brazil*. 2000, págs. 173-178.

[Alo+09] César Luis Alonso y col. «A New Linear Genetic Programming Approach Based on Straight Line Programs: Some Theoretical and Experimental Aspects». En: *International Journal on Artificial Intelligence Tools* 18 (2009), págs. 757-781.

[Ang07] James B. Ang. «CO2 emissions, energy consumption, and output in France». En: *Energy Policy* 35 (2007), págs. 4772-4778.

[AZN94] Ahmed Z. Al-Garni, Syed M. Zubair y Javeed S. Nizami. «A regression model for electric-energy-consumption forecasting in Eastern Saudi Arabia». En: *Energy* 19 (1994), págs. 1043-1049.

[BAB14] M.R. Braun, H. Altan y S.B.M. Beck. «Using regression analysis to predict the future energy consumption of a supermarket in the UK». En: *Applied Energy* 130 (2014), págs. 305-313.

[Bal+13] Bharathan Balaji y col. «Sentinel: Occupancy Based HVAC Actuation Using Existing WiFi Infrastructure Within Commercial Buildings». En: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA, 2013, 17:1-17:14.

[BAN02] Maumita Bhattacharya, Ajith Abraham y Baikunth Nath. «A Linear Genetic Programming Approach for Modelling Electricity Demand Prediction in Victoria». En: *Hybrid Information Systems*. Heidelberg: Physica-Verlag HD, 2002, págs. 379-393.

[BB10] Markus F. Brameier y Wolfgang Banzhaf. *Linear Genetic Programming*. 1st. Springer Publishing Company, Incorporated, 2010. ISBN: 978-1-4419-4048-3.

[BC02a] Mariusz Boryczka y Zbigniew J. Czech. «Solving Approximation Problems By Ant Colony Programming». En: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, págs. 133-. ISBN: 1-55860-878-8. URL: http://dl.acm.org/citation.cfm?id=646205.682953.

[BC07] G. Bandyopadhyay y S. Chattopadhyay. «Single hidden layer artificial neural network models versus multiple linear regression model in forecasting the time series of total ozone». En: *International Journal of Environmental Science & Technology* 4 (2007), págs. 141-149.

[BD02a] Lynne Billard y Edwin Diday. «Symbolic Regression Analysis». En: *Classification, Clustering, and Data Analysis: Recent Advances and Applications*. Berlin, Heidelberg, 2002, págs. 281-288.

[Beh+12] R. Behera y col. «An Application of Genetic Programming for Power System Planning and Operation». En: *International Journal on Control System and Instrumentation, March*. 2012, págs. 15-20.

[Ber+10] Josep Ll. Berral y col. «Towards Energy-aware Scheduling in Data Centers Using Machine Learning». En: *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. New York, NY, USA, 2010, págs. 215-224.

[Ber84] Stuart J. Berkowitz. «On Computing the Determinant in Small Parallel Time Using a Small Number of Processors». En: *Inf. Process. Lett.* 18 (1984), págs. 147-150.

[BF16] Adrien Bibal y Benoît Frenay. «Interpretability of Machine Learning Models and Representations: an Introduction». English. En: *24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2016, págs. 77-82.

[BK13b] Florian Benz y Timo Kötzing. «An effective heuristic for the smallest grammar problem». En: *Genetic and Evolutionary Computation Conference, Amsterdam, The Netherlands, July 6-10*. 2013, págs. 487-494.

[BLS13] Ilhem Boussaïd, Julien Lepagnot y Patrick Siarry. «A survey on optimization metaheuristics». En: *Information Sciences* 237 (2013). Prediction, Control and Diagnosis using Advanced Neural Computations, págs. 82-117. DOI: https://doi.org/10.1016/j.ins.2013.02.041. URL: http://www.sciencedirect.com/science/article/pii/S0020025513001588.

[Cai+09] W.G. Cai y col. «China building energy consumption: Situation, challenges and corresponding measures». En: *Energy Policy* 37 (2009), págs. 2054-2059.

[Cas+15] Mauro Castelli y col. «Prediction of energy performance of residential buildings: A genetic programming approach». En: *Energy and Buildings* 102 (2015), págs. 67-74.

[CFW01] Tsangyao Chang, Wenshwo Fang y Li-Fang Wen. «Energy consumption, employment, output, and temporal causality: evidence from Taiwan based on cointegration and error-correction modelling techniques». En: *Applied Economics* 33 (2001), págs. 1045-1056.

[Cia+14] Lucio Ciabattoni y col. «Fuzzy logic home energy consumption modeling for residential photovoltaic plant sizing in the new Italian scenario». En: *Energy* 74 (2014), págs. 359-367. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2014.06.100. URL: http://www.sciencedirect.com/science/article/pii/S0360544214007993.

[CPB17] Alfonso Capozzoli, Marco Savino Piscitelli y Silvio Brandi. «Mining typical load profiles in buildings to support energy management in the smart city context». En: *Energy Procedia* 134 (2017), págs. 865-874. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2017.09.545. URL: http://www.sciencedirect.com/science/article/pii/S187661021734674X.

[CT14] Jui-Sheng Chou y Abdi Suryadinata Telaga. «Real-time detection of anomalous power consumption». En: *Renewable and Sustainable Energy Reviews* 33 (2014), págs. 400-411. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2014.01.088. URL: http://www.sciencedirect.com/science/article/pii/S1364032114001142.

[CW17a] Wenqiang Cui y Hao Wang. «A New Anomaly Detection System for School Electricity Consumption Data». En: *Information* 8.4 (2017). ISSN: 2078-2489. DOI: 10.3390/info8040151. URL: http://www.mdpi.com/2078-2489/8/4/151.

[DG49] D.Cochrane y G.H.Orcutt. «Application of Least Squares Regression to Relationships Containing Auto-Correlated Error Terms». En: *Journal of the American Statistical Association* 44 (1949), págs. 32-61.

[EK99] Alice E. Smith y Anthony K. Mason. «Cost Estimation Predictive Modeling: Regression versus Neural Network». En: *The Engineering Economist* 42 (1999), págs. 137-161.

[Ekw+13] T. Ekwevugbe y col. «Real-time building occupancy sensing using neural-network based sensor network». En: *7th IEEE International Conference on Digital Ecosystems and Technologies (DEST), Menlo Park, CA*. 2013, págs. 114-119.

[ENP12] Richard E. Edwards, Joshua New y Lynne E. Parker. «Predicting future hourly residential electrical consumption: A machine learning case study». En: *Energy and Buildings* 49 (2012), págs. 591-603.

[FB15] Nelson Fumo y M.A. Rafe Biswas. «Regression analysis for prediction of residential energy consumption». En: *Renewable and Sustainable Energy Reviews* 47 (2015), págs. 332-343.

[Fig+05] V. Figueiredo y col. «An electric energy consumer characterization framework based on data mining techniques». En: *IEEE Transactions on Power Systems* 20.2 (2005), págs. 596-602. DOI: 10.1109/TPWRS.2005.846234.

[FL10] Yongsheng Fang y Jun Li. «A Review of Tournament Selection in Genetic Programming». En: *Advances in Computation and Intelligence: 5th International Symposium, ISICA, Wuhan, China, October 22-24*. 2010, págs. 181-192.

[Fra15] Jr. Frank E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis.* Springer Series in Statistics, 2015.

[GH06] Jan G. De Gooijer y Rob J. Hyndman. «25 years of time series forecasting». En: *International Journal of Forecasting* 22.3 (2006), págs. 443-473. DOI: https://doi.org/10.1016/j.ijforecast.2006.01.001. URL: http://www.sciencedirect.com/science/article/pii/S0169207006000021.

[Giu+98] M. Giusti y col. «Straight-line programs in geometric elimination theory». En: *Journal of Pure and Applied Algebra* 124 (1998), págs. 101-146.

[GNC16] Jun Guan, Natasa Nord y Shuqin Chen. «Energy planning of university campus building complex: Energy usage and coincidental analysis of individual buildings with a case study». En: *Energy and Buildings* 124 (2016), págs. 99-111. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.04.051. URL: http://www.sciencedirect.com/science/article/pii/S0378778816303164.

[Huo+08] L. Huo y col. «Short-Term Load Forecasting Based on Improved Gene Expression Programming». En: *2008 4th IEEE International Conference on Circuits and Systems for Communications*. Mayo de 2008, págs. 745-749.

[Kho+13] Benyamin Khoshnevisan y col. «Modeling of energy consumption and GHG (greenhouse gas) emissions in wheat production in Esfahan province of Iran using artificial neural networks». En: *Energy* 52 (2013), págs. 333-338.

[Kho+16] Hamid R. Khosravani y col. «A Comparison of Energy Consumption Prediction Models Based on Neural Networks of a Bioclimatic Building». En: *Energies* 9 (2016), pág. 1.

[KR13] Arash Kialashaki y John Reisel. «Modeling of the energy demand of the residential sector in the United States using regression models and artificial neural networks». En: *Applied Energy* 108 (ago. de 2013), págs. 271-280.

[Kri02] Teresa Krick. *Straight-line Programs in Polynomial Equation Solving*. 2002.

[Lan98] W. B. Langdon. «Genetic Programming — Computers Using "Natural Selection" to Generate Programs». En: *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!* Boston, MA: Springer US, 1998, págs. 9-42.

[Li+08] G. Li y col. «Instruction-matrix-based genetic programming». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 38.4 (2008), págs. 1036-1049. DOI: 10.1109/TSMCB.2008.922054.

[McC05] John McCall. «Genetic algorithms for modelling and optimisation». En: *Journal of Computational and Applied Mathematics* 184.1 (2005), págs. 205-222. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2004.07.034. URL: http://www.sciencedirect.com/science/article/pii/S0377042705000774.

[McK+10] Robert I. McKay y col. «Grammar-based Genetic Programming: a survey». En: *Genetic Programming and Evolvable Machines* 11.3 (2010), págs. 365-396. ISSN: 1573-7632. DOI: 10.1007/s10710-010-9109-y.

[MKJ12] Alberto Moraglio, Krzysztof Krawiec y Colin G. Johnson. «Geometric Semantic Genetic Programming». En: *Parallel Problem Solving from Nature - PPSN XII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 21-31.

[MPV12] D.C. Montgomery, E.A. Peck y G.G. Vining. *Introduction to Linear Regression Analysis*. Wiley Series in Probability an. Wiley & Sons, 2012.

[MT00] Julian F. Miller y Peter Thomson. «Cartesian Genetic Programming». En: *Genetic Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, págs. 121-132. ISBN: 978-3-540-46239-2.

[MWB95] B. McKay, M. J. Willis y G. W. Barton. «Using a tree structured genetic algorithm to perform symbolic regression». En: *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. 1995, págs. 487-492.

[PC13] Anuradha Purohit y Narendra S. Choudhari. «Code Bloat Problem in Genetic Programming». En: *International Journal of Scientific and Research Publications* 3.4 (2013), págs. 1-5. ISSN: 2250-3153.

[PJP14] J. Pan, R. Jain y S. Paul. «A Survey of Energy Efficiency in Buildings and Microgrids using Networking Technologies». En: *IEEE Communications Surveys Tutorials* 16 (2014), págs. 1709-1731.

[PM05] P A Paul y G P Munkvold. «Regression and artificial neural network modeling for the prediction of gray leaf spot of maize.» En: *Phytopathology* 95 4 (2005), págs. 388-96.

[Pol+07] Riccardo Poli y col. *Genetic programming an introductory tutorial and a survey of techniques and applications*. Inf. téc. Department of Computing y Electronic Systems, University of Essex, 2007.

[Rue+17a] R. Rueda y col. «Preliminary Evaluation of Symbolic Regression Methods for Energy Consumption Modelling». En: *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, February 24-26*. 2017, págs. 39-49.

[Sad+11] Hossein Sadeghi y col. «Estimation of Electricity Demand in Residential Sector Using Genetic Algorithm Approach». En: *International Journal of Industiral Engineering & Producion Research* 22 (2011), págs. 43-50.

[Sad09] Perry Sadorsky. «Renewable energy consumption and income in emerging economies». En: *Energy Policy* 37 (2009), págs. 4021-4028.

[Sal05] Timothy I. Salsbury. «A survey of control technologies in the building automation industry». En: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, págs. 90-100. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20050703-6-CZ-1902.01397. URL: http://www.sciencedirect.com/science/article/pii/S1474667016374092.

[SCS12] J. Sequera, J. del Castillo Diez y L. Sotos. «Document Clustering with Evolutionary Systems through Straight-Line Programs "slp"». En: *Intelligent Learning Systems and Applications* 4 (2012), págs. 303-318.

[Sha+13] Pervez Hameed Shaikh y col. «Intelligent Optimized Control System for Energy and Comfort Management in Efficient and Sustainable Buildings». En: *Procedia Technology* 11 (2013). 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013, págs. 99-106. ISSN: 2212-0173. DOI: https://doi.org/10.

1016/j.protcy.2013.12.167. URL: http://www.sciencedirect.com/science/article/pii/S2212017313003216.

[TAB16] Erdi Tosun, Kadir Aydin y Mehmet Bilgili. «Comparison of linear regression and artificial neural network model of a diesel engine fueled with biodiesel-alcohol mixtures». En: *Alexandria Engineering Journal* 55 (2016), págs. 3081-3089.

[TY07] Geoffrey K.F. Tso y Kelvin K.W. Yau. «Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks». En: *Energy* 32 (2007), págs. 1761-1768.

[VCS14] Leonardo Vanneschi, Mauro Castelli y Sara Silva. «A survey of semantic methods in genetic programming». En: *Genetic Programming and Evolvable Machines* 15.2 (jun. de 2014), págs. 195-214. ISSN: 1573-7632. DOI: 10.1007/s10710-013-9210-0. URL: https://doi.org/10.1007/s10710-013-9210-0.

[WS17] Zeyu Wang y Ravi S. Srinivasan. «A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models». En: *Renewable and Sustainable Energy Reviews* 75 (2017), págs. 796-808. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.10.079. URL: http://www.sciencedirect.com/science/article/pii/S1364032116307420.

[Yu+10] Zhun Yu y col. «A decision tree method for building energy demand modeling». En: *Energy and Buildings* 42 (2010), págs. 1637-1646.

[YZ11] Jian Yao y Neng Zhu. «Enhanced supervision strategies for effective reduction of building energy consumption—A case study of Ningbo». En: *Energy and Buildings* 43 (2011), págs. 2197-2202.

[ZK13] Mahnaz Zarei y Hassan Khademi-Zare. «Energy Consumption Modeling in Residential Buildings». En: *International Journal of Architecture and Urban Development* 3 (2013), págs. 35-38.

[ZOC16] J. Zhong, Y. S. Ong y W. Cai. «Self-Learning Gene Expression Programming». En: *IEEE Transactions on Evolutionary Computation* 20 (2016), págs. 65-80.

2.2. An Ant Colony approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling.

- **Referencia**: R.Rueda, L.G.B.Ruiz, M.P.Cuéllar, M.C.Pejalajar (2020). An Ant Colony approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling. International Journal of Approximate Reasoning, 121, 23-38
- **Estado**: Publicado
- **Factor de impacto** (JCR 2019): 2.678
- **Categoría**: Posición 55/136 en el área "Computer Science, Artificial Intelligence". Q2
- **DOI**: 10.1016/j.ijar.2020.03.005.
- **Revista/Editorial**: International Journal of Approximate Reasoning / Elsevier

An Ant Colony Optimization approach for symbolic regression using Straight Line Programs. Application to energy consumption modelling

R.Rueda, L.G.B.Ruíz, M.P.Cuéllar, M.C.Pegalajar

Department of Computer Science and Artificial Intelligence, University of Granada, Spain.

C/. Pdta. Daniel Saucedo Aranda s.n., 18071 Granada (Spain)

Abstract

The increase of energy consumption and their direct effects on pollution and global warming have motivated governments to develop new strategies to promote a better usage of energy. One of the most important aspects related to energy efficiency is the need for a suitable model of energy consumption that can be used to make predictions or to aid experts in high level decision making processes. Symbolic regression techniques can be used to discover an energy consumption model that fits these purposes. Traditionally, the problem of symbolic regression has been solved by using genetic programming approaches to find the algebraic expression that best fits the regression problem data, where each expression is encoded as a tree structure. In previous works, we found that a different approach using Straight Line Programs as a representation technique could provide promising results for symbolic regression, although the size of the resulting algebraic expression might be increased when compared to the traditional approach. This work proposes an Ant Colony Optimization algorithm for Straight Line Programs to solve the problem, and makes a study to compare the approach with traditional genetic programming in a real energy consumption modelling problem.

Keywords: Energy Efficiency, Straight Line Programs, Ant Colony Optimization

1 Introduction

The increase of energy consumption in the building sector has become a major problem for many governments in the developed world, due to limited energy sources, increases in the price of energy and production costs, and high emissions of CO_2. For all of the aforementioned reasons, the research efforts to reduce energy consumption and to use energy efficiently have been increased substantially during the past two decades [PJP14]. More specifically, the advances in sensor technologies and communications allow us to study multiple problems regarding energy efficiency research, such as energy consumption forecasting [WS17; FB15], anomaly detection [CT14; CW17a], consumer profile mining [CPB17; Fig+05], energy demand planning [GNC16], and energy consumption modelling [CFW01], among others.

In our research, we are interested in the problem of energy consumption modelling, whose objective is to find mathematical or computational models that help to accurately approximate, or explain, energy consumption behaviour. Examples of approaches to model energy consumption in the literature are reference [YZW19], which uses a Bayesian semi-parametric quantile regression technique to model the energy consumption in a municipal wastewater plant; the study of computational intelligence methods to model the household electricity consumption in [KK18], the use of intelligent techniques (Genetic Programming, Multiple Regression, Artificial Neural Network, etc.) to build models that estimate the energy consumption of a building using weekdays energy consumption and outdoor data (temperature, wind speed, humidity, etc.) in [Amb+18], and reference [Rob+17], which makes use of machine learning techniques (linear regression, boosting, SVM, etc.) to estimate hourly energy consumption in residential buildings. As we can see, energy consumption modelling has been applied mostly to solve forecasting and correlation discovery problems, although it can also be applied in other scopes such as anomaly detection. Examples of previous efforts in this topic are reference [Zor+16], which describes a method of detecting abnormal energy consumption in buildings using machine learning, or reference [Ara+17], which proposes a ensemble anomaly detection framework that helps the building manager in decision-making problems.

Designing a suitable model of energy consumption depends greatly on the formulation of the problem and the requisites to be accomplished in the solution. However, although several works have emerged that accurately solve modelling or prediction problems, they fall in its low interpretability [FXW14]. Consequently, recent works attempt to find accurate and interpretable models. By means of example, this is the case of the proposal in reference [Al-+16], which developed a method based on decision trees to forecast energy consumption in buildings; or the approach in reference [RMB19], which proposed a genetic fuzzy system that builds interpretable knowledge bases for predicting energy consumption in smart buildings.

Therefore, as has been argued by Bratko in [Bra97] some data mining applications need to find a balance between accuracy and interpretability, such as applications of decision making in the scope of energy efficiency research. For this reason, in this work we use symbolic regression [BD02a] for energy consumption modelling. More specifically, the solution found by symbolic regression can be represented as algebraic expressions that can approximate the energy consumption data. Then, these algebraic expressions are sufficiently interpretable to provide an explanation of the energy consumption behaviour, and to be used for high-level decision making. Just to cite some scenarios where other researchers use symbolic regression to solve energy consumption problems, we find the work [Beh+12], which uses Genetic Programming (GP) to estimate demand and energy consumption, providing more accurate results than traditional regression analysis. The study [BAN02] uses Linear Genetic Programming to perform consumer electricity demand forecasting, and our previous work in reference [Rue+17b] studies Straight Line Programs and tree representation performance for the energy consumption modelling of a set of buildings in a compound.

Despite their potential benefits, algorithms and methods of the current techniques to solve symbolic regression are still under study. Examples of these problems that directly affect our research goals are a) the *bloating* problem [WD10; SDV12], which consists of the increase of size of variable-length representations during the process of symbolic regression-solving; b) the *representation* problem, aimed at finding the best symbolic representation model that helps to reduce the search space and to facilitate optimization algorithm design; or c) the *generalization* problem to prevent overfitting, which also relates to find simpler algebraic expressions with

reduced size that model training data patterns accurately. Different studies were carried out to minimize these limitations [Rue+17b; RRu+17] which use fixed-size structures such as Straight Line Programs (SLPs) to avoid bloating and representation problems. In the cited article, we concluded that SLPs are able to provide solutions with equal or better accuracy than neural networks in some cases, especially when the neural network models are recurrent and training algorithm gets easily trapped in local optima. In this article we propose an algorithm inspired by Ant Colony Optimization (ACO) [DBS06] to find accurate symbolic regression solutions with reduced size with regards to Genetic Programming algorithms used in the literature [Rue+17b; Alo+12].

In this work, we validate our proposal over a set of energy consumption data of public buildings at the University of Granada. We address our research by assuming that if exists a correlation between the energy consumption of the working days, then we can develop a method able to detect which days are related and how, and find an interpretable solution with high accuracy that explains the energy consumption of a working day in terms of the energy consumption of the remaining working days. The main contribution of this article is the formulation of SLP training as a graph traverse problem for its use within the ACO paradigm, and the design of algorithm components to help us to obtain symbolic regression solutions of a lower size regarding the genetic programming approach. The proposal is firstly validated over a classical ACO approach to compare the results between two approaches formulated as a graph traverse problem, and then we make a comparison with classical genetic algorithms to test the quality of the solutions found. To achieve these objectives, this manuscript is structured as follows: Section 2 describes the main concepts regarding symbolic regression and Ant Colony Optimization, as an introduction to the methods and techniques developed in this piece of research. Section 3 introduces the ACO approach. Experiments are conducted in real energy consumption data problems, and then analyzed in Section 4. Finally, Section 5 concludes and discusses future work.

2 Background in symbolic regression and Ant Colony Optimization

2.1 Symbolic regression and the representation problem

Given a set of so-called independent variables $\vec{x} = (x_1, x_2, ..., x_n)$ and dependent variables $\vec{y} = (y_1, y_2, ..., y_m)$, where $x_i, y_j \in \mathbb{R}, \forall i, j : 1 \leq i \leq n, 1 \leq j \leq m$, symbolic regression attempts to find an algebraic expression $\tilde{f}(\vec{x}, \vec{w})$ and parameters $\vec{w}$, where $\vec{w} = (w_1, w_2, ..., w_k)$, $w_i \in \mathbb{R}, \forall i : 1 \leq i \leq k$, such as $\vec{y} \approx \tilde{f}(\vec{x}, \vec{w})$. Symbolic regression can be viewed as an abstraction of traditional regression analysis techniques widely used in engineering and scientific research, such as linear regression or logistic regression. In traditional regression analysis, the regression hypothesis f is established in advance, and the objective is to find the values for parameters $\vec{w}$ that minimize an error measurement, as for instance $e(f(\vec{x}, \vec{w}), \vec{y}) = ||f(\vec{x}, \vec{w}) - \vec{y}||$. On the other hand, symbolic regression assumes not only that $\vec{w}$ are unknown in advance, but also f, and the objective is to find an approximation $\tilde{f}(\vec{x}, \vec{w})$ of the optimal algebraic expression that minimizes $e(\tilde{f}(\vec{x}, \vec{w}), \vec{y})$.

Symbolic regression problems have traditionally been addressed from the perspective of supervised learning in the machine learning community, where $\vec{x}$ and $\vec{y}$ are the input and output data, respectively, and the goal is to perform a search over a space of algebraic expressions to find the best expression $\tilde{f}$ that minimizes $e(\tilde{f}(\vec{x}, \vec{w}), \vec{y})$. Since the space of algebraic expressions is large [LRW16a; CLJ18], heuristic global search methods, such as Genetic Programming (GP) [MWB95], have been proposed in the literature to tackle the problem. Further information about learning, representation and GP algorithm design can be found in reference [PLM08a].

The traditional representation for algebraic expressions in GP is the tree representation [MWB95]. Recent studies in the past decade have drawn attention to alternative representations, with a special focus on linear model representations [McK+10], due to the simplicity and potential benefits regarding the traditional non-linear representation with trees. This study highlights

additional benefits of a fixed-size linear representation regarding the design of components of the optimization technique, such as the crossover and mutation operators in genetic algorithms, and the simplicity of reducing the effect of the bloating problem. Nowadays, we can find several approaches based on linear grammar representations such as Gene Expression Programming [Huo+08], Linear Programs [BB10], or Straight Line Programs (SLP) [Alo+09], among others.

As described in the introduction, in previous research we have explored the use of Straight Line Programs to solve energy consumption modelling problems from the perspective of Genetic Programming for symbolic regression [Rue+17b; Rue+18a; RRu+17], obtaining promising results regarding accuracy in real problem data. SLPs are grammar-based representations capable of encoding algebraic expressions for symbolic regression [Alo+09], and are inspired by Straight Line Grammars (SLG) [BK13b]. SLG is a formal grammar that can be described as a tuple (V,T,P,S), where V is the set of non-terminal symbols, T is the set of terminal symbols, P is the set of production rules and S is the non-terminal starting symbol of the grammar. Each production rule in P is a context-free grammar production rule, each of these production rules cannot generate loops. A SLG in Chomsky normal form that generates a single non-empty word is a Straight Line Program. On the other hand, in the symbolic regression problem addressed in this work, the set of terminal symbols (T) is composed by a set of known mathematical operators $O \in \{o_1, o_2, ..., o_l\}$ (typically unary or binary arithmetic operators), a set of terminal input data $\{x_1, x_2, ..., x_n\}$ and a set of constant parameters $\{w_1, w_2, ..., w_k\}$. Moreover, a SLP contains N production rules $\{U_1, U_2, ..., U_N\} \in V$, where U_N is the starting symbol (S) of the grammar. Each production rule of a SLP contains a mathematical operator and two operands, whose can be a terminal symbol of T or a non-terminal symbol in V. Finally, the non-terminal symbols used in a production rule must reference subsequent production rules to avoid recursion. Then, given a SLP, the generation of the algebraic expression encoded into a SLP starts at the production rule U_N. Moreover, each non-terminal symbol U_i in the rule consequent is iteratively replaced by its associated production rule from $i = N-1$ down to $i = 1$. Formula 2.5 shows an example of a SLP with maximum size $N = 6$ and parameters $\vec{w} = (w_1, w_2, w_3) = (4, 8, 3)$. If we apply the described procedure, then the algebraic expression encoded can be derived from U_6 as $\tilde{f}(\vec{x}, \vec{w}) = U_6; U_6 \Rightarrow U_3 + U_5 \Rightarrow U_3 + (U_1 + U_4) \Rightarrow U_3 + (U_1 + (w_3 + U_3)) \Rightarrow cos(U_2) + (U_1 + (w_3 + cos(U_2))) \Rightarrow$

$cos(w_2 * x) + (U_1 + (w_3 + cos(w_2 * x))) \Rightarrow cos(w_2 * x) + (x^{w_1} + (w_3 + cos(w_2 * x)))$. For algebraic expression evaluation purposes, the parameters $\vec{w}$ should also be substituted in a last step, therefore providing the expression $\tilde{f}(x, (4, 8, 3)) = cos(8 * x) + (x^4 + (3 + cos(8 * x)))$.

$$
\begin{aligned}
U_1 &\rightarrow pow(x, 4) \\
U_2 &\rightarrow 8 * x \\
U_3 &\rightarrow \cos(U_2) \\
U_4 &\rightarrow 3 + U_3 \\
U_5 &\rightarrow U_1 + U_4 \\
U_6 &\rightarrow U_3 + U_5
\end{aligned}
\tag{2.5}
$$

On the other hand, additional benefits are assigned to SLPs due to it can be represented as a directed acyclic graph (DAG), which implies a potential over classical structures such as trees. For example, the study of reference [SL07] compares tree and graph structures regarding Genetic Programming problems, and the outcomes of this research work suggest that graph structures are a promising alternative representation regarding trees, since the graph structure allows the reuse of nodes that represent pieces of the algebraic expression and reduces the effects of the bloating problem. Nevertheless, although SLPs are fixed-size structures and the bloating problem is limited because of this representation, in previous experimentations [Rue+18a] we observed that the resulting algebraic expressions obtained from SLP optimization were large with regards to their simplified form, and less interpretable. This drives the research study of this article, where we pursue the development of techniques targeted at finding a balance between SLP accuracy and size. Different methods can be found in the literature to solve this problem, such as model regularization [Alo+12], ant colony optimization [BC02b; GWJ04], or multi-objective optimization [Ble+01], among others. As mentioned in the introduction, our proposal is inspired by ant colony optimization. Subsection 2.2 provides a background to ACO, and then Section 3 describes the approach.

2.2 Fundamentals of Ant Colony Optimization

Ant Colony Optimization [DOR92] is a bio-inspired global search metaheuristic that belongs to the set of swarm intelligence methods [DBS06], and it is used to solve combinatorial optimization problems defined as (S, Ω, e), where S is a search space defined over a finite set of discrete decision variables $U = \{U_1, U_2, ..., U_N\}$, Ω is a set constraints defined over U, and $e : U_1 \times U_2 \times ... \times U_N \rightarrow \mathbb{R}_{\geq 0}$ is a loss function to be minimized. It is said that a solution $s \in S$ is feasible if all variables $U_i \in s$ have been assigned values from their domain, and they satisfy the constraints in Ω. An optimal feasible solution to the problem $s^* \in S$ verifies that $e(s^*) \leq e(s_i) \forall s_i \in S : s_i$ is feasible.

The ACO design methodology is based on the problem formulation as a graph traverse over a *construction graph* $G = (V, E)$ that represents the search space S, where V stands for the graph vertices and E for the edges. A solution $s \in S$ is incrementally built from a selected starting node of the graph. Thus, traversing the graph performs the assignment of values to variables $U_i \in s$ until the solution s is constructed. This is a simulation of the real behaviour of an ant that departs from the nest to the food. Each time the ant traverses an edge of the graph (i.e., a value to a variable $U_i \in s$ has been assigned), pheromone is released to mark the edge for other members of the colony that will perform another graph traverse in the future. The literature offers a plethora of Ant Colony Optimization approaches whose algorithm components differ from each other, as for instance the way an ant chooses the path, the way pheromone is released and evaporated, parallel algorithm approaches, etc. We refer the reader to [MB12] for a survey on ACO methods.

The first Ant Colony Optimization method was proposed in [DOR92], and it is known as the *Ant System* optimization. The *Ant System* used a single ant to solve the problem. Nowadays, *Ant Colony Optimization* refers to a variation of this approach where not a single ant, but a population of ants, are deployed together over the construction graph to find the best solution to the problem addressed, and uses heuristic values that encode expert information about the problem instance definition in order to speed up the search procedure. Classically, the heuristic information that defines the construction path is encoded as α and β values in a formula, as for

instance Formula 2.6. We refer the reader to the work [DS04] for a more detailed explanation of the algorithm's component design.

$$p_{jk}^{i}(t) = \begin{cases} l \dfrac{\left[\tau_{jk}(t)\right]^{\alpha}\left[\nu_{jk}(t)\right]^{\beta}}{\sum_{l \in N_j^i(t)}\left[\tau_{jl}(t)\right]^{\alpha}\left[\nu_{jl}(t)\right]^{\beta}} & if\ k \in N_j^i(t) \\ 0 & if\ k \notin N_j^i(t) \end{cases} \tag{2.6}$$

As was previously mentioned, ACO has been applied for symbolic regression and automatic program generation in previous works with promising results. The work in [BC02b] proposed ant colony programming, to solve symbolic regression problems where the construction graph is built from a predefined set of rules. Reference [GWJ04] also shows an approach to solve symbolic regression problems, where the construction graph is built over a fully-connected graph of operators and operands. Later, the work [ORV10] suggested using the ACO approach to evolve grammar structures to find classification rules in data mining problems. The Enhanced Generalized Ant Programming (EGAP) was proposed in [SW08], to solve tree symbolic regression using tree-based grammar representation. In [SW09], GP is compared with the EGAP approach, concluding that GP statistically improves EGAP in the problems addressed.

In this work, we use ACO to search for SLPs with a balance between accuracy and size. Finding SLPs with reduced size is a topic that has been addressed before in reference [Alo+12], which offers an approach to improve accuracy of SLPs for symbolic regression problems in the presence of noisy data, using model regularization. The experimental section of this article compares our approach with the procedure mentioned as a baseline method. *Dynamic Ant Programming* (DAP) [SON11], another ACO-based approach developed to tackle the bloat problem under the assumption of tree representation of algebraic expressions, will also be included in the experimental section as a baseline method for comparison.

3 Ant Colony Optimization for Straight Line Programs

3.1 Design of the construction graph

As it is mentioned in previous sections, a SLP can be represented as a DAG. The DAG is obtained by means of a simple procedure applied over the SLP grammar rules. Starting from rule U_N, the starting node is created and labelled as U_N, and assigned with the operator of rule U_N. One or two nodes are then created, depending on the arity of the operator, and linked to U_N. If the first (or second, respectively) operand is a terminal symbol, then the node is assigned with the value of the terminal symbol. Otherwise, this procedure is applied recursively over the generated nodes until terminal symbols are reached.

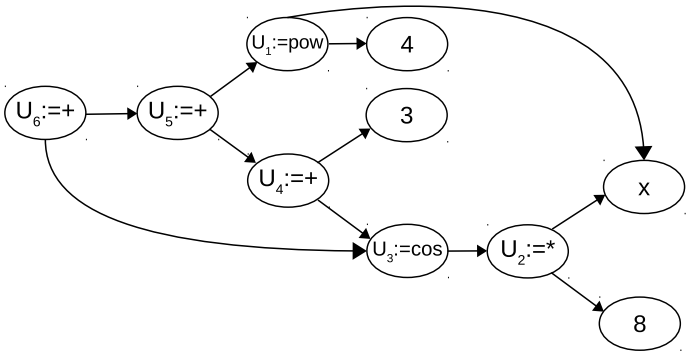


Figura 2.1: Example of the corresponding DAG for the SLP example of Figure 2.5.

As an example of this procedure, Figure 2.1 shows the resulting DAG for the SLP of the Formula 2.5.

In our approach, since SLPs are grammar-based representations of algebraic expressions, the problem of finding the algebraic expression $\tilde{f}$ that accurately fits output data $\vec{y}$ from a set of input data $\vec{x}$, can be formulated as finding the correct grammar production rules that build a valid SLP. Then, given a maximum number of allowed rules (maximum size N), our approach attempts to find the minimum number of production rules that build a valid SLP (which generates only an algebraic expression), that minimize a loss function $e(\tilde{f}(\vec{x}, \vec{w}), \vec{y})$. Besides, the grammar representation can be translated into a DAG, and this fact suggests that the problem

can also be formulated as a graph traverse problem. Thus, the combinatorial problem (S, Ω, e) to be solved in our research work assumes that S is the space of straight line grammar rules with a maximum number of rules equals N, Ω are the constraints of the grammar rules, and e is an error measurement that evaluates the accuracy of an SLP to approximate the desired output data $\vec{y}$.

The construction graph used for ACO algorithms in our research is therefore designed as follows: The starting node is the grammar rule whose antecedent is the non-terminal symbol U_N. The feasible neighborhood of a node U_i is the set of arithmetic operators allowed for building algebraic expressions, $\{O_1, O_2, ..., O_l\}$. Let O_{U_i} be the selected operator for rule U_i. Then, its feasible neighborhood is the set of available operands $\{x_1, x_2, ..., x_n, w_1, w_2, ..., w_k,$
$U_{i-1}, U_{i-2}, ..., U_1\}$, where $x_j, 1 \leq j \leq n$, is the j-th input variable, $w_j, 1 \leq j \leq k$, is the j-th algebraic expression parameter, and $U_j, 1 \leq j < i$, is a non-terminal symbol that makes reference to grammar production rule U_j. Once the first operand has been chosen, then the second operand is selected using the same feasible neighborhood as the one used for the first operand, if O_{U_i} is a binary operator. The feasible neighborhood of the last generated operand is the node corresponding to rule U_{i-1}, and then the process is repeated until the operators and operands of rule U_1 have been generated. This graph representation allows us to store the pheromone trails into 3 matrices:

- Matrix $T_o(1..N, 1..l)$, to store pheromone trails from nodes of rules U_i to operator nodes. $T_o(i, j)$ contains the pheromone regarding the selection of operator O_j at rule U_i.
- Matrices $T_{R_1}(1..N, 1..l, 1..n + k + N - 1)$, and $T_{R_2}(1..N, 1..l, 1..n + k + N - 1)$, to store pheromone trails from operator nodes to the first and second operand nodes, respectively. The value of $T_{R_k}(i, p, j)$ contains the pheromone regarding the selection of symbol j as the value for operand k in the rule U_i, when operator O_p was selected for the rule. Symbol j links to an input variable x_j if $1 \leq j \leq n$, to an algebraic expression parameter w_{j-n} if $n < j \leq n + k$, and to rule U_{j-n-k} if $n + k < j \leq n + k + l$. To ensure verification of the constraints Ω of the combinatorial problem stated, the values $T_{R_k}(i, j) = 0, \forall j : n < j < i + n + k$, i.e. a rule U_i can only contain rule references in the set $\{U_{i-1}, U_{i-2}, ..., U_1\}$.

Figure 2.2a outlines the general design of the construction graph described. As an example, Figure 2.2b assumes a set of operators $O = \{O_1, O_2, O_3, O_4, O_5\} = \{+, -, *, /, cos\}$, a set of variables $\vec{x} = \{x\}$, a set of algebraic expression parameters $\vec{w} = \{w_1, w_2\} = \{3, 4\}$, and a maximum set of grammar rules $N = 3$, and describes an example of the graph traverse to obtain the grammar rules $U_3 := U_2 * U_2$, $U_2 := w_1/U_1$, $U_1 := cos(x)$. Firstly, rule U_3 is generated. The operator $*$ is probabilistically selected, and then the first and second operands are generated. In this case, non-terminal symbol U_2 is selected probabilistically for both operands. After rule U_3 is completed, then rule U_2 is generated. The probabilistically selected operator is /, and w_1 and U_1 as the first and second operands, respectively. Finally, rule U_1 is generated, probabilistically choosing the operator *cos*. Since the operator *cos* requires a single operand, then only the first operand is generated probabilistically. In this case, terminal symbol x is selected. The construction of the solution is completed, and the algebraic expression encoded into the generated SLP is $(3/cos(x)) * (3/cos(x))$.

Once the construction graph is designed, the next section describes the components of the algorithm.

3.2 Algorithm design

In this section, we design an ACO-based algorithm to find algebraic expressions $\tilde{f}(\vec{x}, \vec{w})$ encoded as SLPs, with a local search procedure to optimize the algebraic expression parameters $\vec{w}$ simultaneously. The literature offers different methods to solve the problem of parameter $\vec{w}$ estimation, such as [MZ90; MK89]. Other proposals reduce the problem complexity and select a fixed value for parameters $\vec{w}$ from the beginning [AB00], although this strategy could shrink the search to less accurate solutions. In our approach, a non-linear least-square method (NLS) [Mar63; Kom+13] that minimizes a loss function, is used during the evaluation of each candidate solution $\tilde{f}$ to fit the numerical values for the parameters $\vec{w}$. The loss function used in this work is shown in equation 2.7, where n stands for the number of data samples, $\vec{x}(i)$ is the i-th input sample, and $y(i)$ is the i-th output sample. Using this fitness measure, the optimal solution

$\tilde{f}^*$ has $fitness(\tilde{f}^*) = 1$, while the worst solutions have a fitness value closer to 0. Then, the problem is formulated as a maximization problem.

$$fitness(\tilde{f}) = \frac{1}{1 + \frac{1}{n}\sum_{i=1}^{n}(\tilde{f}(\vec{x}(i), \vec{w}) - y(i))^2} \tag{2.7}$$

The adaptation of classical ACO implementation [DOR92] to our proposal is shown in Algorithm 3. The procedure starts by initializing the pheromone matrices to an initial value, which is found experimentally by means of a trial-and-error process. Also the best solution, named $BestAnt$, is assigned with an empty value. After initialization, the main algorithm repeats until a stopping criterion is satisfied. In this article, the stopping criterion used in the experimentation is to achieve a number of feasible solutions evaluated.

Each algorithm iteration comprises the following steps: *Solution construction*, *Local search*, *Solution evaluation* and *Pheromone update*:

- The step **Solution construction** builds the path for each ant in the algorithm, following the graph traverse process over the construction graph described in Section 3.1. Unlike classic Ant Colony methods, which use α and β values in equation 2.6 to define a balance between pheromone trails and heuristic criteria to explore the search space, our model does not rely on heuristic information to build the ant solution. This is because it is difficult to find an appropriate heuristic for operators and operands in symbolic regression, since the fitness of an algebraic expression also depends on the other subexpressions in the solution, and the heuristic of an operator could fail when used in different contexts. For this reason, we set the values $\alpha = 1$ and $\beta = 0$ in our approach. This decision is not uncommon in symbolic regression problems, and it was assumed in previous works as in [SON11]. The selection of operators and operands is performed probabilistically, using the formula in equation 2.8, where $N_i(t)$ is the feasible neighborhood of vertex i at algorithm iteration t, $\tau_{ij}(t)$ is the pheromone trail from vertex i to vertex j at iteration t,

and $p_{ij}(t)$ is the probability of moving the ant from vertex i to vertex j in the construction graph. This probability is calculated depending on the type of vertex i on which the ant is located in the construction graph. On one hand, if the ant is located on a vertex corresponding to a non-terminal symbol $U_i \in \{U_1, ...U_N\}$, then $\tau_{ij}(t) = T_o(i, j)$, and the feasible neighborhood of vertex i is the set of operators $N_i(t) = \{O_1, O_2, ..., O_l\}$. On the other hand, if the ant is located at operator p of production rule r, then $\tau_{i,j} = T_{R1}(r, p, j)$ and the feasible neighborhood is $\{x_1, x_2, ..., x_n, w_1, w_2, ..., w_k, U_1, U_2, ..., U_{r-1}\}$. In case the operator p selected for rule r is binary, and the ant is located in vertex i associated with a symbol for the first operand, then $\tau_{i,j} = T_{R2}(r, p, j)$, and the second operand is selected. Otherwise, the remaining possible vertices do not fulfill the constraints in Ω and their selection probability is 0.

$$p_{ij}(t) = \begin{cases} \dfrac{\tau_{ij}(t)}{\sum_{p \in N_i(t)} \tau_{il}(t)} & if\ j \in N_i(t) \\ 0 & if\ j \notin N_i(t) \end{cases} \tag{2.8}$$

- The **Local search** performs the algebraic expression parameters $\vec{w}$ optimization, using a non-linear least-square method (NLS) [Mar63; Kom+13].

- The **Solution evaluation** process calculates the fitness for each feasible solution found by ants in the current iteration. A solution evaluation is performed as follows: For each i-th input data sample $\vec{x}(i)$, all rules in the solution from U_1 to U_N are evaluated in ascending order, until U_N is reached. Then $\tilde{f}(\vec{x}(i), \vec{w})$ is assigned with the resulting value of the best rule of the SLP. The fitness is calculated using all $\tilde{f}(\vec{x}(i), \vec{w})$ values according to the formula in equation 2.7.

- The step **Pheromone update** is applied not only to control the amount of pheromone that an ant deposits on the path, but also the pheromone evaporation. Formula 2.9 shows that the pheromone evaporation rate is controlled using an algorithm parameter $\rho \in [0, 1]$, which reduces the quantity of pheromone proportionally at each iteration.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta fitness_{ij}(BestAnt) \tag{2.9}$$

In our approach, only the best ant that was found during the search deposits pheromone in proportion to its fitness, as is performed in the Best-Worst ACO approach [CVH02]. The deposition rate is also controlled by an algorithm parameter Δ. The value $fitness_{ij}(BestAnt) = fitness(BestAnt)$ if the edge E_{V_i,V_j} is included in the path over the construction graph obtained by solution $BestAnt$ and E_{V_i,V_j} is not part of a rule classified as *dead code*. The value $fitness_{ij}(BestAnt) = 0$ otherwise. With the term *dead code* we mean all the production rules that are encoded into the SLP solution provided by an ant, but which cannot be derived from U_N. Since these production rules are not used to generate the algebraic expression encoded into the solution, then pheromone deposition is also avoided for these production rules. Formula 2.10 shows an example of SLP with size $N = 3$, where the production rule U_2 is *dead code*, since $U_3 \Rightarrow U_1 * U_1 \Rightarrow (x - w_2) * (x - w_2)$, and U_2 cannot be derived from U_3.

$$\begin{aligned} U_1 &\rightarrow x - w_2 \\ U_2 &\rightarrow U_1 + w_1 \\ U_3 &\rightarrow U_1 * U_1 \end{aligned} \tag{2.10}$$

Finally, with regards to the computational complexity of our proposal, we remark that each solution construction, solution evaluation and pheromone update methods are $O(n)$, where n is the size of the SLP. Then, although the time complexity of the local search procedure is exponential, it is executed under a set of predefined number of iterations, which implies a constant time complexity, as is shown in the experimental section. Consequently, the time complexity of our proposal is $O(n^3 * m)$, where n is the size of the SLP and m is the number of ants. Once the proposed SLP-ACO algorithm has been described, the next section performs an

experimental study in a real data scenario.

4 Experiments

Since the final goal of our research is to develop data mining techniques aimed at finding energy consumption models that encompass a balance between accuracy and interpretability, the proposal SLP-ACO is validated using a set of real energy consumption data. In order to prove the potential of our proposal, we used ACO and GA baseline methods to compare the results in terms of not only accuracy but also expression size. Firstly, we have selected an ACO algorithm used to solve symbolic regression problems as baseline method for comparison. More specifically, we used Dynamic Ant Programming (DAP) [SON11] which uses a tree representation to encode algebraic expressions. For this comparison, we are motivated to study the potential of SLPs over trees and also to verify if the local search method used in SLP-ACO for parameter estimation allows to perform accurate solutions of reduced size. On the other hand, we also compare our proposal with genetic programming algorithms [RRu+17]. We have used two genetic programming approaches: the first one uses a local search method for parameter estimation and it is compared with SLP-ACO; the second one does not include a parameter estimator and it is compared with DAP.

In order to clarify the comparison carried out in this section, we named each approach as follow: **DAP** for Dynamic Ant Programming; **SLP-GA** for Genetic Algorithm without using parameter estimation; **SLP-GA-Cte** which uses a local search method for parameter estimation and **SLP-ACO** for our proposal. These algorithms help us to cover a wide variety of proposals that focus on different features regarding our approach -representation, strategies to address the bloat problem, and training models-. Therefore, the main goal of this experimentation attempts to verify the quality of the results provided by each algorithm and study the advantages and limitations of our proposal.

4.1 Application to real scenarios

The real scenario to test our approach is an energy consumption modelling problem that attempts to obtain interpretable and accurate models of energy consumption in public buildings. More specifically, we use a dataset containing the energy consumption of four buildings at the University of Granada, measured hourly in kW/h from March 2013 to October 2015. In order to acquire the energy consumption data, each building is equipped with a Building Automation System (BAS) [Sal05] that retrieves the energy consumption data from sensors and stores the values with their timestamp in a database. The raw energy consumption data series for each building were preprocessed and aggregated to obtain a daily consumption data series, which we use as a starting point in this experimentation. The preprocessing also included filling in missing values due to power cuts, sensor malfunctioning and maintenance tasks, etc. Figure 2.3 shows the raw aggregated data series for the four buildings. Finally, to work with uniform data, the data were normalized in the interval [0.0 1.0] (see equation 2.11, where v_i is the response value, v_{max} is the maximum response observed, v_{min} is the minimum response observed and $r_{normalized}$ is the normalized response). For confidentiality reasons, we are not allowed to provide the data, and the buildings are labelled as B_1, B_2, B_3, B_4, and contain two research centers, a large faculty, and a small faculty.

$$r_{normalized} = \frac{v_i - v_{min}}{v_{max} - v_{min}} \tag{2.11}$$

The modelling problem that we tackle attempts to explain the relationships on energy consumption data between working days in the same week. Our goal is to provide an interpretable model that can accurately estimate the energy consumption of a working day considering the remaining working days in the same week. The expected outcomes are models of energy consumption which aid understanding of how the energy consumption of different days relates to each other, in order to include these models in other high-level tasks such as anomaly detection and forecasting, for future research. Assuming we name the energy consumption of the working days as d_1, d_2, d_3, d_4, d_5, equation 2.12 shows that we want to approximate the energy consumption of

day i considering the remaining days j_1, j_2, j_3, j_4, where $j_k \neq i \forall k$, and $\vec{w}$ and f are unknown. For this reason, each energy consumption data series initially had 650 values, and was transformed into a multivariate data series with 5 dimensions (one per each working day), with 130 samples (one sample per week).

$$d_i = f(d_{j_1}, d_{j_2}, d_{j_3}, d_{j_4}, \vec{w}) \tag{2.12}$$

There are 20 experiments, to estimate energy consumption of Mondays, Tuesdays, Wednesdays, Thursdays, and Fridays separately, considering the energy consumption of the remaining days in the week as input data, for buildings B_1 to B_4. A preliminary visual and statistical study was first performed, in order to know if there was a correlation between the energy consumption of the working days. Figure 2.4 shows the correlation matrices for all buildings and working days. The diagonal plots of the figures show the histogram of the energy consumption for each working day, and each cell (row i, column j) shows the correlation of day j to day i. Finally, the text in red in the correlation plots shows the correlation coefficient R for the two days being compared. We observe that, as could be expected, there is a high correlation ($R \geq 0{,}7$) between the energy consumption of two working days in many cases, although there are some cases with an intermediate correlation ($0{,}3 \leq R < 0{,}7$). This fact suggests that symbolic regression could be applied to obtain accurate estimation models in the energy consumption modelling problem addressed.

To study generalization capabilities, all datasets were divided into training (first 70 % of data) and test (last 30 % of data). After that, the training set was used to calculate the fitness value of each method, and the test set was applied over the solution returned from each algorithm execution in order to obtain the results analyzed in this section. For the experimentation, we performed a preliminary extensive experimentation to tune the parameters of both Genetic Algorithms (SLP-GA and SLP-GA-Cte) and Ant Colony approaches (DAP and SLP-ACO). Then, the parameters tuned for both SLP-GA are: 80 % of crossover probability and 20 % for mutation probability, the population size were established at 70. After that, the experimental configuration for SLP-ACO and DAP are: the minimum value of pheromone rate (ρ_{min}) has been set to 0.01, the evaporation

rate (p) were established to 0.5; the number of ants used were 70 and the pheromone value of an inserted node (ρ_{ins}) in DAP is 1. In addition, we allowed a total of 7 parameters ($w_1, w_2, ..., w_7$) for each approach. Moreover, whereas both SLP-GA and DAP have a set of predefined values for each parameter ($w_1 = 1, w_2 = 2, ..., w_7 = 7$), a local search method is used to estimate parameter values for each SLP-GA-Cte and SLP-ACO. Besides, the mathematical operators allowed for all approaches are $\{+, -, *, /, exp, sin, cos, pow, min, max, tan, tanh\}$; the maximum SLP/Tree size are 32 and the stopping criteria are 10000 evaluations. Finally, we performed 30 executions of each algorithms so that we could analyze the results statistically.

Table 2.1 gathers the results obtained for each algorithm over the test data. Column 1 shows the target working day whose energy consumption is estimated. Then, Columns from 2 to 21 describe the median, best and worst fitness, the average execution time in seconds, and the average size of the solutions provided by DAP, SLP-GA, SLP-GA-Cte and SLP-ACO, respectively. Fitness value are calculated as is described in equation 2.7 and the size of an algebraic expression is calculated as the number of operators it contains, i.e. the number of non-leaf nodes in tree representation and the number of valid rules in SLP. Moreover, in order to compare the results of each algorithm in terms of fitness and algebraic expression size, we used a statistical test. Due to the results performed by each algorithm does not come from a normal distribution, we decided to use a non parametric test. Consequently, Columns 22 to 25 plots the results of the Kruskal-Wallis (KW) statistical test with a 95 % confidence level, to compare each method in terms of fitness values, and Columns 26 to 29 show the solutions regarding the algebraic expression size. The KW test was applied as follows: For each experimentation, the algorithms were sorted from best median fitness/size to worst median fitness/size. A paired KW was applied over the two first algorithms. If significant differences were found ($p\text{-}value < 0{,}05$), then the algorithm with the best fitness/size was marked with tag 1, and the other one with tag 2, and then the comparison continues with the next algorithm with the best fitness/size. Otherwise, both algorithms were tagged with 1, and the comparison is performed between the algorithm with best median fitness/size and the third algorithm with best median fitness/size. This procedure is applied for all the remaining algorithms results for each problem, until all algorithms have been compared. Finally, for a better analysis of the results in Table 2.1, we have

included the boxplots of the error distribution of all experiments in Figure 2.6. Each picture contains the boxplots of the error measure for the algorithms being compared: DAP, SLP-GA, SLP-GA-Cte and SLP-ACO, for the same building and working day.

In order to compare baseline methods, the analysis starts by comparing DAP and SLP-GA. Thus, we may observe in Table 2.1 that SLP-GA achieved better solutions in terms of median values in all cases, whereas DAP performed the worst solution in all problems. With regards of the best fitness, SLP-GA achieved the best solution in 5 experiments, DAP did it in 2 cases and similar solutions were achieved in the remaining 13 experiments. From this analysis we may conclude that SLP-GA is potentially better than DAP, which is supported by the KW test, where SLP-GA achieved better solutions in all cases (shown in columns 22 and 23). The worst solutions provided by DAP may be consequence of the tree representation used to encode algebraic expression and also the local search procedure used by SLP-GA, which may help to avoid local optima and perform better solutions.

On the other hand, if we compare ACO methods (DAP and SLP-ACO) we may observe that SLP-ACO was able to find better solutions in terms of median fitness in all cases, whereas DAP achieved the worst solution in all experiments. Moreover, with regards to the best fitness, SLP-ACO achieved the best solutions in 6 of 20 problems and DAP did it in 1 experiment. These results help us to conclude that the SLP proposal may improve the search of the best algebraic expression, which is supported by KW test in Columns 22 and 25 of Table 2.1 where we may observe that SLP-ACO performed better solutions than DAP in all experiments. The analysis continues by comparing SLP-ACO and SLP-GA-Cte. Firstly, regarding median fitness, the reader may observe that SLP-ACO was able to achieve the best solution in 1 problem, whereas both approaches performed similar solutions in the remaining 19 experiments. Regarding the best fitness, SLP-GA-Cte found the best solution in 1 experiment and similar solutions were achieved in the remaining 19 experiments. With regards to the worst fitness, SLP-GA-Cte performed worse solutions in 6 problems and SLP-ACO did it in 3 cases. Finally, regarding the KW test we may conclude that SLP-ACO performed better solutions in 4 experiments, SLP-GA-Cte achieved better results in also 4 problems and significant differences were not found in the remaining 12 problems.

From this first analysis we may conclude that SLP approaches are able to find more accurate solutions than tree approaches. Moreover, if we compare SLP-GA-Cte and SLP-ACO approaches, we cannot conclude which approach is better in terms of fitness. On the other hand, regarding the algebraic expression size, we can conclude that ACO approaches are able to find shorter algebraic expressions with high accuracy. To give support to this conclusion, we may observe the results of the Kruskal-Wallis test in columns 26-27 and 28-29 of Table 2.1. Firstly, regarding the algebraic expression size of DAP and SLP-GA (columns 26 and 27, respectively), we conclude that DAP performed shorter algebraic expressions in 15 of 20 experiments, whereas SLP-GA achieved shorter algebraic expressions in 5 problems. After that, comparing the results of the algebraic expressions found by SLP-GA-Cte and SLP-ACO (columns 28 and 29, respectively), we may confirm that SLP-ACO achieved shorter solutions in all cases. Nevertheless, if we compare SLP-ACO with SLP-GA we may verify that SLP-GA achieved shorter solutions in 9 cases. In contrast, regarding fitness accuracy, the KW test concludes that SLP-ACO was able to perform better solutions in 13 problems. In this way, we want to highlight the main goal of this research, which attempts to find a balance between accuracy and interpretability. Therefore, we may conclude that SLP-ACO was able to find shorter algebraic expressions with potential accuracy.

With regards to the execution time, we may conclude that ACO approaches need more computational time to find a solution. This fact may be verified in the execution time between DAP vs SLP-GA and SLP-ACO vs SLP-GA-Cte, where both ACO methods need by means two times more than GA approaches to perform a solution. Besides, we want to remark that the local search used in SLP-GA-Cte and SLP-ACO introduces a time overhead of almost 200 % regarding the execution time of both methods.

Finally, equations 2.13 to 2.16 show an example of the most accurate algebraic expressions found by DAP, SLP-GA, SLP-GA-Cte and SLP-ACO, respectively, to approximate Thursday's energy consumption of building B_4. In these equations, we use the notation shown in equation 2.12, where d_1, d_2, d_3, d_4, d_5 stand for the energy consumption of Mondays, Tuesdays, Wednesdays, Thursdays, and Fridays, respectively. As we observe, DAP and SLP-GA return simpler algebraic expressions, following by SLP-ACO and SLP-GA-Cte. Nevertheless, although all approaches seem to perform similar fitness, the statistical tests show that SLP-ACO and SLP-GA-Cte

were able to achieve better results, but SLP-ACO reached a simpler algebraic expression. As an example of the equation provided by SLP-ACO, an expert could conclude that Thursday's energy consumption can be explained as the combination of the energy consumption of Monday's, Wednesday's and Friday's. The interpretability of this type of algebraic expression could therefore contribute to a better data analysis in higher level decision-making processes. Regarding the accuracy of all methods, Figure 2.5 shows that the approximation of the whole data series with the algebraic expressions provided by each method fits the real data correctly, and this fact suggests that SLP-ACO is a promising technique to be used for obtaining a suitable balance between accuracy and solution complexity.

$$d_4^{DAP} = max((log(min(2, d_5)) + 1), d_3) \tag{2.13}$$

$$d_4^{SLP-GA} = \frac{d_3/1}{exp(d_3 - d_5)} \tag{2.14}$$

$$d_4^{SLP-GA-Cte} = 0{,}63 * min(((d_3 + min((((d_3 + (d_5 + (d_3 * -0{,}95)))^{1{,}07}) * 0{,}6), (1{,}44^{d_5})))^{1{,}07}), (1{,}44^{d_5})) \tag{2.15}$$

$$d_4^{SLP-ACO} = (\frac{1{,}03^{d_1}}{exp(1{,}38)} + \frac{1{,}03^{d_1}}{exp(1{,}38)}) * ((d_5 + d_3) - log(1{,}01)) \tag{2.16}$$

From the aforementioned analysis, we conclude that both SLP-GA-Cte and SLP-ACO provided promising results regarding accuracy in the real symbolic regression problems addressed. On the other hand, SLP-ACO was able to provide solutions with a lower size in all cases, at a cost of increasing the computational time substantially. These lower-size solutions could be more interpretable by an expert, and therefore more suitable for use in higher-level decision making processes than SLP-GA's solutions.

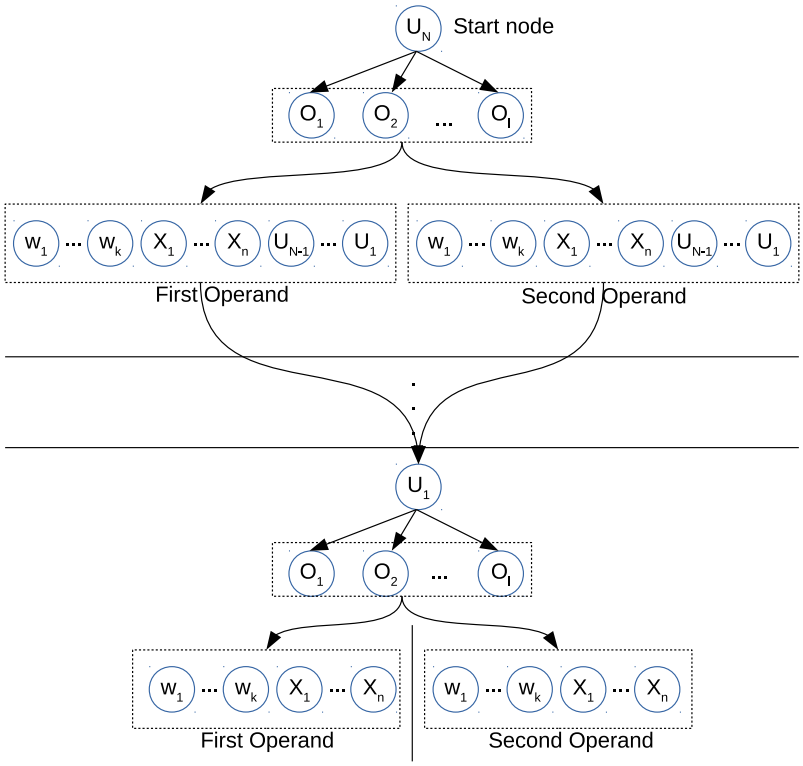
5 Conclusions

In this paper, we have introduced a new algorithm based on Ant Colony Optimization for symbolic regression using Straight Line Programs (SLPs). The approach has been compared with state-of-the-art algorithms with different algebraic representation schemes, and also targeted at minimizing the size of the resulting solutions. The approach has been tested in real enegy consumption data. Regarding accuracy, SLP-based algorithms obtained promising results in the problems studied. The linear representation of SLPs allows us to perform a better search over the solution space of algebraic expressions, and also time complexity is reduced when SLPs are trained with genetic programming, compared to tree-based representation schemes. We have also included a local search to fit the resulting algebraic expression parameters inside GA and ACO algorithms. Our experiments show that time complexity is substantially increased using this strategy, but also that accuracy of the resulting solutions can be improved. Regarding the size of the resulting algebraic expressions, ACO based methods provided smaller algebraic expressions than GA approaches. More specifically, DAP was able to find smaller solutions in 15 of 20 problems compared to SLP-GA and SLP-ACO achieved shorter algebraic expression in all experiments, compared to SLP-GA-Cte.

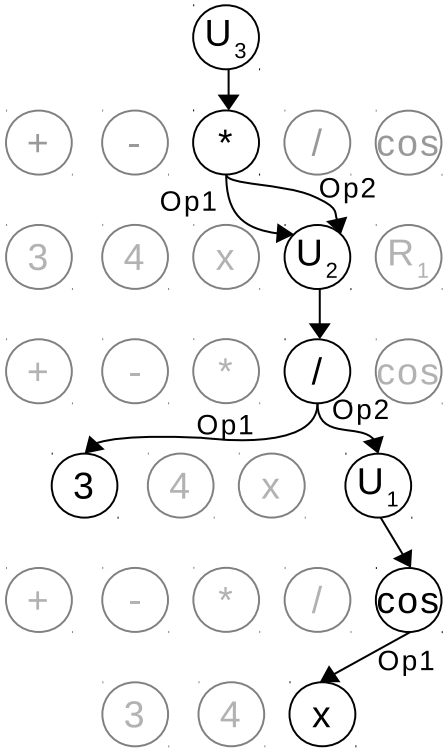
As a general conclusion, the SLP-ACO method proposed in this article has helped to maintain a balance between accuracy and complexity of the solutions provided, and has been tested successfully in real scenarios. Simpler and accurate solutions were obtained using this method, which can help to facilitate a better expert analysis in higher-level decision making processes.

Acknowledgements

This work has been supported by the project TIN201564776-C3-1-R.



(a) Scheme of the design of the construction graph for SLP search using ACO



(b) Example of traversing the creation graph

Figura 2.2: Scheme and example of the construction graph for SLP search using ACO

Algorithm 3 SLP-ACO algorithm

Require: Input: S_p, the number of ants that run in parallel
Require: Input: $\vec{x} = (x_1, x_2, ..., x_n)$, input data variables for SLP evaluation
Require: Input: $O = \{O_1, O_2, ..., O_l\}$, the operators allowed for the algebraic expression
Require: Input: K, the maximum number of algebraic expression parameters $\{w_1, ...w_k\}$
Require: Input: $\vec{y} = (y_1)$, output data for SLP evaluation
Ensure: Output: $SLP(1...N)$ a sequence of rules that encode the algebraic expression

```
{Initialization}
Set initial operator pheromone T_o(i,j) := T_0, ∀i,j : 1 ≤ i ≤ N, 1 ≤ j ≤ l
Set initial operands pheromone T_R1(i,p,j) = T_0, T_R2(i,p,j) := T_0, ∀i,p,j : 1 ≤ i ≤ N, 1 ≤
p ≤ l, 1 ≤ j ≤ n + k + l
BestAnt := {∅}
t := 1 {Current iteration}
{Main loop}
while No stopping criterion is fulfilled do
  for counter a=1 to S_p do
    {Solution construction}
    Initialize ant_a, the a-th ant
    for i:=N downto 1 do
      Select operator and operands for rule ant_a(i) according to equation 2.8
    end for
    {Local search}
    w⃗ := NLS(x⃗, y⃗, ant_a)
    {Solution evaluation}
    Evaluate(ant_a, (w)⃗, x⃗, y⃗)
    {Update best solution}
    if fitness(ant_a) > fitness(BestAnt) then
      BestAnt := ant_a
    end if
  end for
  {Pheromone update}
  Perform pheromone update according to equation 2.9
  Update next iteration t := t + 1
end while
return BestAnt
```

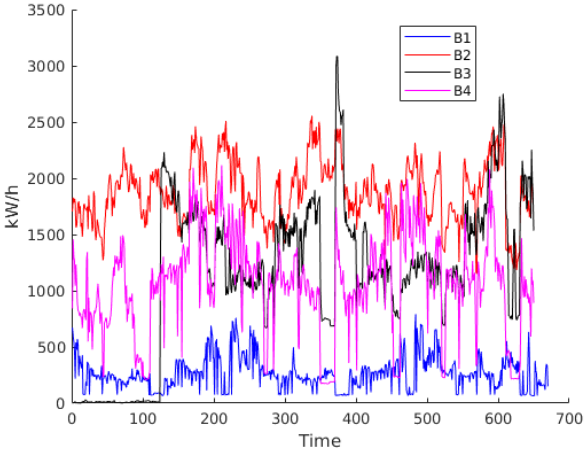


Figura 2.3: Energy consumption data series for buildings B_1, B_2, B_3, B_4.

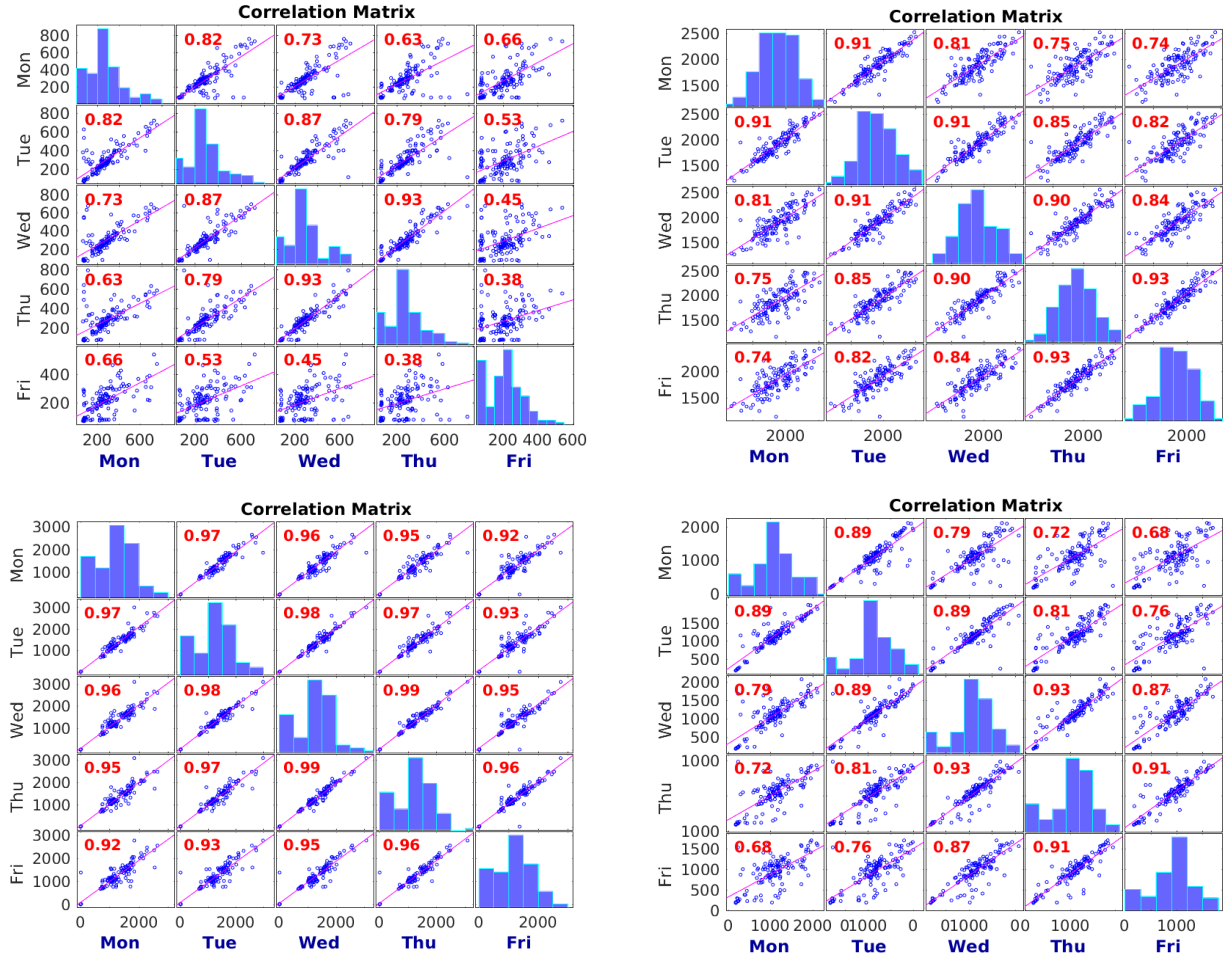


Figura 2.4: Correlation matrices of energy consumption for buildings B_1 to B_4, from Monday to Friday

Working Day	DAP					SLP-GA					SLP-GA-Cte					SLP-ACO					Fitness Test			Size Test		
	Median	Best	Worst	Time	Size	Median	Best	Worst	Time	Size	Median	Best	Worst	Time	Size	Median	Best	Worst	Time	Size						
Building B1																										
Monday	0.42	0.97	0	13.9	5.9	0.98	0.99	0.98	6	3.93	0.99	0.99	0.97	100.14	11.57	0.99	0.99	0.97	174.29	6.07	3	2	1	2	2	1	4	3
Tuesday	0.32	0.99	0	13.78	4.2	0.98	0.98	0.97	6.02	5.66	0.98	0.99	0.98	93.45	8.00	0.98	0.98	0.98	147.73	3.60	3	2	1	1	2	3	4	1
Wednesday	0.49	0.99	0	13.73	5.43	0.99	0.99	0.99	5.89	4.83	0.99	0.99	0.97	96.96	10.97	0.99	0.99	0.99	175.30	4.83	4	2	1	3	2	1	3	1
Thursday	0.49	0.98	$1 * 10^{-3}$	13.309	5.7	0.99	0.99	0.69	5.44	4.36	0.99	0.99	0.94	103.84	10.73	0.99	0.99	0.96	160.68	4.40	2	1	1	1	3	1	4	2
Friday	0.75	0.96	0	13.33	7.36	0.97	0.98	0.97	5.48	5.43	0.98	0.98	0.97	103.04	11.97	0.98	0.98	0.94	176.02	6.30	3	2	1	1	3	1	4	2
Building B2																										
Monday	0.48	0.99	0	14.65	4.8	0.99	0.99	0.99	5.81	6.36	0.99	0.99	0.97	93.91	10.43	0.99	0.99	0.99	165.08	4.70	2	1	1	1	2	3	4	1
Tuesday	0.39	0.99	0	14.4	3.63	0.99	0.99	0.99	5.65	4.93	0.99	0.99	0.98	89.85	9.53	0.99	0.99	0.99	163.35	5.13	3	2	1	1	1	2	4	3
Wednesday	0.15	0.98	0	14.67	4.6	0.99	0.99	0.99	5.86	5.76	0.99	0.99	0.74	96.60	11.37	0.99	0.99	0.99	164.34	4.20	4	3	2	1	2	3	4	1
Thursday	0.83	0.99	0	14.3	4.03	0.99	0.99	0.99	6.21	4.36	0.99	0.99	0.97	89.41	9.33	0.99	0.99	0.98	157.48	3.57	3	2	1	1	2	3	4	1
Friday	0.9	0.99	0	14.3	5.46	0.99	0.99	0.99	6.19	5.6	0.99	0.99	0.99	91.01	8.20	0.99	0.99	0.98	166.70	5.23	3	2	1	1	2	3	4	1
Building B3																										
Monday	0.46	0.99	0	14.63	5.06	0.99	0.99	0.99	5.49	5.13	0.99	0.99	0.99	85.93	10.60	0.99	0.99	0.99	179.88	4.73	2	1	1	1	2	3	4	1
Tuesday	0.3	0.99	0	13.76	4.63	0.99	0.99	0.99	5.49	4.93	0.99	0.99	0.99	79.09	8.90	0.99	0.99	0.99	157.04	3.93	4	3	2	1	2	3	4	1
Wednesday	0.73	0.99	0	13.83	4.9	0.99	0.99	0.99	5.48	5.03	0.99	0.99	0.99	76.81	7.67	0.99	0.99	0.99	167.11	5.30	2	1	1	1	1	2	4	3
Thursday	0.45	0.99	0	12.96	3.73	0.99	0.99	0.99	5.51	5.43	0.99	0.99	0.99	80.93	9.53	0.99	0.99	0.99	157.58	5.93	3	2	1	1	1	2	4	3
Friday	0.32	0.99	0	13.69	4.93	0.99	0.99	0.99	5.46	6.03	0.99	0.99	0.99	84.68	10.53	0.99	0.99	0.99	173.40	6.37	3	2	1	1	1	2	4	3
Building B4																										
Monday	0.44	0.98	0	13.17	5	0.98	0.98	0.98	5.47	6.06	0.99	0.99	0.99	93.79	10.90	0.99	0.99	0.99	169.79	4.90	4	3	1	2	2	3	4	1
Tuesday	0.58	0.99	0	13.45	3.96	0.98	0.98	0.98	5.47	4.83	0.98	0.99	0.98	76.45	8.47	0.99	0.99	0.97	175.12	4.47	4	3	2	1	1	3	4	2
Wednesday	0.55	0.99	0	14.37	4.16	0.99	0.99	0.98	5.49	4.76	0.99	0.99	0.99	78.35	9.70	0.99	0.99	0.99	156.33	3.73	3	2	1	2	2	3	4	1
Thursday	0.67	0.99	0	14.37	4.9	0.99	0.99	0.99	5.44	4.1	0.99	0.99	0.99	81.22	9.43	0.99	0.99	0.99	155.00	4.20	3	2	1	1	3	1	4	2
Friday	0.45	0.98	0	14.49	2.9	0.99	0.99	0.99	5.49	4.76	0.99	0.99	0.99	88.52	9.27	0.99	0.99	0.99	168.21	5.17	4	3	2	1	1	2	4	3

Tabla 2.1: Results of DAP, SLP-GA, SLP-GA-Cte and SLP-ACO in energy consumption modelling problems

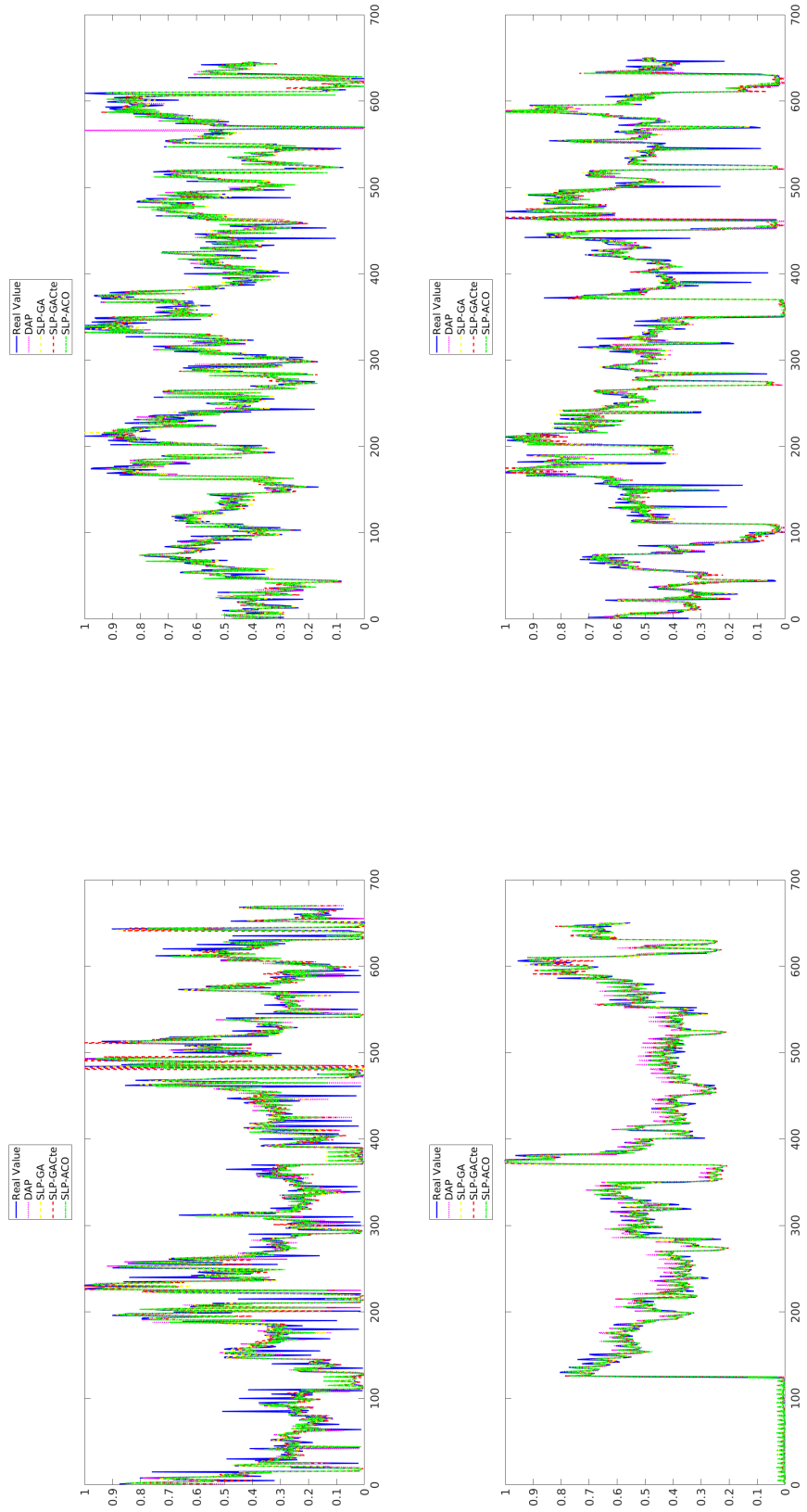
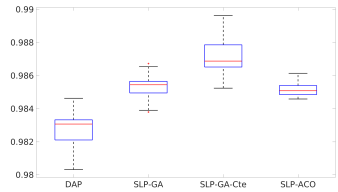
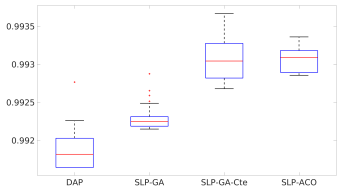


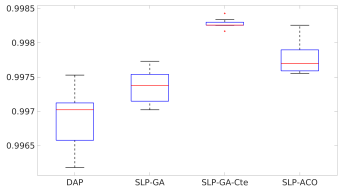
Figura 2.5: Plots of real data (blue), DAP estimated data (magenta), SLP-GA estimated data (yellow), SLP-GA-Cte estimated data (red) and SLP-ACO estimated data (green) for buildings B_1 to B_4



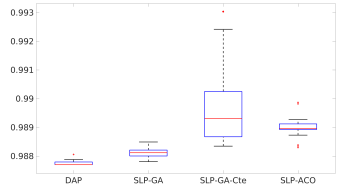
(a) B1-Monday



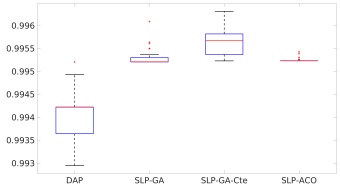
(b) B2-Monday



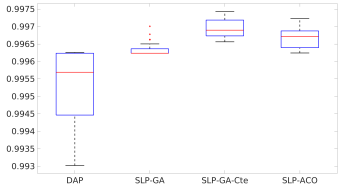
(c) B3-Monday



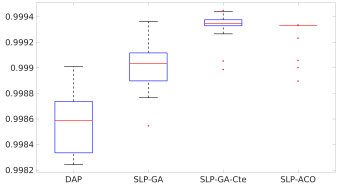
(d) B4-Monday



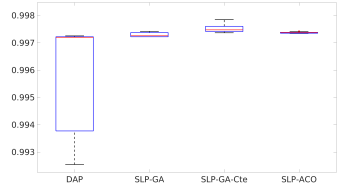
(e) B1-Tuesday



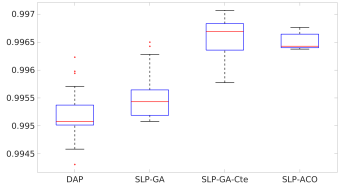
(f) B2-Tuesday



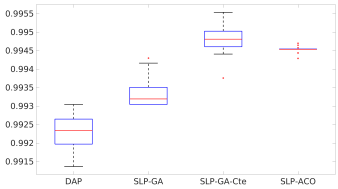
(g) B3-Tuesday



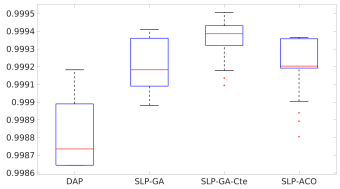
(h) B4-Tuesday



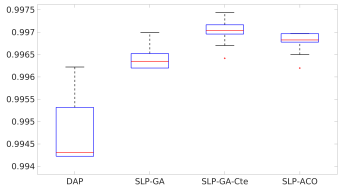
(i) B1-Wednesday



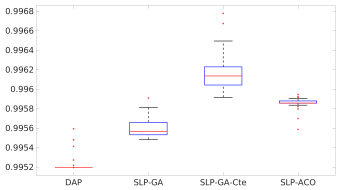
(j) B2-Wednesday



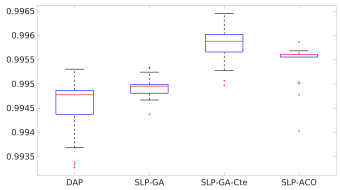
(k) B3-Wednesday



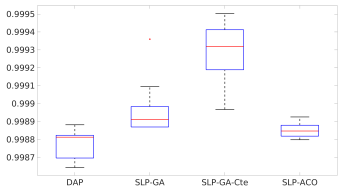
(l) B4-Wednesday



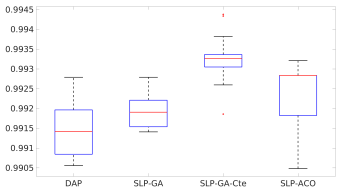
(m) B1-Thursday



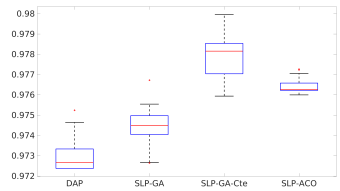
(n) B2-Thursday



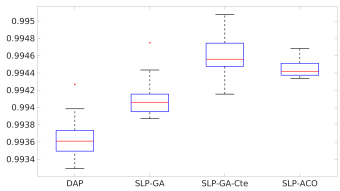
(ñ) B3-Thursday



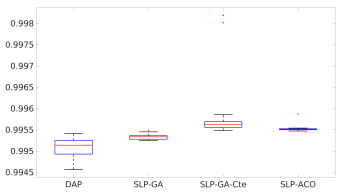
(o) B4-Thursday



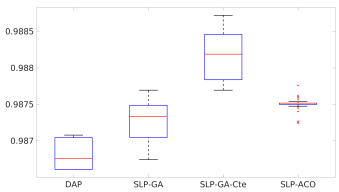
(p) B1-Friday



(q) B2-Friday



(r) B3-Friday



(s) B4-Friday

Figura 2.6: Boxplots of fitness results for each building, working day and algorithm

References

[AB00] D. A. Augusto y H. J. C. Barbosa. «Symbolic regression via genetic programming». En: *6th Brazilian Symposium on Neural Networks, Rio de Janeio, Brazil*. 2000, págs. 173-178.

[Al-+16] M. A. Al-Gunaid y col. «Forecasting energy consumption with the data reliability estimatimation in the management of hybrid energy system using fuzzy decision trees». En: *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*. 2016, págs. 1-8.

[Alo+09] César Luis Alonso y col. «A New Linear Genetic Programming Approach Based on Straight Line Programs: Some Theoretical and Experimental Aspects». En: *International Journal on Artificial Intelligence Tools* 18 (2009), págs. 757-781.

[Alo+12] César L. Alonso y col. «Model Regularization in Coevolutionary Architectures Evolving Straight Line Code». En: *Computational Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 49-65.

[Amb+18] K.P. Amber y col. «Intelligent techniques for forecasting electricity consumption of buildings». En: *Energy* 157 (2018), págs. 886-893.

[Ara+17] Daniel B. Araya y col. «An ensemble learning framework for anomaly detection in building energy consumption». En: *Energy and Buildings* 144 (2017), págs. 191-206.

[BAN02] Maumita Bhattacharya, Ajith Abraham y Baikunth Nath. «A Linear Genetic Programming Approach for Modelling Electricity Demand Prediction in Victoria». En: *Hybrid Information Systems*. Heidelberg: Physica-Verlag HD, 2002, págs. 379-393.

[BB10] Markus F. Brameier y Wolfgang Banzhaf. *Linear Genetic Programming*. 1st. Springer Publishing Company, Incorporated, 2010. ISBN: 978-1-4419-4048-3.

[BC02b] Mariusz Boryczka y Zbigniew J. Czech. «Solving Approximation Problems by Ant Colony Programming». En: *Genetic and Evolutionary Computation Conference (GECCO)*. 2002, págs. 39-46.

[BD02a] Lynne Billard y Edwin Diday. «Symbolic Regression Analysis». En: *Classification, Clustering, and Data Analysis: Recent Advances and Applications*. Berlin, Heidelberg, 2002, págs. 281-288.

[Beh+12] R. Behera y col. «An Application of Genetic Programming for Power System Planning and Operation». En: *International Journal on Control System and Instrumentation, March*. 2012, págs. 15-20.

[BK13b] Florian Benz y Timo Kötzing. «An effective heuristic for the smallest grammar problem». En: *Genetic and Evolutionary Computation Conference, Amsterdam, The Netherlands, July 6-10*. 2013, págs. 487-494.

[Ble+01] S. Bleuler y col. «Multiobjective genetic programming: reducing bloat using SPEA2». En: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*. Vol. 1. 2001, págs. 536-543.

[Bra97] I. Bratko. «Machine Learning: Between Accuracy and Interpretability». En: *Learning, Networks and Statistics*. Vienna, 1997, págs. 163-177.

[CFW01] Tsangyao Chang, Wenshwo Fang y Li-Fang Wen. «Energy consumption, employment, output, and temporal causality: evidence from Taiwan based on cointegration and error-correction modelling techniques». En: *Applied Economics* 33 (2001), págs. 1045-1056.

[CLJ18] Chen Chen, Changtong Luo y Zonglin Jiang. «Block building programming for symbolic regression». En: *Neurocomputing* 275 (2018), págs. 1973-1980.

[CPB17] Alfonso Capozzoli, Marco Savino Piscitelli y Silvio Brandi. «Mining typical load profiles in buildings to support energy management in the smart city context». En: *Energy Procedia* 134 (2017), págs. 865-874. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2017.09.545. URL: http://www.sciencedirect.com/science/article/pii/S187661021734674X.

[CT14] Jui-Sheng Chou y Abdi Suryadinata Telaga. «Real-time detection of anomalous power consumption». En: *Renewable and Sustainable Energy Reviews* 33 (2014), págs. 400-411. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.

2014.01.088. URL: http://www.sciencedirect.com/science/article/pii/S1364032114001142.

[CVH02] Oscar Cordón, Iñaki Fernández de Viana y Francisco Herrera. «Analysis of the Best-Worst Ant System and Its Variants on the QAP». En: *Ant Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, págs. 228-234.

[CW17a] Wenqiang Cui y Hao Wang. «A New Anomaly Detection System for School Electricity Consumption Data». En: *Information* 8.4 (2017). ISSN: 2078-2489. DOI: 10.3390/info8040151. URL: http://www.mdpi.com/2078-2489/8/4/151.

[DBS06] Marco Dorigo, Mauro Birattari y Thomas Stützle. «Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique». En: *IEEE Computational Intelligence Magazine* 1 (ene. de 2006), págs. 28-39.

[DOR92] M. DORIGO. «Optimization, Learning and Natural Algorithms». En: *Ph.D. Thesis, Politecnico di Milano, Italy* (1992).

[DS04] Marco Dorigo y Thomas Stützle. *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004. ISBN: 0262042193.

[FB15] Nelson Fumo y M.A. Rafe Biswas. «Regression analysis for prediction of residential energy consumption». En: *Renewable and Sustainable Energy Reviews* 47 (2015), págs. 332-343.

[Fig+05] V. Figueiredo y col. «An electric energy consumer characterization framework based on data mining techniques». En: *IEEE Transactions on Power Systems* 20.2 (2005), págs. 596-602. DOI: 10.1109/TPWRS.2005.846234.

[FXW14] Cheng Fan, Fu Xiao y Shengwei Wang. «Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques». En: *Applied Energy* 127 (2014), págs. 1-10.

[GNC16] Jun Guan, Natasa Nord y Shuqin Chen. «Energy planning of university campus building complex: Energy usage and coincidental analysis of individual buildings with a case study». En: *Energy and Buildings* 124 (2016), págs. 99-111. ISSN:

0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.04.051. URL: http://www.sciencedirect.com/science/article/pii/S0378778816303164.

[GWJ04] Jennifer Green, Jacqueline L. Whalley y Colin G. Johnson. «Automatic Programming with Ant Colony Optimization». En: *Proceedings of the 2004 UK Workshop on Computational Intelligence*. Loughborough University, sep. de 2004, págs. 70-77.

[Huo+08] L. Huo y col. «Short-Term Load Forecasting Based on Improved Gene Expression Programming». En: *2008 4th IEEE International Conference on Circuits and Systems for Communications*. Mayo de 2008, págs. 745-749.

[KK18] Kostas Karatzas y Nikos Katsifarakis. «Modelling of household electricity consumption with the aid of computational intelligence methods». En: *Advances in Building Energy Research* 12.1 (2018), págs. 84-96.

[Kom+13] Michael Kommenda y col. «Nonlinear Least Squares Optimization of Constants in Symbolic Regression». En: *Computer Aided Systems Theory - EUROCAST*. Berlin, Heidelberg, 2013, págs. 420-427.

[LRW16a] Qiang Lu, Jun Ren y Zhiguang Wang. «Using Genetic Programming with Prior Formula Knowledge to Solve Symbolic Regression Problem». En: *Computational Intelligence and Neuroscience* 2016 (ene. de 2016), págs. 1-17.

[Mar63] Donald W. Marquardt. «An algorithm for least-squares estimation of nonlinear parameters». En: *Journal of the Society for industrial and Applied Mathematics* 11 (1963), págs. 431-441.

[MB12] B. Chandra Mohan y R. Baskaran. «A survey: Ant Colony Optimization based recent research and implementation on several engineering domain». En: *Expert Systems with Applications* 39 (2012), págs. 4618-4627.

[McK+10] Robert I. McKay y col. «Grammar-based Genetic Programming: a survey». En: *Genetic Programming and Evolvable Machines* 11.3 (2010), págs. 365-396. ISSN: 1573-7632. DOI: 10.1007/s10710-010-9109-y.

[MK89] Kazuo Murata y Keiichi Kohno. «Nonlinear least-squares regression analysis by a simplex method using differential equations containing michaelis-menten type rate constants». En: *Biopharmaceutics and Drug Disposition* 10 (1989), págs. 25-34.

[MWB95] B. McKay, M. J. Willis y G. W. Barton. «Using a tree structured genetic algorithm to perform symbolic regression». En: *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. 1995, págs. 487-492.

[MZ90] Marcel. Maeder y Andreas D. Zuberbuehler. «Nonlinear least-squares fitting of multivariate absorption data». En: *Analytical Chemistry* 62 (1990), págs. 2220-2224.

[ORV10] J. L. Olmo, J. R. Romero y S. Ventura. «A grammar based Ant Programming algorithm for mining classification rules». En: *IEEE Congress on Evolutionary Computation*. 2010, págs. 1-8.

[PJP14] J. Pan, R. Jain y S. Paul. «A Survey of Energy Efficiency in Buildings and Microgrids using Networking Technologies». En: *IEEE Communications Surveys Tutorials* 16 (2014), págs. 1709-1731.

[PLM08a] Riccardo Poli, William B. Langdon y Nicholas Freitag McPhee. *A field guide to genetic programming*. 2008.

[RMB19] Pablo Rodriguez-Mier, Manuel Mucientes y Alberto Bugarín. «Feature Selection and Evolutionary Rule Learning for Big Data in Smart Building Energy Management». En: *Cognitive Computation* 11 (2019), págs. 418-433.

[Rob+17] Caleb Robinson y col. «Machine learning approaches for estimating commercial building energy consumption». En: *Applied Energy* 208 (2017), págs. 889-904.

[RRu+17] R.Rueda y col. «Experimental Evaluation of Straight Line Programs for Hydrological Modelling with Exogenous Variables». En: *Hybrid Artificial Intelligent Systems: 12th International Conference, HAIS 2017, La Rioja, Spain, June 21-23, 2017, Proceedings*. 2017, págs. 447-458.

[Rue+17b] R. Rueda y col. «Preliminary Evaluation of Symbolic Regression Methods for Energy Consumption Modelling». En: *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal.* 2017, págs. 39-49.

[Rue+18a] Ramón Rueda Delgado y col. «A Comparison Between NARX Neural Networks and Symbolic Regression: An Application for Energy Consumption Forecasting». En: *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*. Springer International Publishing, 2018, págs. 16-27.

[Sal05] Timothy I. Salsbury. «A survey of control technologies in the building automation industry». En: *IFAC Proceedings Volumes* 38.1 (2005). 16th IFAC World Congress, págs. 90-100. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20050703-6-CZ-1902.01397. URL: http://www.sciencedirect.com/science/article/pii/S1474667016374092.

[SDV12] Sara Silva, Stephen Dignum y Leonardo Vanneschi. «Operator equalisation for bloat free genetic programming and a survey of bloat control methods». En: *Genetic Programming and Evolvable Machines* 13 (2012), págs. 197-238.

[SL07] Michael Schmidt y Hod Lipson. «Comparison of Tree and Graph Encodings As Function of Problem Complexity». En: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. GECCO '07. 2007, págs. 1674-1679.

[SON11] Shinichi Shirakawa, Shintaro Ogino y Tomoharu Nagao. «Automatic Construction of Programs Using Dynamic Ant Programming». En: *Ant Colony Optimization*. Rijeka: IntechOpen, 2011.

[SW08] Amirali Salehi-Abari y Tony White. «Enhanced Generalized Ant Programming (EGAP)». En: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA, 2008, págs. 111-118.

[SW09] Amirali Salehi-Abari y Tony White. «The Uphill Battle of Ant Programming Vs. Genetic Programming.» En: *IJCCI 2009 - International Joint Conference on Computational Intelligence, Proceedings*. Ene. de 2009, págs. 171-176.

[WD10] P. A. Whigham y G. Dick. «Implicitly Controlling Bloat in Genetic Programming». En: *IEEE Transactions on Evolutionary Computation* 14 (2010), págs. 173-190.

[WS17] Zeyu Wang y Ravi S. Srinivasan. «A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models». En: *Renewable and Sustainable Energy Reviews* 75 (2017), págs. 796-808. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.10.079. URL: http://www.sciencedirect.com/science/article/pii/S1364032116307420.

[YZW19] Yang Yu, Zhihong Zou y Shanshan Wang. «Statistical regression modeling for energy consumption in wastewater treatment». En: *Journal of Environmental Sciences* 75 (2019), págs. 201-208.

[Zor+16] Angel L. Zorita y col. «A statistical modeling approach to detect anomalies in energetic efficiency of buildings». En: *Energy and Buildings* 110 (2016), págs. 377-386.

2.3. Generalised Regression Hypothesis Induction for Energy Consumption Forecasting.

- **Referencia**: R.Rueda, M.P.Cuéllar, M.Molina-Solana, Y.Guo, M.C.Pegalajar (2019). Generalised Regression Hypothesis Induction for Energy Consumption Forecasting. Energies, 12, 1-22
- **Estado**: Publicado
- **Factor de impacto** (JCR 2019): 2.702
- **Categoría**: Posición 63/112 en el área "Energy & Fuels". Q3
- **DOI**: 10.3390/en12061069.
- **Revista/Editorial**: Energies / MDPI

Generalised Regression Hypothesis Induction for Energy Consumption Forecasting

R.Rueda [1], M.P.Cuéllar [1], M.Molina-Solana [2], Y.Guo [2], M.C.Pegalajar [1]

[1] Department of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, Spain

[2] Data Science Institute, Imperial College, London SW7 2AZ, UK

Abstract

This work addresses the problem of energy consumption time series forecasting. In our approach, a set of time series containing energy consumption data is used to train a single, parameterised prediction model that can be used to predict future values for all the input time series. As a result, the proposed method is able to learn the common behaviour of all time series in the set (i.e., a fingerprint) and use this knowledge to perform the prediction task, and to explain this common behaviour as an algebraic formula. To that end, we use symbolic regression methods trained with both single- and multi-objective algorithms. Experimental results validate this approach to learn and model shared properties of different time series, which can then be used to obtain a generalised regression model encapsulating the global behaviour of different energy consumption time series.

Keywords: Symbolic Regression, Energy Consumption, Forecasting, Pattern Recognition

1 Introduction

Energy efficiency in the building sector has become an important research area for two main reasons: firstly, because the residential sector represents around 25 % of global energy consumption; and secondly, because the building sector is also considered as the main contributor to the

energy shortage and climate change effects regarding the worldwide increased population and the large environmental impacts [San16; Ber17]. Building infrastructures include sensor technologies [Höl+14] that provide a huge amount of energy consumption data that allows researchers to address the problem of energy usage and its environmental impact from different perspectives [Mol+17], such as anomaly detection [CT14; CW17b], energy consumption modelling [Lü+15; GK17], energy demand planning [GNC16] or consumer profile mining [GA17; CPB17].

Each of these problems has been addressed with different techniques, according to the nature of the problem and the desired output. For instance, in anomaly detection problems, [CT14] used a neural network to predict the energy consumption of future days and attempted to detect the energy consumption anomalies identifying the differences between the real and predicted energy data. On the other hand, [CW17b] proposed a hybrid model that combines polynomial regressions and Gaussian distributions to build data detection and visualisation systems that help to identify anomalies in electricity consumption data. Regarding consumer profile problems, Gomez et al. [GA17] used a data-fitting approach and a multi-class classifier to estimate the electricity needed in a building, and [CPB17] used pattern recognition and classification algorithms to provide knowledge about the energy usage in a building. Regarding energy consumption modelling, Zhao et al. [Zha+14] used occupant information in addition to the heating, ventilation and air conditioning (HVAC) energy consumption data to determine the pollution impact of buildings. As another example, Balaji et al. [Bal+13] built a control system that uses the WiFi network traffic in a building to estimate the occupancy and control the HVAC. Due to the heterogeneous nature of the different sources of data, every approach usually needs a preprocessing stage to provide useful data to the models, as argued in [Lü+15].

Our current work focuses on energy consumption forecasting in public buildings. Traditionally, energy consumption forecasting in residential and public buildings has been implemented by means of analysing a single energy consumption time series [Amb+17; SZ17; NF08; BRF16; Jai+14]. In these cases, this single time series is given as input to a prediction model, with the potential addition of some external data (temperature, building occupancy, etc.). These models have proven able to provide suitable results in their respective case studies, obtaining accurate prediction models. On the other hand, there are many works that combine historical energy

data with statistical and machine learning techniques to provide more accurate models able to improve the energy consumption in buildings.

A good and recent review on this topic [Deb+17] summarises the state of the art in the energy consumption forecasting paradigm, and describes the most used techniques to that end. To only cite a few, Yasmin et al. developed ARIMA models (Autorregresive Intregrated Moving Average) [YS14] to predict electricity consumption in Pakistan and [Rui+16] proposed nonlinear autoregressive artificial neural networks with exogenous inputs (NARX) to predict the energy consumption in public buildings. There is a plethora of proposals in the literature for energy consumption forecasting in the last few years, and a complete survey paper would be necessary to analyse all these research approaches.

Unlike traditional approaches in energy consumption forecasting for buildings, which use a single time series that contains historical data of energy consumption and exogenous information, we address our research from a different perspective and we assume that there are multiple buildings whose energy consumption must be predicted. We hypothesise that, if the energy consumption of different buildings is medium or highly correlated, then these buildings share the same energy consumption ground behaviour. Then, our goal is to obtain a general forecasting model able to learn structural relationships of all time series in a set, and to parameterise this model for its particular adaptation for each specific building, obtaining a more accurate prediction model that explains the overall energy consumption behaviour of the whole compound.

To address this problem, we use symbolic regression and genetic algorithms to find interpretable regression models that describe the relationships in the energy consumption data that are common to all buildings under study in a compound. More specifically, we assume that each related building provides a dataset with their energy consumption, and our motivation is to find a general regression hypothesis f able to explain all datasets of each building. These regression models will then be parameterised for its adaptation to a specific building forecasting task. As we have not found any work in the literature that solves a similar problem, we attempt to use two alternatives, namely classical genetic algorithms and multi-objective optimisation approaches, to identify the strengths and weaknesses of each type of techniques. In addition, in the experiments,

we firstly validated the approach with synthetic data generated in the laboratory, and then in a real scenario. In both cases, we departed from several energy consumption time series that are medium or highly correlated, and we attempted to find a single regression model able to learn the shared ground behaviour of all these data series. The ultimate goal was to predict their future values accurately with the same model, parameterised for each data series.

To achieve these objectives, this manuscript is organised as follows. Section 2 introduces the fundamentals of symbolic regression and the different historical alternatives used to solve multi-objective problems. After that, Section 3 describes the formulation of the problem and the proposed methods. Section 4 shows the experimental results in two scenarios: synthetic data generated in the laboratory and real energy consumption data. Finally, conclusions and future works are described in Section 5.

2 Related Work

The literature gathers several techniques able to solve energy forecasting problems, such as neural networks [Afr+17; JSZ15], support vector machines, decision trees [AMR17; WS17] or regression analysis techniques [YBS17; BAB14]. Since we were interested in finding not only accurate but also interpretable solutions for the final user, we focus this research in regression analysis techniques, and study their limitations and the solutions provided by different authors in the literature in Section 2.1. After that, we establish the basic concepts of multi-objective optimisation techniques required for our research in Section 2.2.

2.1 Symbolic Regression

Regression analysis is a classical tool widely used by researchers in prediction and data modelling problems. Given an algebraic expression f as model hypothesis, a set of input data (independent variables) $\bar{x} = (x_1, x_2, ..., x_n)$ and output data (dependent variables) $\bar{y} = (y_1, y_2, ..., y_m)$, the regression analysis attempts to find the optimal model parameters $\bar{w} = (w_1, w_2, ..., w_k)$ such as

$\bar{y} = f(\bar{x}, \bar{w})$, where $x_i, y_j, w_l \in \mathbb{R}, \forall i, j, l : 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq l \leq k$. The parameters $\bar{w}$ are usually found using numerical procedures, such as least squares optimisation, that minimise an error measurement, for instance $e(f(\bar{x}, \bar{w}), \bar{y}) = ||f(\bar{x}, \bar{w}) - \bar{y}||$. However, the main limitation of traditional regression analysis appears when not only the parameters $\bar{w}$ are unknown, but also the model hypothesis f, or when f is very difficult to formulate manually. To solve this limitation, symbolic regression [MWB95] combines a set of predefined atomic operators (such as +, -, *, /, and sin), independent variables $\bar{x}$ and parameters $\bar{w}$ to build an algebraic expression $\tilde{f}$ as an approximation for the optimal model f. To do so, symbolic regression uses optimisation algorithms to explore the search space and finds the best approximation $\tilde{f}$ that minimises an error measure, such as $||\bar{y} - \tilde{f}(\bar{x}, \bar{w})||$.

Although there are many techniques designed to solve optimisation problems, we chose genetic programming due to its demonstrated potential in several areas [Wil+97], including symbolic regression. Genetic programming [Lan98] is a supervised machine learning method based on biological evolution and is used in symbolic regression problems since it evolves a population of candidate algebraic expressions $\tilde{f}(\bar{x}, \bar{w})$ and applies a set of genetic operators [Koz94] to obtain the best candidate $\tilde{f}$. Although tree structures are highly used to encode the algebraic expressions of the population in genetic programming algorithms, our previous research [Rue+18b] has demonstrated that alternative representations such as Straight Line Programs (SLP) are able to improve the solutions of symbolic regression problems in terms of not only accuracy and computational time but also obtaining more interpretable algebraic expressions.

Symbolic regression has been proposed previously as a tool to model energy consumption. The state of the art in this topic includes the work by [Yan+16], who used symbolic regression and evolutionary algorithms to identify the main factors in the energy consumption in China. Whereas [BAN02] used linear genetic programming to perform consumer electricity demand forecasting, [Beh+12] used a genetic algorithm in order to develop an effective planning system able to estimate demand and energy consumption. In our works, we have also studied the use of symbolic regression for energy consumption modelling with good results [Rue+18b].

Benefits of symbolic regression for energy consumption prediction are the simplicity of the

resulting models, in contrast to more complex and difficult to analyse models such as neural networks or support vector machines, and also their interpretability by a non-expert in machine learning. On the other hand, limitations arise in the side of accuracy when dependencies between input and output data are difficult to find. In these cases, universal approximators such as neural networks have shown good performance [Rui+16]. Overall, we have selected symbolic regression as representation of forecasting model hypotheses due to the balance between their easy interpretability and good accuracy.

2.2 Multi-Objective Optimisation Paradigm

As previously described, our goal is to obtain a single general and parameterised forecasting model able to predict the energy consumption of different buildings. Thus, given a set of N energy consumption time series as input (one for each building), the task at hand is to find a unique parameterised regression model $\tilde{f}(\bar{x^i}, \bar{w^i})$ that consistently models and predict future values of each ith time series separately. If the values $\bar{y^i}$ are the desired outputs for forecasting the ith time series, then we can measure N prediction errors as $e(\tilde{f}, i) = ||\bar{y^i} - \tilde{f}(\bar{x^i}, \bar{w^i})||$. Finding the desired model $\tilde{f}$ therefore requires the minimisation of the N error measurements separately. In this work we consider two approaches to address this problem: (1) a single-objective approach, where all error measurements are aggregated into an unique measurement as $e(\tilde{f}) = \sum_{i=1}^{N} e(\tilde{f}, i)$; and (2) using multi-objective optimisation, where each error measurement $e(\tilde{f}, i)$ would be considered as a separate target function to be minimised.

Multi-objective optimisation problems attempt to find models that minimise/maximise a set of objective measures simultaneously, and represent the solution of each objective in a vector function e. Formally, a multi-objective problem is defined as the minimisation of multiple criteria $e(\bar{x}) = (e_1(\bar{x}), e_2(\bar{x}), ..., e_m(\bar{x}))$, where $\bar{x} = (x_1, x_2, ..., x_n)$ is the vector of decision variables, and $e(x) = (e_1, e_2, ..., e_m)$ are the objective functions that must be minimised/maximised. An important aspect to take into account solving a multi-objective problem is related with the problem statement. As [Has10] argued, a multi-objective problem can be addressed from

three different perspectives: converting the multi-objective problem in a single-objective one [MA10; JGB92; MA04], using population-based algorithms [MA04; Sch85; KCV02] and studying Pareto-optimal based techniques [ZT99; ZLT01; KC99; SD94; Deb+02].

Multi-objective optimisation techniques have been used to solve energy consumption problems (e.g., [Yan+17; Wu+17; Asc+17; HHS11]). Therefore, in this work, we compare single-objective and NSGA-II multi-objective approaches to train symbolic regression forecasting models. The next section describes the problem statement of each approach, the representation used to solve symbolic regression and the main components of the algorithmic procedure.

3 Methods

Our method is based on the assumption that there is a set of several datasets (time series) composed of input/output patterns with the same structure, for which we want to model the output data with respect to the input data and obtain an algebraic expression that models the input/output relationship. We also hypothesise that there is a shared ground common behavior that is present in all these datasets, and that can explain output data regarding input data (at least partially). The model for this shared behavior could be unknown in advance. Due to this assumption, we consequently expect a medium/high correlation between the output data of all datasets.

As a toy example to understand these hypotheses, we sampled data from two linear functions $y_1 = f_1(x_1) = 3 * x_1 + 5 + \epsilon_1$, and $y_2 = f_2(x_2) = 6 * x_2 + 1 + \epsilon_2$ (where ϵ_1, ϵ_2 stand for an error in each dataset, respectively) and obtained two datasets (X_1, Y_1) and (X_2, Y_2). Both datasets come from different data distributions, but share a common ground behavior that could explain y_1 from x_1 and y_2 from x_2, and could be written as $y = f(x) = a * x + b$. Thus, the regression hypothesis f can be used to explain both datasets, when it is parameterised by the coefficients a and b.

We could use traditional symbolic regression methods to solve this toy example, by means of solving each dataset separately, and probably obtaining algebraic expressions that match f_1

and f_2, respectively. In this article, we provide an automatic method to find general regression hypotheses that could help to explain multiple datasets, providing a single parameterised algebraic expression that model the common ground behavior of all datasets. A necessary constraint of our proposal is that all datasets must have the same data structure, i.e., they are composed by a set of independent (input) variables $X = (x_1, x_2, ..., x_k)$ where $x_m \in \mathbb{R}, \forall m : 1 \leq m \leq k$, and the number of variables k is the same for all datasets, and a single dependent (output) variable $Y = (y_1)$ where $y_1 \in \mathbb{R}$.

The problem formulation assumes a set of n datasets $(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$ whose data have this structure, i.e., $X_i = (x_1^i, x_2^i, ..., x_k^i)$ and $Y_i = (y_1^i)$ $\forall i : 1 \leq i \leq n$ are the input and output variables, respectively, of the ith dataset. We denote $N(i) = (X_i, Y_i)$ as the number of input/output patterns of the ith dataset. We remark that two datasets can have different sizes (number of input/output patterns), so that $N(i)$ can be different to $N(j)$ for two different datasets, without loss of generality. Thus, the main goal of this work consists of finding a general parameterised algebraic expression $F(X, W)$ able to explain the relationships between dependent (output) and independent (input) data from all datasets simultaneously, where $W = (w_1, w_2, ..., w_l) : w_q \in \mathbb{R} \forall q : 1 \leq q \leq l$ is a set of parameters or coefficients estimated for each specific dataset. Finding the values for the parameters of each dataset, W_i, is also a component of the proposed method and implemented as a local search procedure within a genetic evolutionary approach. We deepen into the algorithms description in the following sections.

3.1 Straight Line Programs for Time Series Prediction

A time series is a sequence of data (that we call x) evenly sampled in time (t) that is used to predict futures values $x(t + k)$. The goal of time series prediction is to find a model f that combines h previous values of the time series data and parameters w, to forecast the next values of the time series as $x(t + 1) = f(x(t), x(t - 1), ..., x(t - h), w)$. The value h is usually called *time horizon*.

In this paper, we use symbolic regression to predict future values of energy consumption time series. By using symbolic regression, we assume that the prediction model f can be written as an algebraic expression. Although there are different alternatives to encode algebraic expressions, including trees and linear genetic programming, Straight Line Programs have proved their potential over classical representations [APM08]. Straight Line Programs (SLP) are represented as a sequence of grammar production rules, where each production rule is composed by a set of known mathematical operators $O_{U_i} \in \{o_1, o_2, ..., o_m\}$ (as for instance $\{+, -, *, /, \sin, \cos\}$), a set of terminal symbols $T = \{t_1, t_2, ..., t_k\}$ (as for instance, parameters $\bar{w} = \{w_1, w_2, ..., w_k\}$ or independent variables $\bar{x} = \{x_1, x_2, ..., x_n\}$) and references to other rows $R_{U_{i,1}}, R_{U_{i,2}} \in \{T \cup \{U_1, U_2, ..., U_{i-1}\}\}$. We may verify that the production rules that appear in the consequent must be references to previous rules, in order to avoid recursion.

$$\begin{cases} U_1 \rightarrow 2 * x \\ U_2 \rightarrow 8 + U_1 \\ U_3 \rightarrow sin(U_2) \\ U_4 \rightarrow U_3 - U_2 \\ U_5 \rightarrow x * U_4 \\ U_6 \rightarrow U_2 / U_5 \end{cases} \tag{2.17}$$

We can build an algebraic expression from a SLP by evaluating the Nth rule, which is the starting symbol of the grammar U_N. As an example, Equation (2.17) gathers a SLP that encodes the algebraic expression $f(x, \bar{w}) = \dfrac{w_1 + (w_2 * x)}{x * \sin(w_1 + (w_2 * x)) - w_1 + (w_2 * x)}$, where $\bar{w} = (w_1, w_2) = (8, 2)$.

The following subsections gather the problem formulation for each approach and also a description of the main genetic operators to train SLP representations using genetic algorithms.

3.2 Single-Objective Problem Formulation

The problem addressed in this research is applied over multiple time series simultaneously,

and it assumes that symbolic regression should find a general algebraic expression $F(X, W)$ that models the dependent variables of several datasets simultaneously, where X represents the independent variables of the datasets and W is the set of parameters values estimated by symbolic regression. The single-objective formulation of the problem assumes that symbolic regression should minimise an error function $e(F)$ in order to find the general algebraic expression F. We calculate this error $e(F)$ as the sum of the errors of F of all datasets to approximate the desired output Y (Equation (2.18)).

$$e(F) = \sum_{i=1}^{n} (\frac{1}{N(i)} * \sum_{p=1}^{N(i)} (F(X_i(p), W_i) - Y_i(p))^2) \tag{2.18}$$

where n stands for the number of datasets, $N(i)$ is the number of data samples of the ith dataset, $X_i(p)$ is the pth input sample of the ith dataset, $Y_i(p)$ is the pth output sample of the ith dataset, and W_i are the values of the algebraic function parameters of the ith dataset. The optimisation problem is formulated as finding the algebraic expression $\tilde{F} = min_F(e(F))$. Once the problem is formulated as explained, we use a genetic algorithm to find a SLP that provides the best regression hypothesis $\tilde{F}$.

Using the previous formulation, the accuracy error of the prediction model F over a time series X_i is aggregated into a single measurement $e(F)$. In contrast, the multi-objective formulation shown in the next section treats the error minimisation of each time series separately.

3.3 Multi-Objective Problem Formulation

The single objective problem formulation described in the previous section is likely to have limitations if datasets are not normalised, since $e(F)$ is defined as the sum of squared errors in all datasets. If the data scale varies significantly from one dataset to another, then the errors could also vary significantly, and then the datasets with higher absolute errors could dominate the search in the solution space. In cases in which the data cannot be normalised, or if we do not know lower and upper bounds to perform a normalisation, then the single objective strategy could lead us to obtain undesired local optima solutions. To solve this limitation, we also provide

a problem formulation from a multi-objective optimisation perspective.

In it, we define a minimisation objective per each dataset, so we assume a set of n objectives $(O_1, O_2, ..., O_n)$, where each objective O_i attempts to minimise the error function of the corresponding ith dataset (X_i, Y_i). Therefore, in the searching process of the general algebraic expression F, the multi-objective approach attempts to minimise each objective individually, finding the function F that minimises the error of the ith objective without worsening the quality of the jth objective. In this way, this formulation may avoid local optima when the data are not normalised since bad solutions in the estimation of some datasets do not influence the exploration of the search space for the remaining objectives. Thus, the goal is to find a general algebraic expression F that minimises an error function $e(F, i)$ for each of the objectives. The error measurement that must be minimised for each objective is shown in Equation (2.19).

$$\begin{aligned}
O_1 &= e(F,1) = \frac{1}{N(1)} * \sum_{p=1}^{N(1)} (F(X_1(p), W_1) - Y_1(p))^2 \\
&\vdots \\
O_i &= e(F,i) = \frac{1}{N(i)} * \sum_{p=1}^{N(i)} (F(X_i(p), W_i) - Y_i(p))^2 \\
&\vdots \\
O_n &= e(F,n) = \frac{1}{N(n)} * \sum_{p=1}^{N(n)} (F(X_n(p), W_n) - Y_n(p))^2
\end{aligned} \tag{2.19}$$

where n stands for the number of objectives (i.e., the number of datasets), $N(i)$ is the number of data samples of the ith dataset, $X_i(p)$ is the pth input sample of the ith dataset, $Y_i(p)$ is the pth output sample of the ith dataset, and W_i is the set of parameter values for the target algebraic expression F for the ith dataset. The multi-objective formulation of the problem is to find the optimal algebraic expression $\tilde{F}$, or the set of Pareto-optimal algebraic expressions $\tilde{F}$, such as $\tilde{F} = min_F(e(F,i)) \forall 1 \leq i \leq n$. Once the problem is formulated as explained, we train SLPs to minimise $(O_1, O_2, ..., O_n)$ using the multi-objective algorithm NSGA-II [Deb+02].

3.4 Algorithm Description

We use a classic genetic algorithm [Rue+18b] to optimise the single-objective approach, and the NSGA-II [Deb+02] method for the multi-objective optimisation approach. Since no changes were made to these template algorithms, in this section, we only describe the genetic operators to be used with SLP representation. We also remark that both crossover and mutation operators were proposed by Alonso et al. [APM08] and we summarise the procedure of each operator with the aim of improving the understanding of this work.

- Crossover operator. Two parents P_1 and P_2 are used in both single- and multi-objective approaches in order to generate two new children C_1 and C_2. The operator starts out selecting a random rule $U_i \in \{U_1, U_2, ..., U_{N-1}\}$ from P_1. After that, an ordered set of rules R is calculated as the set of rules $U = \{U_1, U_2, ..., U_{i-1}\}$ that can be reached from the selected rule U_i. Then, a random rule $U_k \in \{U_1, U_2, ..., U_{N-|R|+1}\}$ from P_2 is selected, where $|R|$ is the number of rules included in R. The offspring C_1 is created as a copy of the parent P_2 and the rules in R are copied into C_1 and renamed from $U_{k-|R|+1}$ to U_k. Finally, the offspring C_2 is generated with the same procedure, but exchanging the roles of both parents P_1 and P_2. An example of this operator is shown in Figure 2.1, where rule $r_1 = 5$ was selected randomly from parent P_1. After that, the ruleset U is created as the set of rules that can be reached from U_5. In this case, $U = \{U_5, U_{2,1}\}$, and is renamed as $R = \{r_1, r_2, r_3\}$. Then, a random position $r_2 = 4$ is selected in P_2, and the offspring C_1 is created as a copy of P_2 with the replacement of rules in R, starting from r_2.

- Mutation operator. Given a SLP table of an individual of the population, a random element of the consequent of a random rule is exchanged for another random symbol. If the selected element is an operator, it is exchanged by another valid operator and if the selected element is an operand, it is exchanged by a terminal symbol or a reference to other rule, as shown in Figure 2.2. On the other hand, if the mutation operator exchanges a binary operator by an unary operator, the second operand of the rule is left to the value $\emptyset$. Nevertheless, if the operator mutes an unary operator to a binary operator, then the

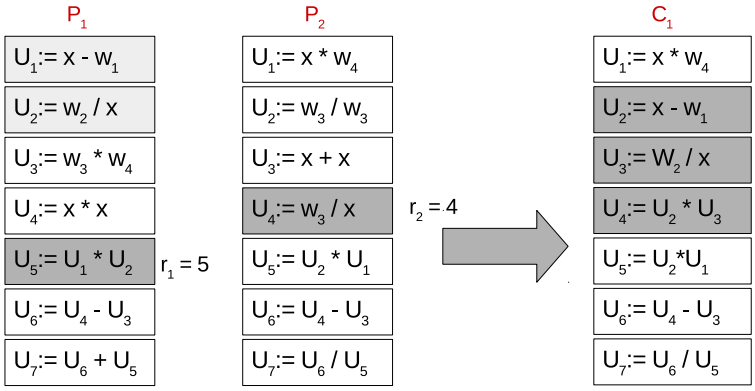


Figura 2.1: Example of a SLP crossover.

second operand is randomly selected from the set of valid operands of the production rule (independent variables, parameters or references to other rules of the SLP table).

Once both single- and multi-objective algorithms have found an algebraic expression, a local search procedure is applied in order to estimate the parameters W for each objective. In this way, both algorithms are able to provide a general algebraic expression which share common behaviour from all datasets (objectives) and parameters are conveniently fitted to satisfy each objective.

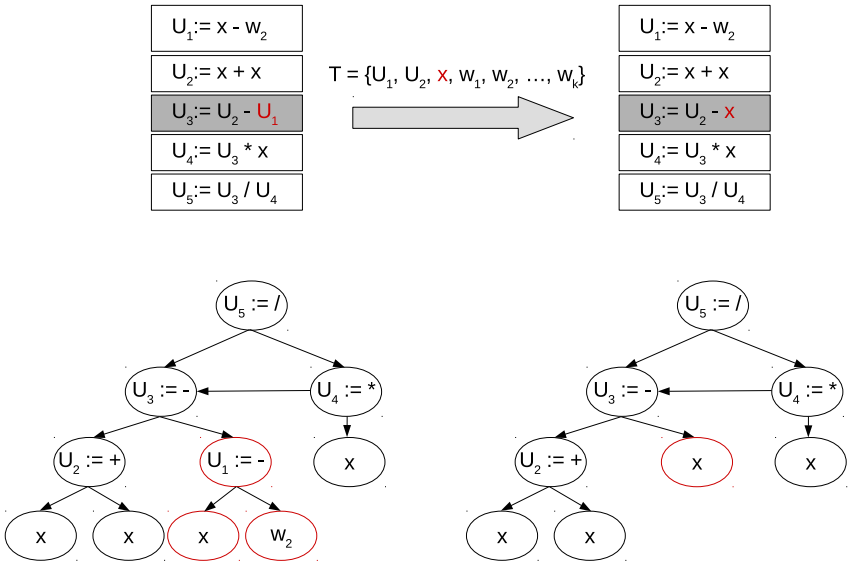


Figura 2.2: Example of a SLP mutation.

4 Experimentation

We tested and experimentally validated both single-objective and multi-objective approaches in two experiment setups:

- A first study attempted to empirically validate whether the described problem can be solved with the proposed formulations. Thus, we built different scenarios with synthetic data, with the aim of validating the performance of each approach under a controlled experimental environment that eases the analysis of performance of the approaches. To that end, Section 4.1 describes a set of benchmark algebraic expressions to be used in the experiments, and the results obtained with each approach. This section ends with a discussion of the results obtained.

- In the second experiment (Section 4.2), we tackled a real problem about energy consumption prediction. In it, we were provided with the energy consumption time series of a set of buildings for which we assumed there is a medium or high correlation and the goal was to find a single parameterised algebraic expression $F(X_i, W_i)$ that can explain the global/common behaviour of all related energy consumption data series.

4.1 Experimentation with Synthetic Data

We used a set of benchmark algebraic expressions to create a set of synthetic datasets with the same ground behaviour. The main goal of this experimentation was to test if the proposal of this paper can be used to obtain a general regression hypothesis that explains all datasets coming from the same algebraic expression, with different parameters W for each dataset, in this controlled environment.

This section is divided into three parts: firstly, Section 4.1.1 explains the set of benchmark algebraic expressions, and how all artificial datasets were generated. Then, Sections 4.1.2 and

4.1.3 show the experimental configuration used in each approach and the results obtained, respectively.

4.1.1 Data Acquisition

We used six benchmark algebraic expressions (see Equations (2.20)–(2.25)) that are widely used in the literature to test the quality of symbolic regression algorithms [Nic+15]. More specifically, we selected benchmark algebraic expressions with different complexity (which include polynomial, trigonometrical and exponential expressions) and also with a varying number of parameters (from one parameter (Equations (2.23)–(2.25)) to six parameters (Equation (2.22))). Besides, we generated different datasets for each benchmark algebraic expression to empirically validate if both single-objective and multi-objective approaches have the same behavior, or in contrast, to find which technique carries out a better exploration of the search space when the number of datasets increases.

$$f_1(x_1, x_2) = \frac{e^{-(x_1 - w_1)^2}}{w_2 + (x_2 - w_3)^2} \tag{2.20}$$

$$f_2(x_1, x_2) = \frac{w_1}{w_1 + x_1^{w_2}} + \frac{w_1}{w_1 + x_2^{w_2}} \tag{2.21}$$

$$f_3(x_1, x_2, x_3, x_4, x_5) = w_1 + w_2 * \frac{w_3 * x_2 + w_4 * x_3^2}{w_5 * x_4^3 + w_6 * x_5^4} \tag{2.22}$$

$$f_4(x_1, x_2) = w_1 * \sin(x_1) * \cos(x_2) \tag{2.23}$$

$$f_5(x_1, x_2) = (x_1 - w_1) * (x_2 - w_1) + w_2 * \sin((x_1 - w_3) * (x_2 - w_3)) \tag{2.24}$$

$$f_6(x_1, x_2) = x_1^4 - x_1^3 + \frac{x_2}{w_1} - x_2 \tag{2.25}$$

We generated five datasets for each benchmark algebraic expression shown in Equations (2.20)–(2.25). All five datasets regarding a single algebraic expression share the same values for the input data, i.e., for each independent variable x_i of each benchmark algebraic expression, we generated 200 samples uniformly distributed in the range [0,0, 1,0]. After that, we selected different values for the parameters w_i of each dataset regarding the same algebraic expression. Each value w_i was randomly generated in the interval [0,0, 5,0]. Thus, for each benchmark algebraic expression, we were provided with five datasets with the same inputs but different outputs.

As an example, let us consider the dataset generation for the benchmark algebraic expression of Equation (2.21): firstly, we generated 200 random samples for x_1 and x_2, which were used as inputs for all datasets. After that, we generated five different values for each parameter w_i: $(w_1^1, w_2^1) = (3{,}23, 4{,}12)$ for the first dataset; $(w_1^2, w_2^2) = (1{,}09, 2{,}35)$ for the second dataset; and $(w_1^3, w_2^3) = (2{,}17, 0{,}59)$, $(w_1^4, w_2^4) = (3{,}0, 0{,}64)$, and $(w_1^5, w_2^5) = (3{,}83, 1{,}1)$ for the remaining datasets. This allowed us to obtain different datasets with the same inputs but different outputs, and with a high correlation between all datasets, which is a prerequisite of our proposal.

Both single- and multi-objective approaches were tested in two cases: considering a low number of datasets, and considering a larger number of datasets. We performed an experimentation using three of the five datasets generated for each benchmark algebraic expression, and another experimentation using all datasets, with the goal of discovering if the number of datasets has an influence in the performance of the multi-objective approach.

4.1.2 Experimental Settings

For the experimentation, we used 12 mathematical operators (*+, -, *, /, sqrt, pow, exp, sin, cos, log, min,* and *max*) to allow symbolic regression to build algebraic expressions with any combination of these operators. We performed a parameter tuning with different values for both single- and multi-objective approaches to find a suitable configuration that allows us to achieve a better exploration of the search space. Thus, we tested different values for SLP size, genetic operators (crossover and mutation), stopping criterion, etc. After that test, the experimental

settings that provided the best results in both approaches were: the population size of 180; allowing 15 rules for each SLP; and crossover and mutation probabilities of 90 % and 10 %, respectively. Finally, the stopping criteria was evaluating 40,000 solutions.

Each dataset was randomly divided into train (70 % of data) and test (remaining 30 %). The training data were used during the algorithm execution, and the test data were used once the algorithm finished and returned an algebraic expression, so that we could prevent over-fitting and validate the results in previously unseen data. Finally, we ran 30 executions for each scenario (algorithm and problem) with different random seed numbers, in order to carry out a statistical test that helped us determine if there exist significant differences between the results obtained.

4.1.3 Results and Discussion

The results obtained for each approach and dataset (train and test results) are shown in Table 2.1. The first column of this table describes the item evaluated for each approach, i.e., Median, Best (lowest), and Worst (highest) Mean Square Error (MSE), the average execution time of each approach (item *Time (s.)*), the average size of the resulting algebraic expression found by the algorithms (item *Size*), and the average number of parameters used by the resulting algebraic expressions (item *Parameters*). We highlight that we used the median error as statistical analysis estimator since the resulting MSE error distributions of the algorithms do not follow a normal distribution. In these cases, the mean error cannot be considered an appropriate metric, and it is replaced by the median value.

Columns 2–7 contain the results provided by both single- and multi-objective approaches for each benchmark algebraic expression (f_1–f_6) that was used to generate three datasets, and Columns 8–13 gather the results after modelling five datasets generated with each benchmark algebraic expression. Finally, Rows 3–17 describe the results in train data, whereas Rows 18–26 gather the results in test data. To give support to the analysis, we also provide box plots about the MSE distributions in the test sets for each single- and multi-objective approach (see Figure 2.3).

In a preliminary analysis of the test results in Table 2.1 and Figure 2.3, we observed that the single-objective approach achieved the lowest median MSE in five of six problems with three datasets, and all problems with five datasets. In addition, the best solution found was provided by the single-objective approach in both cases. Finally, the worst MSE also suggests that the single-objective algorithm performed better in all problems with three datasets, and in five of six problems with five datasets. After this preliminary analysis, we validated this assumption statistically, using a non-parametric Kruskal–Wallis (KW) test with 95 % confidence level to verify whether significant differences exist between the results found with each algorithm. Table 2.2 summarises the results of the test, and it contains the resulting p-value of the test for both training (Columns 2 and 3) and test data (Columns 4 and 5). If significant differences were found between the results found with the single- and multi-objective approaches (p-value $< 0{,}05$), then the results were marked: with "$+$" if the single-objective approach found a better solution; with "-" if the multi-objective algorithm was better; and with "x" if there were no significant differences between the solutions.

As shown in Table 2.2, our preliminary analysis was confirmed by results in Columns 4 and 5, since the single-objective approach improved significantly the multi-objective algorithm in all cases. After the statistical test was applied over the training error distributions, Table 2.2 confirms that the multi-objective approach was better in two of six problems with three datasets than the single-objective approach, considering training results; and there were no statistical differences in the remaining four problems. Regarding the problems with five datasets, the multi-objective approach was equivalent to the single-objective approach in three problems also considering training results, and the single-objective algorithm was better in the remaining three. On the other hand, the results provided in Table 2.1 as well as the statistical test results in Table 2.2 suggest that the single-objective algorithm improves performance over the multi-objective approach in the test data. This fact suggests that the multi-objective approach might be over-fitting the training data, therefore providing worse results in the test sets of each problem than the single-objective approach. This assumption is supported by the size of the resulting algebraic expressions, provided in Table 2.1. The single-objective algorithm could provide simpler (shorter) expressions, while the multi-objective approach provided more

complex expressions that could overfit training data and could not generalise well to all datasets. This also affected the training time of each method. As the multi-objective algorithm considered larger expressions, its evaluation was computationally more expensive than the single-objective algorithm (see item *Time (s.)* in Table 2.1).

Items	3 Datasets						5 Datasets					
	f_1	f_2	f_3	f_4	f_5	f_6	f_1	f_2	f_3	f_4	f_5	f_6
TRAIN												
Single objective												
Median	$3{,}12 \times 10^{-16}$	$1{,}53 \times 10^{-2}$	$1{,}89 \times 10^{8}$	$1{,}45 \times 10^{-7}$	4,92	$4{,}46 \times 10^{-5}$	$7{,}81 \times 10^{-10}$	$5{,}5 \times 10^{-2}$	$3{,}99 \times 10^{7}$	$6{,}27 \times 10^{-6}$	3,75	$5{,}55 \times 10^{-5}$
Best	$6{,}45 \times 10^{-18}$	$6{,}2 \times 10^{-4}$	$2{,}18 \times 10^{6}$	$1{,}37 \times 10^{-9}$	$1{,}2 \times 10^{-1}$	$2{,}62 \times 10^{-11}$	$2{,}97 \times 10^{-11}$	$2{,}1 \times 10^{-3}$	$8{,}77 \times 10^{6}$	$2{,}03 \times 10^{-9}$	$2{,}1 \times 10^{-1}$	$4{,}91 \times 10^{-11}$
Worst	$1{,}52 \times 10^{-14}$	$4{,}5 \times 10^{-2}$	$4{,}86 \times 10^{8}$	$6{,}61 \times 10^{-3}$	$1{,}76 \times 10^{1}$	$1{,}77 \times 10^{-3}$	$5{,}67 \times 10^{-8}$	$8{,}9 \times 10^{-2}$	$9{,}41 \times 10^{7}$	$6{,}33 \times 10^{-3}$	$1{,}83 \times 10^{1}$	$3{,}18 \times 10^{-3}$
Time (s.)	$1{,}03 \times 10^{3}$	$1{,}3 \times 10^{3}$	$1{,}81 \times 10^{3}$	$1{,}21 \times 10^{3}$	$1{,}35 \times 10^{3}$	$1{,}17 \times 10^{3}$	$2{,}03 \times 10^{3}$	$2{,}14 \times 10^{3}$	$2{,}14 \times 10^{3}$	$2{,}06 \times 10^{3}$	$2{,}16 \times 10^{3}$	$1{,}96 \times 10^{3}$
Size	8,93	10,6	11,76	9,96	11,16	11,23	9,63	10,5	10,76	10,33	11,56	10,9
Parameters	2,8	3,33	2,83	3,2	3,16	3,33	3,13	3,3	2,66	2,83	3,36	3,2
Multi objective												
Median	$4{,}6 \times 10^{-15}$	$3{,}1 \times 10^{-3}$	$8{,}49 \times 10^{6}$	$2{,}87 \times 10^{-7}$	$9{,}4 \times 10^{-1}$	$2{,}12 \times 10^{-6}$	$1{,}13 \times 10^{-8}$	8×10^{-2}	$4{,}43 \times 10^{7}$	$5{,}62 \times 10^{-7}$	$1{,}75 \times 10^{1}$	$3{,}95 \times 10^{-6}$
Best	$3{,}54 \times 10^{-18}$	$3{,}83 \times 10^{-5}$	$1{,}21 \times 10^{5}$	$1{,}36 \times 10^{-9}$	$1{,}09 \times 10^{-1}$	$2{,}58 \times 10^{-11}$	$2{,}92 \times 10^{-11}$	$1{,}2 \times 10^{-3}$	$1{,}11 \times 10^{6}$	2×10^{-9}	$7{,}2 \times 10^{-1}$	$4{,}86 \times 10^{-11}$
Worst	$2{,}59 \times 10^{-13}$	$4{,}3 \times 10^{-1}$	$1{,}32 \times 10^{9}$	$3{,}92 \times 10^{-2}$	$3{,}78 \times 10^{2}$	$1{,}19 \times 10^{-3}$	$8{,}14 \times 10^{-7}$	2,22	2×10^{9}	$2{,}54 \times 10^{-1}$	$3{,}07 \times 10^{2}$	$6{,}88 \times 10^{-3}$
Time (s.)	$1{,}38 \times 10^{3}$	$1{,}91 \times 10^{3}$	$2{,}58 \times 10^{3}$	$1{,}72 \times 10^{3}$	$3{,}12 \times 10^{3}$	$1{,}67 \times 10^{3}$	$2{,}43 \times 10^{3}$	$2{,}89 \times 10^{3}$	$4{,}47 \times 10^{3}$	$2{,}96 \times 10^{3}$	$5{,}49 \times 10^{3}$	$2{,}9 \times 10^{3}$
Size	11,76	12,1	12,76	12,7	12	13,23	7,56	9,96	11,76	12,36	10,33	13,1
Parameters	3	3,53	2,7	3,3	3,1	3,86	2,43	3,03	2,23	3,53	3,06	3,8
TEST												
Single-Objective												
Median	$2{,}01 \times 10^{-15}$	$2{,}42 \times 10^{-2}$	$8{,}21 \times 10^{4}$	$8{,}94 \times 10^{-8}$	5,52	$9{,}63 \times 10^{-5}$	$1{,}79 \times 10^{-9}$	$4{,}9 \times 10^{-2}$	$1{,}09 \times 10^{5}$	$4{,}38 \times 10^{-5}$	3,91	$3{,}95 \times 10^{-4}$
Best	$7{,}94 \times 10^{-18}$	$1{,}3 \times 10^{-3}$	$2{,}26 \times 10^{4}$	$7{,}19 \times 10^{-10}$	$1{,}4 \times 10^{-1}$	$2{,}44 \times 10^{-11}$	$4{,}21 \times 10^{-11}$	$3{,}2 \times 10^{-3}$	$3{,}67 \times 10^{4}$	$1{,}19 \times 10^{-9}$	$1{,}5 \times 10^{-1}$	$4{,}36 \times 10^{-11}$
Worst	$3{,}6 \times 10^{-1}$	6,18	$1{,}65 \times 10^{5}$	$4{,}7 \times 10^{-2}$	$9{,}38 \times 10^{2}$	1,97	$1{,}1 \times 10^{-5}$	$2{,}4 \times 10^{-1}$	$2{,}28 \times 10^{5}$	$1{,}67 \times 10^{-2}$	$1{,}23 \times 10^{3}$	$2{,}72 \times 10^{-1}$
Multi-Objective												
Median	$1{,}36 \times 10^{-12}$	$1{,}05 \times 10^{1}$	$1{,}65 \times 10^{5}$	5,3	$9{,}7 \times 10^{2}$	$2{,}6 \times 10^{-1}$	$1{,}59 \times 10^{-6}$	$1{,}83 \times 10^{1}$	$2{,}28 \times 10^{5}$	3,05	$9{,}33 \times 10^{2}$	$6{,}6 \times 10^{-1}$
Best	$1{,}36 \times 10^{-12}$	8,57	$1{,}45 \times 10^{5}$	1,08	$4{,}67 \times 10^{2}$	6×10^{-2}	$1{,}32 \times 10^{-6}$	$1{,}23 \times 10^{1}$	$1{,}97 \times 10^{5}$	1,65	$8{,}1 \times 10^{2}$	7×10^{-2}
Worst	3	$4{,}22 \times 10^{1}$	$1{,}66 \times 10^{5}$	5,3	$1{,}08 \times 10^{3}$	$1{,}52 \times 10^{1}$	$1{,}58 \times 10^{-6}$	$8{,}6 \times 10^{1}$	$2{,}29 \times 10^{5}$	7,51	$1{,}28 \times 10^{3}$	$6{,}7 \times 10^{-1}$

Tabla 2.1: Results of Single and Multi objective approaches in benchmark train and test datasets

	Train		Test	
	3 Datasets	**5 Datasets**	**3 Datasets**	**5 Datasets**
f_1	$1{,}7 \times 10^{-1}$ (x)	$4{,}9 \times 10^{-3}$ (+)	$2{,}87 \times 10^{-9}$ (+)	$1{,}39 \times 10^{-10}$ (+)
f_2	$1{,}3 \times 10^{-1}$ (x)	$2{,}8 \times 10^{-2}$ (+)	$2{,}83 \times 10^{-11}$ (+)	$2{,}86 \times 10^{-11}$ (+)
f_3	$1{,}4 \times 10^{-2}$ (-)	$7{,}9 \times 10^{-1}$ (x)	$3{,}95 \times 10^{-11}$ (+)	$1{,}22 \times 10^{-10}$ (+)
f_4	$8{,}01 \times 10^{-1}$ (x)	$8{,}01 \times 10^{-1}$ (x)	$1{,}67 \times 10^{-11}$ (+)	$2{,}62 \times 10^{-11}$ (+)
f_5	$5{,}8 \times 10^{-1}$ (x)	$2{,}8 \times 10^{-4}$ (+)	$2{,}79 \times 10^{-10}$ (+)	$2{,}41 \times 10^{-11}$ (+)
f_6	7×10^{-3} (-)	$1{,}9 \times 10^{-1}$ (x)	$7{,}55 \times 10^{-11}$ (+)	$1{,}81 \times 10^{-11}$ (+)

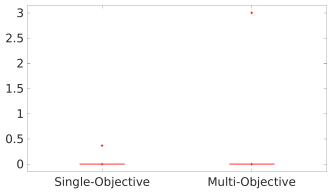
Tabla 2.2: Statistical tests to compare algorithms in the results of all benchmark algebraic expressions

To conclude the analysis of results in this section, we note that the single-objective approach could find better solutions than the multi-objective approach in terms of generalisation, without any regards to the number of datasets for each problem. In this way, the multi-objective approach overfit the training data, especially when the number of objectives (datasets) was low.

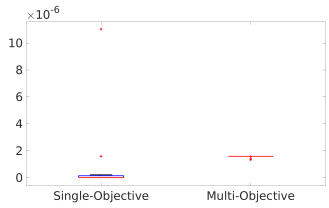
4.2 Experimentation with Real Data

With the approaches validated under a controlled environment, this section describes the testing with real data. As problem statement, we were provided with a set of energy consumption time series of different buildings (one time series x_i for each building i), together with an additional time series of exogenous data with the ambient temperature T. We name the energy consumption of a building i at time instant t as $x_i(t)$, and the temperature at time instant t as $T(t)$. All buildings are located in the same area, so that the temperature time series is the same for all buildings.

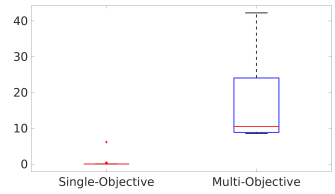
Our goal was to find a single algebraic expression f such as $x_i(t+1) = f(x_i(t), x_i(t-1), \cdots, x_i(t-h), T(t+1), T(t), \cdots, T(t-h+1), w_i)$, which can provide us with an approximation of the next energy consumption value of any building, $x_i(t+1)$, considering previous values of the same energy consumption time series up to a time horizon h, and the h previous values of the temperature. The resulting algebraic expression that models the energy consumption time series must be the same for all buildings, except for the parameters W_i, which are different for each



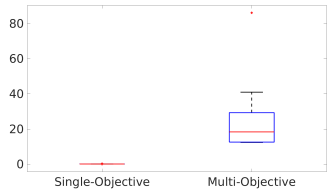
(a) Box plots for f_1 (three outputs)



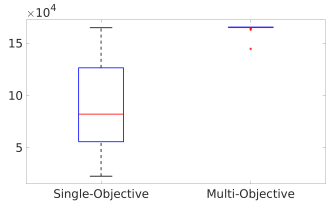
(b) Box plots for f_1 (five outputs)



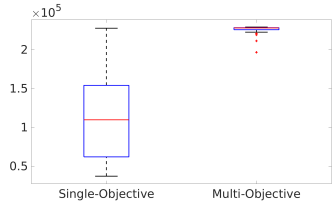
(c) Box plots for f_2 (three outputs)



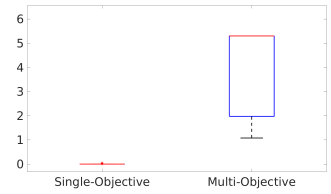
(d) Box plots for f_2 (five outputs)



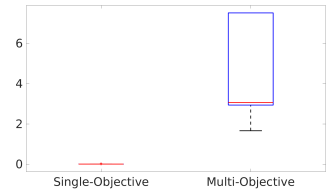
(e) Box plots for f_3 (three outputs)



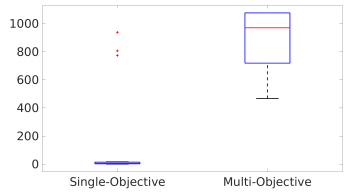
(f) Box plots for f_3 (five outputs)



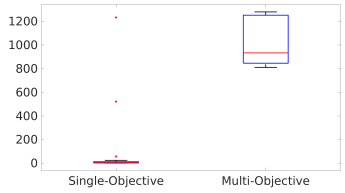
(g) Box plots for f_4 (three outputs)



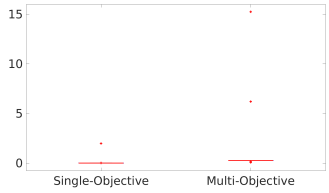
(h) Box plots for f_4 (five outputs)



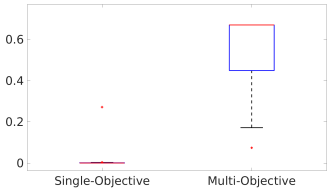
(i) Box plots for f_5 (three outputs)



(j) Box plots for f_5 (five outputs)



(k) Box plots for f_6 (three outputs)



(l) Box plots for f_6 (five outputs)

Figura 2.3: Box plots of accuracy for each algorithm and benchmark algebraic expression in test data.

building.

This could be possible only if the data series of all buildings are not independent and there exists a relationship among them. For this reason, it was important to check as prerequisite that the correlation coefficient of all time series in a dataset is medium or high, as a measurement of dependency (not necessarily causality) between all time series. If this hypothesis were confirmed, then the proposed methods would be expected to find a generalisation of that behavior in terms of a parameterised general algebraic expression able to predict the energy consumption of each building.

4.2.1 Data Acquisition

We used a dataset that contains the energy consumption of seven different buildings of University of Granada (south of Spain) from March 2013 to October 2015, named from B_1 to B_7 for confidentiality reasons. Each building is equipped with a set of sensors that monitor their energy consumption (kW/h) hourly. A Building Automation System (BAS) is responsible to monitor the energy consumption measured for each sensor of each building and store all data in a database.

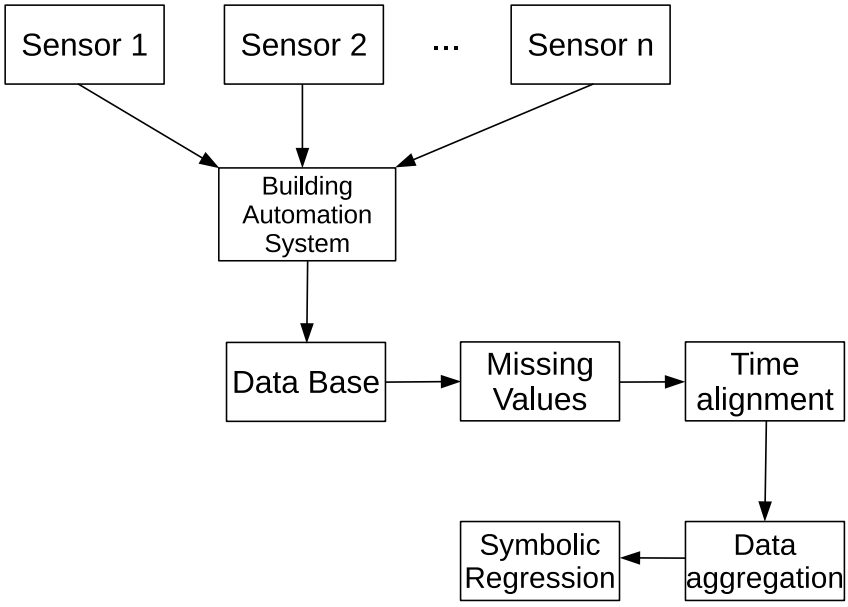


Figura 2.4: Building Automation System (BAS) and data preprocessing.

The raw data stored in the database needed to be preprocessed because there could be missing data due to light cuts, sensor failures, etc. The data preprocessing process used in this experimentation is shown in Figure 2.4. The first step of this preprocessing stage consisted in seeking missing values and interpolating each of them (which are 5 % of the data). After that, a time alignment was necessary to obtain the data consumption in the same temporal range. Besides, since this experiment attempted to predict the energy consumption of different buildings using the data of previous days, it was necessary to calculate the energy consumption for each weekday

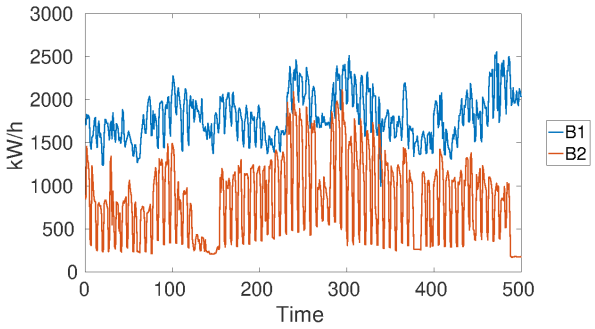
as the aggregation (addition) of the kW consumed in 24 h of the same day. Finally, to work with uniform data, we normalised the data in the interval $[0{,}0 - 1{,}0]$ (see Equation (2.26), where v_i is the response value, v_{min} is the minimum response observed, v_{max} is the maximum response observed and $r_{normalised}$ is the normalised response value). Once all data were preprocessed, we transformed the univariate dataset into a multivariate dataset, with eight dimensions, each one for a weekday plus the weekly temperature.

$$r_{normalized} = \frac{v_i - v_{min}}{v_{max} - v_{min}} \tag{2.26}$$

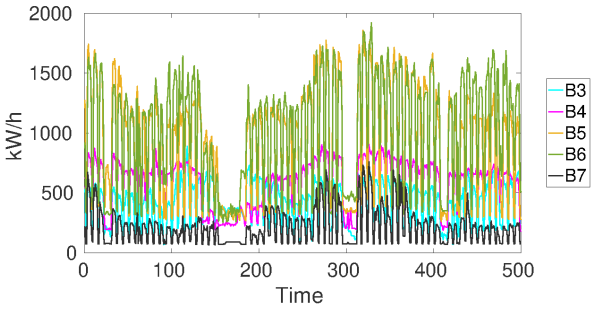
Once all data were preprocessed, it was necessary to carry out a preliminary correlation study between the energy consumption of all buildings to know if energy consumption data series are medium or highly correlated as the initial hypothesis to apply the proposed models. As the labels *low*, *medium* or *high* correlation levels are subjective and problem dependent, in our research, we considered low correlations with values less than 0,3, whereas correlation values between 0,3 and 0,7 were considered to medium correlation and values higher than 0.7 were considered highly correlated. As a result of this study, we found two clusters with medium or high correlation coefficients. Buildings B_1 and B_2 formed a cluster, and the second cluster was composed of Buildings B_3–B_7. An example of the energy consumption data series of each set of buildings is shown in Figure 2.5.

Figures 2.6a and 2.6b show the correlation plot matrices of the energy consumption time series for each building. The diagonal of the plot matrices gathers the histogram, to know how the energy consumption is distributed in each building. Then, the remaining subfigures show the scatter plot between the energy consumption of each pair of buildings. A subfigure in a cell (i,j) shows the relationships between building at row i (y-axis) and the building at column j (x-axis), plus the correlation coefficient R highlighted in red, whose values are in the interval $[-1{,}0, 1{,}0]$. Values of R closer to 1,0 mean a positive correlation, whereas values near to $-1{,}0$ suggest negative correlation. Finally, values closer to 0,0 suggest low correlation.

As shown in Figures 2.6a and 2.6b, there are medium ($0{,}3 < R < 0{,}7$) and high ($R > 0{,}7$)



(a) Energy consumption data series of Buildings B_1 and B_2

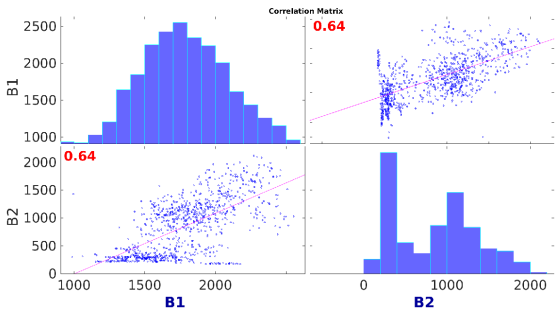


(b) Energy consumption data series of Buildings B_3 and B_7

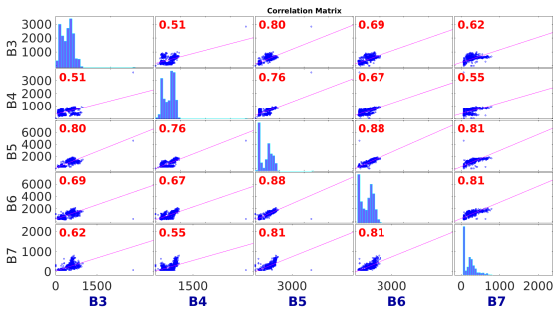
Figura 2.5: Energy consumption data series of cluster of buildings E_1 and E_2 during 500 days.

positive correlations between the energy consumption of the selected buildings in each cluster. Thus, we were provided with two different scenarios to test our approaches: One being composed of two datasets with Buildings B_1 and B_2 and another one with five datasets with Buildings B_3–B_7. To ease the explanation in the experimental section, we named each cluster of buildings as E_1 (for Buildings B_1 and B_2) and E_2 (for Buildings B_3–B_7).

In the experiments, we divided each dataset into two subsets: training (70 %) and test (30 %), to avoid over-fitting in the solutions found. In this way, the training set was used to build the model of each approach and the test set was used to check the quality of the solutions found, following the same methodology presented in Section 4.1. Therefore, to verify the robustness of each approach, we show in Section 4.2.3 the results obtained in each training and test subsets.



(a) Energy consumption correlation between days for Buildings B_1 and B_2.



(b) Energy consumption correlation between days for Buildings B_3–B_7.

4.2.2 Experimental Settings

To define the experimental settings of each approach used in a real scenario with energy consumption data, we performed a trial-and-error procedure to find the optimal parameters for the algorithm. As is described, we allowed a maximum size of 15 operations (SLP size), which can use a set of 12 mathematical operators (*+, -, *, /, sqrt, pow, exp, sin, cos, log, min,* and *max*). Thus, as mentioned above, the input variables for each approach were composed of the energy consumption registered in the previous h weekdays ($h = 6$ for this experimentation) and the temperature registered for the last weekday. In addition, we permitted a set of five parameters $W_i = (w_i^1, w_i^2, w_i^3, w_i^4, w_i^5)$ for each building, which were estimated for each building during the algorithm execution. Then, the crossover and mutation probabilities were established as 90 % and 10 %, respectively. Finally, both approaches ran 30 times with different random seeds to analyse the results statistically. The stopping criteria used for each algorithm was to have 40,000 solutions evaluated.

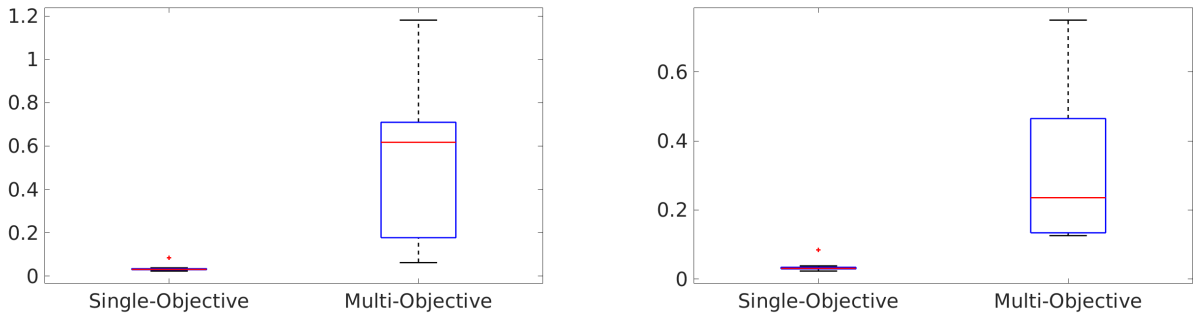
4.2.3 Results and Discussion

The results of this experimentation are organised in Table 2.3. In this table, Column 1 shows the item evaluated for each single- and multi-objective approach. Thus, rows labeled as *Median* describe the median Mean Square Error (MSE) obtained in the 30 experiments for both approaches. Rows named as *Best* and *Worst* gather the minimum and maximum MSE obtained in the 30 runnings. Then, the average time needed to obtain a solution by each approach is shown in rows labeled as *Time (s.)*. Rows labeled as *Size* encode the average size of each algebraic expression found in each run (calculated as the number of mathematical operators used in the algebraic expression found), and the number of parameters used in the algebraic expression found with each single- and multi-objective approach are also shown in rows tagged as *Parameters*. Finally, Columns 2 and 3 of the table gather the results obtained by both single- and multi-objective approach for each cluster of buildings (E_1 and E_2), respectively, in

	Train		Test	
	E_1	E_2	E_1	E_2
	Single-Objective			
Median	$2{,}3 \times 10^{-2}$	$6{,}7 \times 10^{-2}$	$3{,}1 \times 10^{-2}$	$7{,}2 \times 10^{-2}$
Best	$1{,}6 \times 10^{-2}$	$5{,}5 \times 10^{-2}$	$2{,}3 \times 10^{-2}$	$5{,}3 \times 10^{-2}$
Worst	3×10^{-2}	$8{,}1 \times 10^{-2}$	$8{,}4 \times 10^{-2}$	$2{,}4 \times 10^{-1}$
Time	$1{,}02 \times 10^{3}$	$2{,}53 \times 10^{3}$	-	-
Size	10.83	11.56	-	-
Parameters	2.43	3	-	-
	Multi-Objective			
Median	$1{,}9 \times 10^{-2}$	$6{,}8 \times 10^{-2}$	$6{,}17 \times 10^{-1}$	$2{,}35 \times 10^{-1}$
Best	$1{,}4 \times 10^{-2}$	$5{,}1 \times 10^{-2}$	$6{,}2 \times 10^{-2}$	$1{,}26 \times 10^{-1}$
Worst	$3{,}7 \times 10^{-2}$	$1{,}1 \times 10^{-1}$	1,18	$7{,}49 \times 10^{-1}$
Time	$1{,}61 \times 10^{3}$	$4{,}12 \times 10^{3}$	-	-
Size	11.03	4.73	-	-
Parameters	2.9	1.46	-	-
KW Test	$4{,}49 \times 10^{-5}$ (-)	0,35 (x)	$2{,}9 \times 10^{-11}$ (+)	$1{,}21 \times 10^{-10}$ (+)

Tabla 2.3: Results for cluster of buildings E_1 and E_2 in train and test data

training data and Columns 3 and 4 show the results obtained by both single- and multi-objective approach in test data. Finally, last row of Table 2.3 describes the results of Kruskal–Wallis test. To provide a better analysis of the results of Table 2.3 we have also included the box plots of the MSE distributions in the test sets for both single- and multi-objective approach in Figure 2.7.



(a) Box plot for cluster E_1 in test data

(b) Box plot for cluster E_2 in test data

Figura 2.7: Box plots of accuracy for both single- and multi-objective approach and cluster of buildings.

In a first analysis of the results of both single- and multi-objective approach in test data in Table 2.3, we may observe that the single-objective approach obtained the lowest MSE median value in both clusters of buildings. Moreover, since the single-objective approach provided the

best solution in terms of best MSE in both scenarios and the multi-objective approach found the worst solution in both sets of buildings E_1 and E_2, this preliminary analysis suggests that the single-objective approach is potentially better than the multi-objective approach.

To give support of this analysis, we again performed a non-parametric Kruskal–Wallis test (KW) with a 95 % confidence level to statistically validate if there are significant differences between each approach. The results of the KW test are presented in the last row of Table 2.3, where Columns 2–3 and 4–5 show the resulting p-value of the KW test for each cluster of buildings and approaches for both training and test datasets. If significant differences were found between the results obtained with the single- and multi-objective approach (p-value $< 0{,}05$), then the results were marked with the symbol $+$ if the single-objective approach was better; with the symbol - if the multi-objective approach provided better solutions; and with the symbol x if both approaches were equivalents.

Regarding the results of the KW test in Table 2.3 in test data (Columns 4 and 5), the single-objective approach provided better results than the multi-objective approach in all cases, thus confirming our hypothesis that the single-objective is better than the multi-objective approach in test data. Box plots in Figure 2.7 visually confirm that there are significant differences between the results found by the multi-objective approach in the problems of buildings in E_1 and buildings in E_2, respectively. Nevertheless, regarding the results of the KW test in train data (Columns 2 and 3 of the last row of Table 2.3), we may observe that the multi-objective approach provided better solutions for E_1 (two objectives), and no significant differences between the single- and multi-objective approach were found in E_2 (five objectives).

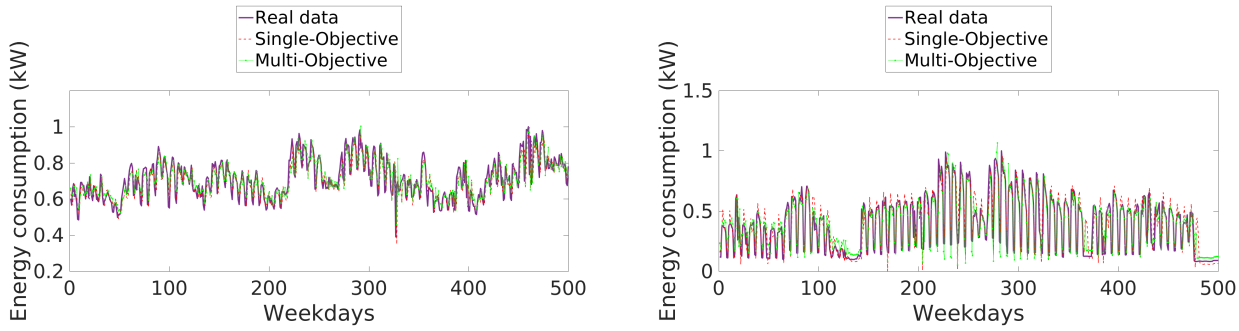
If we analyse the results of each approach in train data in Table 2.3 regarding the median and best MSE value, the multi-objective approach provided the lowest value in the cluster of buildings E_1, and the single-objective approach achieved better results in the second cluster of buildings E_2. Nevertheless, in both scenarios, the multi-objective approach performed worse in terms of MSE value. This fact, together with the results of the KW test, suggests that the multi-objective approach suffered over-fitting in the training procedure, and therefore it shows the same performance we found in synthetic datasets in Section 4.1. Therefore, as an example of

the solutions found by both approaches in real energy consumption data, Equations (2.27) and (2.28) gather the best SLP found by each single- and multi-objective approach, respectively. We also want to remark that, although we allowed SLPs with size equal to 15, the equations shown are the useful code for each SLP. Then, the algebraic expression encoded by each SLP can be calculated by generating the last rule of each equation. Finally, regarding the average algebraic expression size shown in Table 2.3, we cannot conclude that there are significant differences between both single- and multi-objective approaches, since the multi-objective approach found smaller solutions for E_2, whereas the single-objective approach provided smaller solutions in the first cluster of buildings E_1. Nevertheless, regarding the computational time, the multi-objective approach was considerably higher than the single-objective approach in both cases, which may be a consequence of the Pareto-optimal solution search in multi-objective algorithms that need more computational time, as argued in Section 2.

$$
E_1 \begin{cases} U_6 \to min(w_4, x_1) \\ U_7 \to x_5/U_6 \\ U_8 \to w_2 + U_7 \\ U_9 \to max(x_6, U_8) \\ U_{10} \to pow(U_9, x_2) \\ U_{11} \to w_2 + U_{10} \\ U_{12} \to max(x_6, U_{11}) \\ U_{13} \to w_2 + U_{12} \\ U_{14} \to pow(x_6, U_{13}) \\ U_{15} \to w_2 + U_{14} \end{cases} \qquad E_2 \begin{cases} U_2 \to exp(w_2) \\ U_3 \to x_6 - x_1 \\ U_4 \to pow(U_2, U_3) \\ U_5 \to cos(x_1) \\ U_6 \to pow(U_4, U_5) \\ U_7 \to w_4 * U_6 \\ U_8 \to w_4 * U_7 \\ U_9 \to w_3 + x_5 \\ U_{10} \to U_8 * U_9 \\ U_{11} \to U_{10} - x_5 \\ U_{12} \to x_4 + x_2 \\ U_{13} \to pow(U_{11}, U_{12}) \\ U_{14} \to pow(x_6, U_{13}) \\ U_{15} \to U_{14} * w_1 \end{cases} \tag{2.27}
$$

$$E_1 \begin{cases} U_1 \rightarrow w_3 + x_4 \\ U_8 \rightarrow exp(U_1) \end{cases} \qquad E_2 \begin{cases} U_2 \rightarrow tan(x_1) \end{cases} \tag{2.28}$$

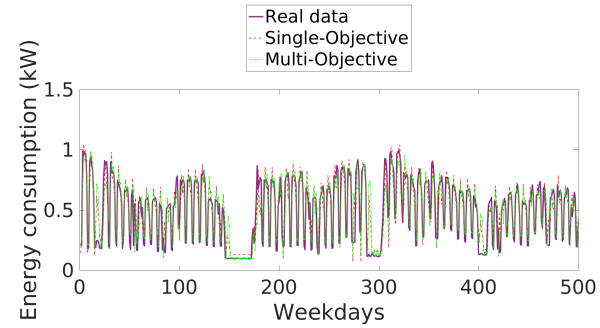
To conclude with the analysis of the results, Figures 2.8 and 2.9 show in colour purple the original dataset of the cluster of buildings E_1 (Buildings B_1 and B_2) and E_2 (Buildings B_3–B_7), and the results of the predicted data from both single- and multi-objective approaches in red and green colours, respectively. The results plotted with each approach were obtained with the best algebraic expression found from each approach (see Equations (2.27) and (2.28)). In this way, we may conclude that not only was the single-objective approach able to find better solutions than the multi-objective approach in terms of generalisation, without regards the number of datasets for each problem, but also that the provided algebraic expression is an accurate model of energy consumption for all buildings in the same compound E_1 or E_2.



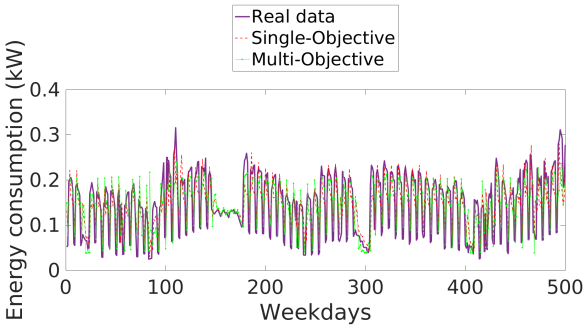
(a) Building B_1 (b) Building B_2

Figura 2.8: Real and predicted energy consumption from both single- and multi-objective approaches in cluster of buildings E_1.

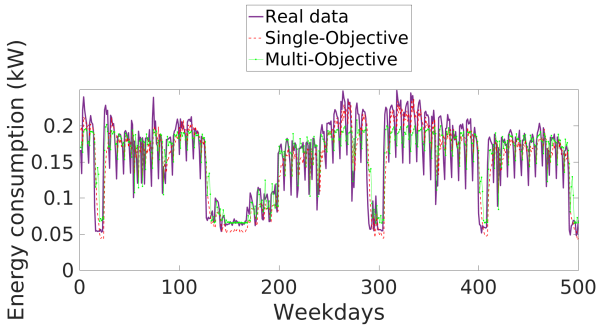
As a general summary of this experimental section, we may conclude that the approach in this manuscript is able to find a generalised regression hypothesis able to explain multiple datasets. More specifically, when applied to real energy consumption data series, the proposal could find a general explanation about how energy consumption evolves in different buildings, and to provide a single formula that is valid to explain the energy consumption behaviour of all buildings.



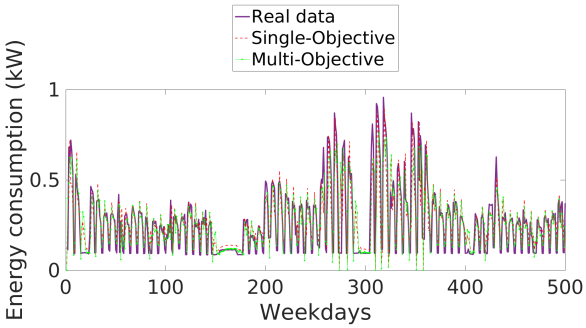
(a) Building B_3



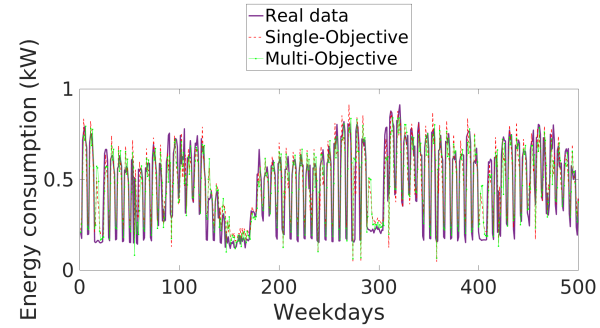
(b) Building B_4



(c) Building B_5



(d) Building B_6



(e) Building B_7

Figura 2.9: Real and predicted energy consumption from both single- and multi-objective approaches in cluster of buildings E_2.

5 Conclusions

This work describes our new formulation of energy consumption forecasting in the case where multiple energy consumption time series are under study. To use such approach, we require that all the input time series have a medium or high correlation and, consequently, that a common ground behaviour that can explain (at least partially) all time series exists. As we have not found any other approach in the literature that addresses this particular problem, we developed an experimentation to test the approach under a controlled environment in the laboratory, and

then using real data. According to the experimental results, we can confirm that the results in both synthetic and real data are consistent, and that the single-objective algorithm proposed is the best option to find such algebraic expression. Besides, we observed that the performance of Pareto-based multi-objective algorithms decreases as the number of objectives increases, since both experiments showed that the multi-objective algorithm was significantly worse than the single-objective approach when the number of datasets was high. A final observation is that there were significant differences between the error distribution in training and test datasets, which suggest us that the multi-objective approach may over-fit the training data.

In summary, we have provided the scientific community with a new tool to analyse and generalise multiple datasets, and to perform data mining over energy consumption modelling problems. Thus, the proposed model opens new opportunities not only in energy consumption forecasting, but also in other topics such as time series data summarisation, energy profile mining, and anomaly detection. Future works attempt to apply ontologies not only to automatically select the most affordable Pareto-solution regarding semantic knowledge, but also to reduce the search space including knowledge about algebraic expressions.

Acknowledgements

This work has been supported by the project TIN201564776-C3-1-R.

References

[Afr+17] Abdul Afram y col. «Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system». En: *Energy and Buildings* 141 (2017), págs. 96-113. DOI: 10.1016/j.enbuild.2017.02.012.

[Amb+17] Khuram Pervez Amber y col. «Energy Consumption Forecasting for University Sector Buildings». En: *Energies* 10.10 (2017). DOI: 10.3390/en10101579.

[AMR17] Muhammad Waseem Ahmad, Monjur Mourshed y Yacine Rezgui. «Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption». En: *Energy and Buildings* 147 (2017), págs. 77-89. DOI: 10.1016/j.enbuild.2017.04.038.

[Asc+17] Fabrizio Ascione y col. «CASA, cost-optimal analysis by multi-objective optimisation and artificial neural networks: A new framework for the robust assessment of cost-optimal energy retrofit, feasible for any building». En: *Energy and Buildings* 146 (2017), págs. 200-219. DOI: 10.1016/j.enbuild.2017.04.069.

[BAB14] M.R. Braun, H. Altan y S.B.M. Beck. «Using regression analysis to predict the future energy consumption of a supermarket in the UK». En: *Applied Energy* 130 (2014), págs. 305-313.

[Bal+13] Bharathan Balaji y col. «Sentinel: Occupancy Based HVAC Actuation Using Existing WiFi Infrastructure Within Commercial Buildings». En: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. New York, NY, USA, 2013, 17:1-17:14.

[BAN02] Maumita Bhattacharya, Ajith Abraham y Baikunth Nath. «A Linear Genetic Programming Approach for Modelling Electricity Demand Prediction in Victoria». En: *Hybrid Information Systems*. Heidelberg: Physica-Verlag HD, 2002, págs. 379-393.

[Beh+12] R. Behera y col. «An Application of Genetic Programming for Power System Planning and Operation». En: *International Journal on Control System and Instrumentation, March*. 2012, págs. 15-20.

[Ber17] Umberto Berardi. «A cross-country comparison of the building energy consumptions and their trends». En: *Resources, Conservation and Recycling* 123 (2017), págs. 230-241. DOI: https://doi.org/10.1016/j.resconrec.2016.03.014.

[BRF16] M.A. Rafe Biswas, Melvin D. Robinson y Nelson Fumo. «Prediction of residential building energy consumption: A neural network approach». En: *Energy* 117 (2016), págs. 84-92. DOI: https://doi.org/10.1016/j.energy.2016.10.066.

[CPB17] Alfonso Capozzoli, Marco Savino Piscitelli y Silvio Brandi. «Mining typical load profiles in buildings to support energy management in the smart city context». En: *Energy Procedia* 134 (2017), págs. 865-874. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2017.09.545. URL: http://www.sciencedirect.com/science/article/pii/S187661021734674X.

[CT14] Jui-Sheng Chou y Abdi Suryadinata Telaga. «Real-time detection of anomalous power consumption». En: *Renewable and Sustainable Energy Reviews* 33 (2014), págs. 400-411. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2014.01.088. URL: http://www.sciencedirect.com/science/article/pii/S1364032114001142.

[CW17b] Wenqiang Cui y Hao Wang. «A New Anomaly Detection System for School Electricity Consumption Data». En: *Information* 8.4 (2017). DOI: 10.3390/info8040151.

[Deb+02] K. Deb y col. «A fast and elitist multiobjective genetic algorithm: NSGA-II». En: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), págs. 182-197. DOI: 10.1109/4235.996017.

[Deb+17] Chirag Deb y col. «A review on time series forecasting techniques for building energy consumption». En: *Renewable and Sustainable Energy Reviews* 74 (2017), págs. 902-924. DOI: https://doi.org/10.1016/j.rser.2017.02.085.

[GA17] Juan A. Gomez y Miguel F. Anjos. «Power capacity profile estimation for building heating and cooling in demand-side management». En: *Applied Energy* 191 (2017), págs. 492-501. DOI: 10.1016/j.apenergy.2017.01.064.

[GK17] Georgios Gourlis e Iva Kovacic. «Building Information Modelling for analysis of energy efficient industrial buildings – A case study». En: *Renewable and Sustainable Energy Reviews* 68 (2017), págs. 953-963. DOI: 10.1016/j.rser.2016.02.009.

[GNC16] Jun Guan, Natasa Nord y Shuqin Chen. «Energy planning of university campus building complex: Energy usage and coincidental analysis of individual buildings with a case study». En: *Energy and Buildings* 124 (2016), págs. 99-111. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.04.051. URL: http://www.sciencedirect.com/science/article/pii/S0378778816303164.

[Has10] Ghada Nasr Aly Hassan. «Multiobjective genetic programming for financial portfolio management in dynamic environments». Tesis doct. University College London, UK, 2010.

[HHS11] Mohamed Hamdy, Ala Hasan y Kai Siren. «Applying a multi-objective optimization approach for Design of low-emission cost-effective dwellings». En: *Building and Environment* 46.1 (2011), págs. 109-123. DOI: 10.1016/j.buildenv.2010.07.006.

[Höl+14] Jan Höller y col. «Chapter 13 - Commercial Building Automation». En: *From Machine-To-Machine to the Internet of Things*. Academic Press, 2014, págs. 269-279.

[Jai+14] Rishee K. Jain y col. «Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy». En: *Applied Energy* 123 (2014), págs. 168-178. DOI: https://doi.org/10.1016/j.apenergy.2014.02.057.

[JGB92] Wilfried Jakob, Martina Gorges-Schleuter y Christian Blume. «Application of Genetic Algorithms to Task Planning and Learning». En: *Parallel Problem Solving from Nature 2, PPSN-II, Brussels, Belgium, September 28-30, 1992*. 1992, págs. 293-302.

[JSZ15] Radisa Z. Jovanovic, Aleksandra A. Sretenovic y Branislav D. Zivkovic. «Ensemble of various neural networks for prediction of heating energy consumption». En: *Energy and Buildings* 94 (2015), págs. 189-199. DOI: 10.1016/j.enbuild.2015.02.052.

[KC99] J. Knowles y D. Corne. «The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation». En: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. Vol. 1. 1999, págs. 98-105. DOI: 10.1109/CEC.1999.781913.

[KCV02] N Keerativuttiumrong, Nachol Chaiyaratana y Vara Varavithya. «Multiobjective Co-operative Co-evolutionary Genetic Algorithm». En: *Parallel Problem Solving from Nature-PPSN VII* 2439 (2002), págs. 288-297.

[Koz94] John R. Koza. «Genetic programming as a means for programming computers by natural selection». En: *Statistics and Computing* 4.2 (1994), págs. 87-112. DOI: 10.1007/BF00175355.

[Lan98] W. B. Langdon. «Genetic Programming — Computers Using "Natural Selection" to Generate Programs». En: *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!* Boston, MA: Springer US, 1998, págs. 9-42.

[Lü+15] Xiaoshu Lü y col. «Modeling and forecasting energy consumption for heterogeneous buildings using a physical–statistical approach». En: *Applied Energy* 144 (2015), págs. 261-275. DOI: https://doi.org/10.1016/j.apenergy.2014.12.019.

[MA04] R.T. Marler y J.S. Arora. «Survey of multi-objective optimization methods for engineering». En: *Structural and Multidisciplinary Optimization* 26.6 (2004), págs. 369-395. DOI: 10.1007/s00158-003-0368-6.

[MA10] R. Timothy Marler y Jasbir S. Arora. «The weighted sum method for multi-objective optimization: new insights». En: *Structural and Multidisciplinary Optimization* 41.6 (2010), págs. 853-862. DOI: 10.1007/s00158-009-0460-7.

[Mol+17] Miguel Molina-Solana y col. «Data science for building energy management: A review». En: *Renewable and Sustainable Energy Reviews* 70 (2017), págs. 598-609. DOI: https://doi.org/10.1016/j.rser.2016.11.132.

[MWB95] B. McKay, M. J. Willis y G. W. Barton. «Using a tree structured genetic algorithm to perform symbolic regression». En: *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. 1995, págs. 487-492.

[NF08] Alberto Hernandez Neto y Flávio Augusto Sanzovo Fiorelli. «Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption». En: *Energy and Buildings* 40.12 (2008), págs. 2169-2176. DOI: https://doi.org/10.1016/j.enbuild.2008.06.013.

[Nic+15] M. Nicolau y col. «Guidelines for defining benchmark problems in Genetic Programming». En: *2015 IEEE Congress on Evolutionary Computation (CEC)*. 2015, págs. 1152-1159. DOI: 10.1109/CEC.2015.7257019.

[Rue+18b] Ramón Rueda Delgado y col. «A Comparison Between NARX Neural Networks and Symbolic Regression: An Application for Energy Consumption Forecasting». En: *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*. 2018, págs. 16-27. ISBN: 978-3-319-91479-4.

[Rui+16] L.G.B. Ruiz y col. «An Application of Non-Linear Autoregressive Neural Networks to Predict Energy Consumption in Public Buildings». En: *Energies* 9 (ago. de 2016), pág. 684. DOI: 10.3390/en9090684.

[San16] Mat Santamouris. «Innovating to zero the building sector in Europe: Minimising the energy consumption, eradication of the energy poverty and mitigating the local climate change». En: *Solar Energy* 128 (2016), págs. 61-94. DOI: https://doi.org/10.1016/j.solener.2016.01.021.

[Sch85] J. David Schaffer. «Multiple Objective Optimization with Vector Evaluated Genetic Algorithms». En: *Proceedings of the 1st International Conference on Genetic Algorithms*. 1985, págs. 93-100.

[SD94] N. Srinivas y K. Deb. «Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms». En: *Evolutionary Computation* 2.3 (1994), págs. 221-248. DOI: 10.1162/evco.1994.2.3.221.

[SZ17] Aulon Shabani y Orion Zavalani. «Hourly Prediction of Building Energy Consumption: An Incremental ANN Approach». En: *European Journal of Engineering Research and Science* 2 (2017), pág. 27. DOI: 10.24018/ejers.2017.2.7.397.

[Wil+97] Mark Willis y col. «Genetic programming: An introduction and survey of applications». En: *Second International Conference On Genetic Algorithms In Engineering Systems: Innovations And Applications, Glasgow, UK*. 1997, págs. 314-319. DOI: 10.1049/cp:19971199.

[WS17] Zeyu Wang y Ravi S. Srinivasan. «A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models». En: *Renewable and Sustainable Energy Reviews* 75 (2017), págs. 796-808. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.10.079. URL: http://www.sciencedirect.com/science/article/pii/S1364032116307420.

[Wu+17] Raphael Wu y col. «Multiobjective optimisation of energy systems and building envelope retrofit in a residential community». En: *Applied Energy* 190 (2017), págs. 634-649. DOI: 10.1016/j.apenergy.2016.12.161.

[Yan+16] Guangfei Yang y col. «A comparative study on the influential factors of China's provincial energy intensity». En: *Energy Policy* 88 (2016), págs. 74-85. DOI: 10.1016/j.enpol.2015.10.011.

[Yan+17] Ming-Der Yang y col. «Multiobjective optimization design of green building envelope material using a non-dominated sorting genetic algorithm». En: *Applied Thermal Engineering* 111 (2017), págs. 1255-1264. DOI: 10.1016/j.applthermaleng.2016.01.015.

[YBS17] B. Yildiz, J.I. Bilbao y A.B. Sproul. «A review and analysis of regression and machine learning models on commercial building electricity load forecasting».

En: *Renewable and Sustainable Energy Reviews* 73 (2017), págs. 1104-1122. DOI: 10.1016/j.rser.2017.02.023.

[YS14] F. Yasmeen y M. Sharif. «Forecasting electricity consumption for Pakistan». En: *Int J Emerg Technol Adv Eng* 4 (2014), págs. 496-503.

[Zha+14] Jie Zhao y col. «Occupant behavior and schedule modeling for building energy simulation through office appliance power consumption data mining». En: *Energy and Buildings* 82 (2014), págs. 341-355. DOI: 10.1016/j.enbuild.2014.07.033.

[ZLT01] E. Zitzler, M. Laumanns y L. Thiele. «SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization». En: *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*. Athens, Greece: International Center for Numerical Methods in Engineering, 2001, págs. 95-100.

[ZT99] E. Zitzler y L. Thiele. «Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach». En: *IEEE Transactions on Evolutionary Computation* 3.4 (nov. de 1999), págs. 257-271. DOI: 10.1109/4235.797969.

2.4. A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming

- **Referencia**: R.Rueda, M.P.Cuéllar, L.G.B.Ruiz, M.C.Pegalajar. A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming
- **Estado**: En revisión

A similarity measure for Straight Line Programs and its application to control diversity in Genetic Programming

R.Rueda, M.P.Cuéllar, L.G.B.Ruíz, M.C.Pegalajar

Department of Computer Science and Artificial Intelligence, University of Granada, Granada,

C/. Pdta. Daniel Saucedo Aranda s.n., 18071 Granada, Spain

Abstract

Finding a balance between diversity and convergence plays an important role in evolutionary algorithms to avoid premature convergence and to perform a better exploration of the search space. In the case of Genetic Programming, and more specifically for symbolic regression problems, different mechanisms have been devised to control diversity, ranging from novel crossover and/or mutation procedures to the design of distance measures that help genetic operators to increase diversity in the population. In this paper, we start from previous works where Straight Line Programs are used as an alternative representation to expression trees for symbolic regression, and develop a similarity measure based on edit distance in order to determine how different the Straight Line Programs in the population are. This measure is used in combination with the CHC algorithm strategy to control diversity in the population, and therefore to avoid local optima to solve symbolic regression problems. The proposal is first validated in a controlled scenario of benchmark datasets and it is compared with previous approaches to promote diversity in Genetic Programming. After that, the approach is also evaluated in a real world dataset of energy consumption data from a set of buildings of the University of Granada.

Keywords: Edit Distance, Symbolic Regression, Genetic Programming, Straight Line Program

1 Introduction

In evolutionary computation, diversity and convergence play an important role in the exploration of the search space. As many authors argued [BGK04; BR07; CLY09], finding a balance between diversity and convergence is critical in genetic algorithms, since premature convergence causes the end of the evolution in local optima, while an uncontrolled divergence may reduce exploitation of the search space. Two main mechanisms that guide the exploration of the search space in genetic algorithms have been identified in the literature [SE10]: *Variation*, which promotes diversity, and *Selection* to reinforce convergence. A suitable combination of these mechanisms helps to explore the search space to avoid falling in local optima [ČLM13]. Indeed, many approaches emerged to tackle the problem of premature convergence in genetic algorithms by proposing new evolutionary algorithms or improving genetic operators with the aim of delaying a premature convergence. For example, the work [Mc +11] implemented *ACROMUSE*, a genetic algorithm that adapts crossover, mutation and parameter selection to preserve diversity in the population. Lozano et al. [LHC08] proposed replacement strategies that consider both fitness quality and the diversity of an individual in the population, in order to maintain individuals with high fitness and diversity for the next generation. Aslam et al. [AZN18] presented a selection operator that determines whether two individuals can be recombined considering their distance. On the other hand, other authors established a criterion that helps to select the individuals that will be combined: e.g. techniques based on neighborhoods such as niching methods [Mar+16] or approaches that consider behavior similarities by using fitness sharing [EN00; EN02]. A recent article showed how multi-objective optimization can be used to promote diversity in the population, considering both fitness and a diversity measure as objectives to be optimized [Seg+17].

In addition to the aforementioned problems of diversity and convergence, Genetic Programming (GP) has to deal with additional issues regarding the solution encoding [Koz92]. As the encodings used in GP have a non-linear structure, such as trees, it is harder to tackle the control of diversity [BGK04]. The problem of tree uncontrolled growth, known as the *bloating problem* in GP, leads to premature convergence [PM16]. Preventing this problem is an implicit goal for researchers

in GP, and different authors have proposed to modify genetic operators, fitness evaluation or selection schemes [Alf+08; Liu+07; JWP01] to solve the bloating problem while maintaining diversity in the population. Diversity measures may be classified into three main categories: *(a) behavioural or phenotypic diversity*, that considers differences in solution performance (fitness value) [KRK15; HB15; LCY16], *(b) syntactic or genotypic diversity*, which computes structural differences between individuals (shape and content of solutions in the population) [QHL15; Fer+17] and *(c)* a combination of both previous approaches [Aff+17; KHO19].

In this piece of research, we focus on improving diversity in formal grammar evolution by studying a combination of both phenotypic and genotypic approaches. As the solutions in GP are encoded using tree data structures traditionally, genotypic diversity measures focus on this type of representation [EN00; EN02; BP17; PA16; KS17; Bur+19]. We may classify the cited methods as distance measures or metrics: whereas a metric holds the properties of non-negativity, identity, symmetry, and triangle inequality, the remaining distance measures fail to accomplish one or more of these properties (usually the triangle inequality), but they can provide a value to estimate how distant two encoded solutions are, and have provided good results in the problems they have been used. Examples of (non-metric) distance measures are described in Burks et al. [BP17], which implement a density measure that considers a portion of each tree and determine how genetic material is distributed in the population; or the work [Bur+19], that uses isomorphic properties to measure structural diversity between two trees as the number of common nodes. Regarding metric proposals, Mateusz et al. [PA16] developed a metric that determines the differences between two individuals as the sequence of minimum cost of operations needed to transform one tree into another. Besides, Ekárt and Neméth described in [EN00; EN02] a metric that computes the structural difference of two encoded programs, distinguishing terminal and operator nodes.

Regarding phenotypic or behavioural diversity in genetic programming, the literature offers a wide variety of works that obtain semantic information from individuals during the evolutionary process and it is used it to improve the search space exploration in GP. These works range from classical methods such as the traditional *Ramped Half and Half* method to prevent the insertion of duplication trees into the population [Koz92] to more recent works such as [Cas+15]

that proposed the Geometric Semantic Genetic Programming (GGSP) [MKJ12] algorithm that designs an operator which measures semantic differences between two individuals to guide the search space exploration, or Nguyen et al. [Uy+10] whose developed a Semantic Similarity Crossover (SSC) which add semantic knowledge to control the changes of the semantic of individuals by comparing similarities of random subtrees. In summary, the main works proposed to directly or indirectly control diversity in Genetic Programming go from the structures used to represent the population to genetic operators and measures to control the population growth [Urs02].

In this piece of research, we focus on improving diversity in GP in two ways: (i) studying alternative structures to classical trees and (ii) developing measures to control diversity during the genetic procedure for these alternative structures. In previous works, we studied an alternative representation scheme to tree encoding, using Straight Line Programs (SLP) [Rue+19], and we concluded that using this representation may help to overcome limitations of classic tree encoding and to overcome local optima solutions. In this article, our main objective is to develop a metric based on edit distance that allows us to quantify how different two SLPs are, and use this metric to measure diversity in a population of SLPs to find a balance between diversity and convergence that helps to improve the exploration of the search space. More specifically, we combine the developed metric distance with the CHC algorithm [Esh91], to achieve a balance in the exploration and exploitation of the search space. Thus, the main novelty presented in this manuscript is the design of the similarity measure for Straight Line Programs, the proof that this measure is a metric, and its application in combination with a well tested evolutionary scheme such as CHC to prove its practical application. We remark that the classic edit distance is applied over sequences, and the similarity measure proposed in this work is adapted to grammars as formal languages. As no previous works have been proposed to quantify the distance between Straight Line Programs, we test our approach against tree-based encodings as baseline methods.

The remaining of the manuscript is structured as follows: Section 2 describes the background of our research introducing the fundamentals of Symbolic Regression, the representation problem and an outline of the classic CHC algorithm. Section 3 works out the proposed similarity measure. Section 4 applies the proposed metric in combination with the CHC algorithm to

control diversity and convergence in genetic programming. Section 5 shows the experimental results in synthetic data and real energy consumption data and discusses the comparative study of the proposal with state-of-the-art algorithms. Finally, Section 6 summarizes the conclusions obtained and describes future works.

2 Background

2.1 Symbolic regression and the representation problem

Regression analysis [Har15] is a statistical method that allows to find the relationships between dependent and independent variables. More specifically, regression analysis is composed by a model hypothesis $f(\bar{x}, \bar{w}) + \epsilon$, a set of input data $\bar{x} = \{x_1, x_2, ..., x_n\}$, a set of output data $\bar{y} = \{y_1, y_2, ..., y_m\}$, a set of constant parameters $\bar{w} = \{w_1, w_2, ..., w_k\}$, and an error ϵ that represents the part of the data that the model $f(\bar{x}, \bar{w})$ is unable to model. The main goal of regression analysis is to approximate the best values for the parameters $\bar{w}$ such that $\bar{y} \approx f(\bar{x}, \bar{w})$. With the aim of estimating the parameters, an error function is minimized such as $e(f, \bar{y}) = ||\bar{y} - f(\bar{x}, \bar{w})||$ as the sum of squared errors between the estimated functional model $f(\bar{x}, \bar{w})$ and the model hypothesis response $\bar{y}$.

The main limitation of regression analysis arises when the model hypothesis f is unknown and it is difficult to formulate manually. To solve this limitation, symbolic regression (SR) [BD02b] combines a set of primitive operators (such as $+, -, *, /$), independent variables $\bar{x}$ and parameters $\bar{w}$ to build an algebraic expression $\tilde{f}$ as an approximation to the optimal model f. Since symbolic regression is a NP-hard problem [LRW16b], Genetic Programming [Koz92] or Grammatical Evolution [OR01] algorithms have been traditionally used to explore the search space and to find the best approximation $\tilde{f}$ that minimizes an error measure with respect to the desired output data. GP is an evolutionary method based on biological evolution and simulates the evolutionary process. More specifically, GP builds a population of individuals which stochastically transforms into a new population in order to simulate the evolutionary cycle. During that cycle, it is

expected that the best individual survives and will contribute to a better population. Similarly, different mechanisms were studied to simulate the evolutionary cycle (crossover, mutation, etc) [Ang94] as well as the individual representation. Indeed, with regards to SR problems, tree structures have been highly used to encode algebraic expressions [MWB95] achieving promising results. As the representation problem determines the size of the search space, recent studies proved that alternative representations may reduce the search space, such as: linear genetic programming [BB01], that encodes programs as a sequence of instructions that operate over a memory equipped with a set of registers, instruction matrix [Li+08] that evolves tree nodes and subtrees separately, or linear strings of integers [MT00] that encode a graph as a list of node connections and functions. Also, this is the case of Straight Line Programs [APM08; Rue+19].

2.2 Straight Line Programs

A Straight Line Program (SLP) encodes a Straight Line Grammar (SLG) in Chomsky Normal Form [CN08]. A SLG is a context-free non-recursive grammar (V, T, P, S) able to generate a language with a single word, where V is the set of variables/non-terminal symbols of the grammar, T is the set of terminal symbols, P is the set of production rules and S is the starting non-terminal symbol of the grammar. Then, a SLP encodes a set of SLG production rules that can be used in SR to generate a single algebraic expression. In the SR problem addressed in this work, the set of terminal symbols is $T \equiv O_u \cup O_b \cup X \cup W$, where O_u is a set of unary operators, O_b is a set of binary operators, X is a set of terminal input data variables $\{x_1, x_2, ..., x_n\}$, and W is a set of constant parameters $\{w_1, w_2, ..., w_k\}$. A SLP contains N production rules $U_1, U_2, ..., U_N \in V$, where U_N is the starting symbol of the grammar and each production rule is of the form $U_i \rightarrow o_u r_{i1}$ or $U_i \rightarrow o_b r_{i1} r_{i2}$, where $o_u \in O_u$, $o_b \in O_b$ are operators, and $r_{i_1}, r_{i2} \in X \cup W \cup \{U_{i-1}, U_{i-2}, ..., U_1\}$ are the first and second operands, which can be a data terminal symbol or a non-terminal symbol that references subsequent production rules to avoid recursion.

In SR, a subset of elements in O_b can have the commutative property. In these cases, we decide

to establish an order for the construction of each production rule to reduce the search space. If a rule uses a commutative operator, then $r_{i_1} \prec r_{i_2}$ must be fulfilled, where the partial order relationship $\prec$ is defined as: $x_i \prec x_j \iff i < j$, $w_i \prec w_j \iff i < j$, $U_i \prec U_j \iff i < j$, $x_i \prec w_j \forall i, j$, and $w_i \prec U_j \forall i, j$. Equation 2.29 shows a sample SLP (A) and its representation (A') considering the partial order relationship constraint. As it can be seen, such constraint reduces the search space without shrinking the space of possible solutions to a SR problem. While the SLP A generates the algebraic expression $A = (\frac{w_3}{w_2} - w_1) * x_1 + \frac{w_3}{w_2} - w_1$, the SLP A' derives the equivalent expression $A' = \frac{w_3}{w_2} - w_1 + x_1 * (\frac{w_3}{w_2} - w_1)$.

$$A = \begin{cases} U_0 \ \rightarrow \ / \ w_3 \ w_2 \\ U_1 \ \rightarrow \ - \ U_0 \ w_1 \\ U_2 \ \rightarrow \ * \ U_1 \ x_1 \\ U_3 \ \rightarrow \ + \ U_2 \ U_1 \end{cases} \iff A' = \begin{cases} U_0 \ \rightarrow \ / \ w_3 \ w_2 \\ U_1 \ \rightarrow \ - \ U_0 \ w_1 \\ U_2 \ \rightarrow \ * \ x_1 \ U_1 \\ U_3 \ \rightarrow \ + \ U_1 \ U_2 \end{cases} \tag{2.29}$$

2.3 CHC Algorithm

The CHC algorithm (Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation) is an evolutionary algorithm proposed by Eshelman [Esh91] for binary encoding, and it was designed to hold a balance between diversity and convergence in the population. This algorithm was adapted for real-coded chromosomes in [ES93] and [CDS06]. Unlike a classical genetic algorithm, CHC does not use a mutation operator, and introduces four components to achieve the aforementioned balance:

- An elitist selection. The N best individuals of the current and generated offspring populations are selected to compose the new population in the next generation, where N stands for the population size.
- The HUX (original binary encoding proposal [Esh91]) or BLX-α (extended real-coded CHC [ES93]) crossover operators, to avoid premature convergence caused by recombination.

- An incest prevention mechanism to avoid crossover of similar solutions. If a distance measure over two parent solutions is over a threshold τ, then the parent crossover is allowed. In the original binary CHC proposal, the Hamming distance was used to compare parent chromosomes, and the value τ was calculated initially as $L/4$ (L is the size of the chromosomes). On the other hand, the real-coded CHC used the Euclidean distance to measure the similarity between two parents, and τ was initialized to $0{,}1 * d_{max}$ (d_{max} is the maximum distance between two elements in the population).

- A restart procedure to reinitialize the population if it has converged. If two populations in consecutive algorithm iterations contain the same solutions, then τ is decreased. When $\tau \leq 0$, the population is reinitialized with random solutions and a copy of the best solution found during the evolutionary process.

In this article, we use a variant of the real-coded CHC algorithm in Section 4 adapted to the SLP encoding scheme. We use the metric proposed in Section 3 as a diversity measure between SLPs.

3 Similarity measure for Straight Line Programs

Our goal is to define a similarity measure that provides the structural difference between SLPs and can be computed efficiently. We are inspired by the edit distance metric [RY98]. Hence, the proposed similarity measure provides the minimum number of operations required to transform one SLP into another. As in edit distance, the available operations to compute such transformation are insertions, deletions and substitutions. Then, the more similar two SLPs are, the lower the proposed distance value should be and in contrast, the more dissimilar two SLPs are, the greater value the measure should provide.

Given two SLPs A and B coming from two SLGs $G_A = (V, T, P_A, A_s)$ and $G_B = (V, T, P_B, B_s)$, we define the similarity measure between both SLPs as $d(A, B) = d(A_s, B_s)$, i.e. the similarity measure between the starting symbols of each SLP. The definition of $d(A_s, B_s)$ is provided as a

recursive formula with general and base cases.

Base case. Let $t_1, t_2 \in T \cup \{\epsilon\}$ be two terminal symbols of the grammar (operators, data variables, constant parameters, or the empty word). Then we define $d(t_1, t_2)$ as shown in Equation 2.30.

$$d(t_1, t_2) = \begin{cases} 1, & t_1 \neq t_2 \\ 0, & t_1 = t_2 \end{cases} \tag{2.30}$$

In the base case, $d(t_1, t_2) = 1$ means a substitution of t_1 with t_2, $d(t_1, \epsilon)$ means deletion of t_1, and $d(\epsilon, t_2)$ means insertion of t_2.

General case. For the general case, we use the following notation: $o_i^A, o_j^B \in O_u \cup O_b$ are the operators used in the i-th and j-th rules of SLPs A and B, respectively; $r_{i1}^A, r_{i2}^A, r_{j1}^B, r_{j2}^B \in V \cup T \cup \{\epsilon\}$ are the first and second operands of the i-th and j-th rules of SLPs A and B. We focus on rules of the form $U_i^A \rightarrow o_i^A r_{i1}^A r_{i2}^A$ and $U_j^B \rightarrow o_j^B r_{j1}^B r_{j2}^B$ without loss of generality. In case o_i^A (respectively o_j^B) is an unary operator, then it is assumed that $r_{i2}^A = \epsilon$ (respectively $r_{j2}^B = \epsilon$). We also distinguish a set $C \in O_b$ as the subset of binary operators that meet the commutative property. With this in mind, Equation 2.31 describes how to compute the distance $d(U_i^A, U_j^B)$ between the aforementioned rules.

$$d(U_i^A, U_j^B) = \begin{cases} d(o_i^A, o_j^B) + d(r_{i1}^A, r_{j1}^B) + d(r_{i2}^A, r_{j2}^B) \text{ if } o_i^A, o_j^B \in \{O_u \cup O_b\} \backslash C \\ d(o_i^A, o_j^B) + min\{d(r_{i1}^A, r_{j1}^B) + d(r_{i2}^A, r_{j2}^B), \\ \quad d(r_{i1}^A, r_{j2}^B) + d(r_{i2}^A, r_{j1}^B)\} \text{if } o_i^A \in C \vee o_j^B \in C \end{cases} \tag{2.31}$$

In Equation 2.31, we may observe that the similarity measure between rules U_i^A and U_j^B computes the minimum number of insertion, deletion and substitution operations to transform U_i^A into U_j^B, considering special cases when the commutative property allows to exchange the order of the rule operands.

A final consideration must be taken into account when computing the distance between operands, as for instance $d(r_{i1}^A, r_{j1}^B)$. We have defined the similarity measure for terminal symbols in Equation 2.30 and for non-terminal symbols in Equation 2.31. As two arbitrary rule operands (named as $u_1, u_2 \in V \cup T$) being compared can be terminal or non terminal indistinguishably, we define $d(u_1, u_2)$ for these cases as Equation 2.32 shows.

$$d(u_1,u_2) = \begin{cases} d(o_1,\epsilon) + d(r_{11},u_2) + d(r_{12},\epsilon) \text{ if } u_1 \in V, u_2 \in T, o_1 \in \{O_u \cup O_b\} \backslash C \\ d(o_1,\epsilon) + min\{d(r_{11},u_2) + d(r_{12},\epsilon), \\ \quad d(r_{11},\epsilon) + d(r_{12},u_2)\} \text{ if } u_1 \in V, u_2 \in T, o_1 \in C \\ d(\epsilon,o_2) + d(u_1,r_{21}) + d(\epsilon,r_{22}) \text{ if } u_1 \in T, u_2 \in V, o_2 \in \{O_u \cup O_b\} \backslash C \\ d(\epsilon,o_2) + min\{d(u_1,r_{21}) + d(\epsilon,r_{22}), \\ \quad d(\epsilon,r_{21}) + d(u_1,r_{22})\} \text{ if } u_1 \in T, u_2 \in V, o_2 \in C \end{cases} \tag{2.32}$$

The first two cases in Equation 2.32 assume that $u_1 \rightarrow o_1 r_{11} r_{12}$ is a non-terminal symbol and $u_2 \in T$, and distinguishes if the rule operator o_1 meets the commutative property or not and, in contrast, the latter two cases are met when $u_2 \rightarrow o_2 r_{21} r_{22}$ and $u_1 \in T$, respectively.

An efficient algorithm with complexity $O(N * M)$ can be designed using Dynamic Programming [RY98] to compute the distance between A and B, where N and M stand for the number of rules of A and B, respectively. As an example, we show the calculation of the proposed measure using two sample SLPs A and B, with initial symbols A_2 and B_1, respectively (see Equation 2.33).

$$A = \begin{cases} A_0 \rightarrow + \; x_1 \; x_1 \\ A_1 \rightarrow / \; A_0 \; A_0 \\ A_2 \rightarrow * \; A_1 \; w_1 \end{cases} \qquad B = \begin{cases} B_0 \rightarrow / \; w_1 \; x_1 \\ B_1 \rightarrow - \; B_0 \; w_1 \end{cases} \tag{2.33}$$

Table 2.1 shows the arrangement of rules A_i of SLP A and B_j of SLP B in columns and rows, respectively. Each cell contains the value of the distance $d(A_i, B_j)$. The table must be filled from top to the bottom and from left to right. As an example, the target value $d(A_2, B_1)$ is computed as follows:

$$d(A_2, B_1) = d(*, -) + \text{mín}\{d(A_1, B_0) + d(w_1, w_1), d(A_1, w_1) + d(w_1, B_0)\} = 6 \tag{2.34}$$

	ϵ	x_1	w_1	A_0	A_1	A_2
ϵ	0	1	1	3	7	9
x_1	1	0	1	2	6	8
w_1	1	1	0	3	7	8
B_0	3	3	2	2	5	7
B_1	5	5	4	5	6	6

Tabla 2.1: Example of calculation of d(A,B)

The proposed measure $d(A, B)$ is a metric. To prove such statement, we follow the same reasoning that was used in [WSB76] for the edit distance, although specified for SLPs. For this reason, we must first provide some prior definitions.

Definition I ***(τ space).*** Let $Q : V \cup T \cup \{\epsilon\} \to V \cup T \cup \{\epsilon\}$ be a transformation of a grammar symbol into another, plus the empty word. We define $\tau = \{Q\}$, i.e. the set of all possible transformations of symbols, including identity transformation I. As $V \cup T \cup \{\epsilon\}$ is finite, then τ is finite and every transformation in τ can be numbered as $\tau = \{Q_1, Q_2, Q_3, ...\}$, and contains all possible insertions, substitutions and deletions over the grammar symbols. Each transformation Q_i has an associated weight $w(Q_i)$. In our case, all weights $w(Q_i) = 1$ except for the identity, $w(I) = 0$, according to equation 2.30.

Definition II ***(Transformation sequence).*** Suppose a SLP A and its j-th rule $U_j \to o_j r_{j1} r_{j2}$ and a transformation $Q \in \tau$. We define $Q^{j,0}(A) = U_j \to Q(o_j) r_{j1} r_{j2}$, $Q^{j,1} = U_j \to o_j Q(r_{j1}) r_{j2}$, $Q^{j,2} = U_j \to o_j r_{j1} Q(r_{j2})$, i.e. $Q^{j,i}$ is the use of transformation Q at the i-th symbol of the consequent of the j-th rule.

We define a transformation sequence over a SLP A as $\bar{Q}(A) = (Q_{k_l}^{j_l,i_l} \circ Q_{k_{l-1}}^{j_{l-1},i_{l-1}} \circ ... \circ Q_{k_1}^{j_1,i_1})(A) = Q_{k_l}^{j_l,i_l}(Q_{k_{l-1}}^{j_{l-1},i_{l-1}}(...(Q_{k_1}^{j_1,i_1}(A))...))$. Each transformation sequence $\bar{Q}$ has also a weight, that is calculated as $w(\bar{Q}) = \sum_{p=1}^{l} w(Q_{k_p}^{j_p,i_p})$.

Finally, we define $\{A \to B\}_\tau = \{\bar{Q}(A) : \bar{Q}(A) = B\}$, i.e. the set of sequences of transformations in τ that transform SLP A into SLP B.

Definition III ***(Equivalence relation =).*** Let SLP be the set of all possible SLPs, and $A, B \in SLP$ two SLPs with starting symbols U_N^A, U_M^B respectively, and rules $A = \{U_i^A \to o_i^A r_{i1}^A r_{i2}^A\}$ and $B = \{U_j^B \to o_j^B r_{j1}^B r_{j2}^B\}$, where $o_i^A, o_j^B \in O_u \cup O_b; r_{i1}^A, r_{i2}^A, r_{j1}^B, r_{j2}^B \in V \cup T \cup \{\epsilon\}$. We write the subset of binary operators with commutative property as $C \subseteq O_b$. We define the equivalence relation $= (A, B)$, which we write as $A = B$, as follows:

$A = B \Leftrightarrow U_N^A$ and U_M^B generates the same word or $\exists i, j : 1 \leq i \leq N, 1 \leq j \leq M \wedge o_i^A, o_j^B \in C$ such as the change of operands in the rules $\{U_i^A \to o_i^A r_{i2}^A r_{i1}^A\}$ and/or $\{U_j^B \to o_j^B r_{j2}^B r_{j1}^B\}$ make U_N^A and U_M^B generate the same word. We remark that the existence of rules i and/or j does not have to be unique.

It is easy to verify that $A = A \forall A \in SLP$ (reflexivity), $A = B \Leftrightarrow B = A \forall A, B \in SLP$ (symmetry) and, if $A = B$ and $B = C$ then $A = C, A, B, C \in SLP$ (transitivity), and therefore $=$ is an equivalence relation.

The defined equivalence relation states that two SLPs that provide two algebraic expressions for a SR problem are considered equivalent if both expressions are the same even if their syntax is different due to the effect of commutative operators. Also, $=$ partitions the space into a set of equivalence classes $SLP/=$, where all SLPs that belong to the same class are equivalent under $=$.

Theorem: Let SLP be the set of all possible SLPs, and $A, B \in SLP$. The proposed similarity measure $d(A, B)$ is a metric over the quotient space $SLP/=$.

Proof: If A, B, C are three different SLPs, and $d(A, B)$ is a metric, then the following conditions must be met [GGS87]:

$$d(A,B) \geq 0, \text{ (non negativity)}$$
$$d(A,B) = 0 \iff A = B, \text{ (identity)}$$
$$d(A,B) = dist(B,A) \text{ (symmetry)},$$
$$d(A,C) \leq d(A,B) + d(B,C) \text{ (triangle inequality)}$$

Proof of conditions of non negativity and identity are trivial from Equations 2.30 and 2.31, since $d(A,B)$ is not allowed to have negative values, and two SLPs have $d(A,B) = 0$ only if A and B belong to the same equivalence class. Condition of symmetry is also derived directly from the symmetric property of the equivalence relation $=$ in Definition III.

Regarding the triangle inequality condition, let us rewrite that the distance between the SLPs A and B is computed as the weight of the transformation sequence with minimum number of transformations in τ as $d(A,B) = \text{mín}_{\{A\to B\}_\tau} \sum_{p=1}^{l_1} w(Q_{k_p}^{j_p i_p})$.

Then, $d(B,A) = \text{mín}_{\{B\to A\}_\tau} \sum_{p=1}^{l_2} w(Q_{k_p}^{j_p i_p})$. As every transformation Q in set τ has an inverse Q^{-1}, (a deletion for an insertion and viceversa, and an inverse transformation for a substitution), if $d(A,B)$ is minimum then $d(A,B) = d(B,A) = \text{mín}_{\{A\to B\}_\tau} \sum_{p=1}^{l_1} w(Q_{k_p}^{j_p i_p}) =$ $\text{mín}_{\{B\to A\}_\tau} \sum_{p=1}^{l_1} w((Q_{k_p}^{j_p i_p})^{-1})$. Also, the distance from A and B to a third SLP C can be written as $d(A,C) = \text{mín}_{\{A\to C\}_\tau} \sum_{p=1}^{l_3} w(Q_{k_p}^{j_p i_p})$ and $d(B,C) = \text{mín}_{\{B\to C\}_\tau} \sum_{p=1}^{l_4} w(Q_{k_p}^{j_p i_p})$.

As every transformation weight is unitary, except for the identity for which $w(I) = 0$, then:

$$\min_{\{A\to B\}_\tau} \sum_{p=1}^{l_1} w(Q_{k_p}^{j_p i_p}) \quad + \quad \min_{\{B\to C\}_\tau} \sum_{p=1}^{l_4} w(Q_{k_p}^{j_p i_p}) \quad \geq \quad \min_{\{A\to C\}_\tau} \sum_{p=1}^{l_3} w(Q_{k_p}^{j_p i_p}) \quad (2.35)$$

Meaning that the number of steps with unitary weight to transform A into B and then B into C must be greater or equals than the number of steps with unitary weight to transform A into C directly, and then

$$d(A,B) + d(B,C) \geq d(A,C)$$

The opposite condition cannot hold, since all weights are unitary and negative weights are not allowed in the defined distance, so that
$\text{mín}_{\{A\to C\}_\tau} \sum_{p=1}^{l_3} w(Q_{k_p}^{j_p i_p}) < d(A,B) + d(B,C)$ is not possible according to Equations 2.30 and 2.31. This concludes with the proof.

4 Control of diversity of Straight Line Programs evolution with CHC

In this section we describe an application of the proposed metric to control diversity in Genetic Programming using SLPs as representation for symbolic regression problems. More specifically, we use the proposed distance as a diversity measure in an adapted CHC evolutionary algorithm as incest prevention mechanism. We selected the CHC algorithm since it is a classic approach that combines a balance in diversity and convergence and it has been widely tested in the literature. We name our approach as SLP-CHC and it is based on the real-coded CHC adaptation [ES93]. As we evolve SLPs instead of real-coded chromosomes, we use the crossover proposed in [APM08].

The adaptation of classic CHC implementation to our proposal is shown in Algorithm 4. The procedure starts by initializing a population $P(t)$ of N random SLPs at iteration $t = 0$, then each individual is evaluated by using the Mean Square Error (MSE) as fitness measure between the real output data y and the computed $\tilde{y}$ (see Equation 2.37). The procedure $averageDistance$ calculates the distance threshold Th as the average distance between all individuals of the population, as it is shown in Equation 2.36, where d is the proposed SLP distance, N is the population size and c_i, c_j are the i-th and j-th SLPs in the population. After initialization, the main algorithm repeats until a stopping criterion is fulfilled. In this work, the stopping criterion used is to reach a number of solutions evaluated. Each algorithm iteration encompasses the following steps: *elitist selection*, *SLP recombination*, *solution evaluation* and the *divergence procedure*. Firstly, we build our population of parents by copying all individuals of the current

population in random order. After that, the *SLP recombination* operator [APM08; Rue+19] is applied to generate two new offspring if the SLP distance between the candidate parents exceeds the threshold Th. Once the recombination operator is used, each offspring is evaluated according to the fitness measure, and the new population for the next iteration $P(t+1)$ is built with the N best SLPs between parents and offspring.

Before next iteration starts, it is checked if $P(t+1) = P(t)$. If so, then the threshold Th is decreased by $rate$. The value of $rate$ is defined as $rate = d_{max} * T$ (where d_{max} is the maximum distance between two individuals of the population during initialization, and T is a value in the range $(0, 1)$), and controls the convergence speed according to the diversity of the whole population. Afterwards, if the difference threshold Th is less than 0, the *diverge* procedure is triggered: The new population is composed by $(N-1)$ random SLPs and the best SLP found in the previous generation. Then the difference threshold Th is recalculated by using Equation 2.36.

$$Th = \frac{1}{N(N-1)} \sum_{1 \leq i < N} \sum_{i < j \leq N} d(c_i, c_j) \tag{2.36}$$

$$MSE = \frac{1}{N} \sum_{i=1}^{n} (\tilde{y} - y)^2 \tag{2.37}$$

5 Experimentation

The main goal of this experimentation is to test if the proposed metric together with the CHC algorithm can improve the exploration and exploitation of the Symbolic Regression solution space in Genetic Programming. As no previous works have been devised to measure similarities between SLPs, we compare the approach with classic metrics for tree representation [EN00; EN02] and Genetic Programming evolution of SLPs with no diversity control. In particular, we want to compare our proposal with methods used in symbolic regression problems that were specifically designed to increase the diversity of the population using genotypic diversity measures, efficiently computed. More specifically, we used as baseline methods the proposals of Ekart et al. [EN00;

Algorithm 4 SLP-CHC algorithm

Require: N, the number of individuals of the population
Require: $T \in (0, 1)$
Require: $\bar{x} = \{(x_1, x_2, ..., x_n)\}$ input data for SLP evaluation
Require: $\bar{w} = (w_1, w_2, ..., w_k)$ a set of constant parameters
Require: $\bar{y} = \{y_1\}$ output data for SLP evaluation
Ensure: SLP(1...N) a sequence of rules that encode the algebraic expression of the best individual

```
{Initialization of population}
Initialize P(t)
Evaluate(P(t), w̄, x̄, ȳ)
Set t = 1
Set d=averageDistance
{SLP-CHC procedure}
while No stopping criterion is fulfilled do
  t = t + 1
  select C(t) from P(t-1)
  C'(t)= SLP recombination(C(t))
  Evaluate(C'(t), w̄, x̄, ȳ)
  P(t) = elitist selection(P(t-1), C'(t))
  if P(t) = P(t − 1) then
    Th = Th − rate
    if Th < 0 then
      P(t)=diverge
      Evaluate(P(t), w̄, x̄, ȳ)
      Th=averageDistance
    end if
  end if
end while
return Best solution of P(t)
```

EN02]. These approaches compute syntactical differences of the population, represented with tree structures, using a metric based on the edit distance and they demonstrated to be able to increase the diversity, achieving equivalent solutions than classical Genetic Programming algorithms. Due to the approaches of Ekart et al. were able to achieve as robust solution as classical GP algorithms as well as to increase diversity, we consider these approaches as baseline methods of GP.

In order to clarify the comparison carried out in this section, we name each approach as follows: **SLP-GA** for Genetic Programming using SLP representation [Rue+19]; **SDM** for Genetic Programming approach using fitness sharing and tree representation [EN00]; **DDM** for Genetic Programming that also used fitness sharing and adaptive maintenance of diversity [EN02]; and

SLP-CHC to refer the method proposed in Section 4. Finally, we performed two experiments to test if our proposal has potential over the mentioned baseline methods: the first one is carried out over a set of synthetic data in subsection 5.1 with the aim of validating each approach in a controlled environment. After that, in the second experiment (section 5.2), we deal with a real world problem about energy consumption modelling.

5.1 Experimentation with Synthetic Data

5.1.1 Data acquisition and experimental settings

We use 19 benchmark algebraic expressions (see Equations 2.38 to 2.57) that are widely used in the literature [Nic+15]. For each algebraic expression, we generated 500 random data in the domain [0.0,1.0] for each input variable. Finally, each dataset was randomly divided into train (70 % of data) and test (remaining 30 %). We used the training data to evolve each algorithm to find the best solution and the test data were used to validate each approach. Therefore, results in Subsection 5.1.2 focus on the test set.

The available operators for the symbolic regression datasets are $\{+, -, *, /, \sin, \cos, \log, \text{mín}, \text{máx}\}$ in all cases, and the set of constant parameters was set to $\bar{w} = (1, 2, 3)$. We performed a preliminary trial-and-error procedure to find the optimal parameters for each algorithm and the best results were provided with the following parameters: we allowed a set of 31 rules for SLP (**SLP-GA** and **SLP-CHC**) and 31 nodes for trees (**SDM** and **DDM**). The population size was set to 100 individuals and the stopping criterion was having 20000 solutions evaluated. Then, for **SLP-GA** we set the crossover and mutation probabilities as 90 % and 10 % respectively; for both **SDM** and **DDM** we tuned the niche size ($\sigma = 0{,}5$), $K = 1$ and the crossover and mutation probabilities to to 70 % and 30 % respectively. With regards to the **SLP-CHC**, the value T used to compute the decrease rate was tuned to 0.3. The fitness measure used was the mean square error (MSE), to be minimized. Finally, we performed 30 executions for each algorithm and problem with different random seeds to carry

out a statistical test that helped us to determine if there exist significant differences between the results obtained.

$$f_1(x_1, x_2) = \frac{e^{-(x_1-1)^2}}{1{,}2 + (x_2 - 2{,}5)^2} \tag{2.38}$$

$$f_2(x_1, x_2) = e^{-x_1} * x_1^3 * \cos(x_1) * \sin(x_1) * (\cos(x_1) * \sin^2(x_1) - 1) * (x_2 - 5) \tag{2.39}$$

$$f_3(x_1, x_2, x_3) = 30 * \frac{(x_1 - 1) * (x_3 - 1)}{x_2^2 * (x_1 - 10)} \tag{2.40}$$

$$f_4(x_1, x_2) = 6 * \sin(x_1) * \cos(x_2) \tag{2.41}$$

$$f_5(x_1, x_2) = (x_1 - 1) * (x_2 - 3) + 2 * \sin((x_1 - 4) * (x_2 - 4)) \tag{2.42}$$

$$f_6(x_1, x_2) = \frac{(x_1 - 3)^4 + (x_2 - 3)^3 - (x_2 - 3)}{(x_2 - 2)^4 + 10} \tag{2.43}$$

$$f_7(x_1, x_2) = \frac{1}{1 + x_1^{-4}} + \frac{1}{1 + x_2^{-4}} \tag{2.44}$$

$$f_8(x_1, x_2) = x_1^4 - x_1^3 + \frac{x_2^2}{2} - x_2 \tag{2.45}$$

$$f_9(x_1, x_2) = \frac{8}{2 + x_1^2 + x_2^2} \tag{2.46}$$

$$f_{10}(x_1, x_2) = \frac{x_1^3}{5} + \frac{x_2^3}{2} - x_2 - x_1 \tag{2.47}$$

$$f_{11}(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) = x_1 * x_2 \tag{2.48}$$

$$+x_3 * x_4 + x_5 * x_6 + x_1 * x_7 * x_9 + x_3 * x_6 * x_{10} \tag{2.49}$$

$$f_{12}(x_1, x_2, x_3, x_4, x_5) = -5{,}41 + 4{,}9 * \frac{x_4 - x_1 + \frac{x_2}{x_5}}{3 * x_4} \tag{2.50}$$

$$f_{13}(x_1, x_2, x_3, x_4, x_5, x_6) = \frac{(x_5 * x_6)}{\frac{x_1}{x_2} * \frac{x_3}{x_4}} \tag{2.51}$$

$$f_{14}(x_1, x_2, x_3, x_4, x_5) = 0{,}81 + 24{,}3 * \frac{2x_2 + 3x_3^2}{4x_4^3 + 5x_5^4} \tag{2.52}$$

$$f_{15}(x_1, x_2, x_3, x_4, x_5) = 32 - 3 * \frac{\tan(x_1)}{\tan(x_2)} * \frac{\tan(x_3)}{\tan(x_4)} \tag{2.53}$$

$$f_{16}(x_1, x_2, x_3, x_4, x_5) = 22 - 4{,}2 * (\cos(x_1) - \tan(x_2)) * (\frac{\tanh(x_3)}{\sin(x_4)}) \tag{2.54}$$

$$f_{17}(x_1, x_2, x_3, x_4, x_5) = x_1 * x_2 * x_3 * x_4 * x_5 \tag{2.55}$$

$$f_{18}(x_1, x_2, x_3, x_4, x_5) = 12 - 6 * \frac{\tan(x_1)}{e^{x_2}} * (x_3 - \tan(x_4)) \tag{2.56}$$

$$f_{19}(x_1, x_2, x_3, x_4, x_5) = 2 - 2{,}1 * \cos(9{,}8 * x_1) * \sin(1{,}3 * x_5) \tag{2.57}$$

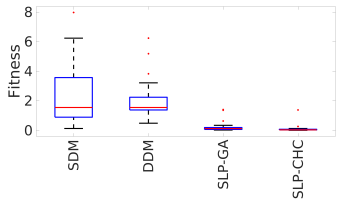
5.1.2 Results and discussion

The results obtained in the test datasets for each approach are shown in Table 2.2. Each row is associated with the results of its corresponding benchmark dataset (Column 1). Then, for each algorithm we show the median MSE and the average execution time measured in seconds. We remark that we use the median value instead of the mean since the results do not follow a normal distribution and, in these cases, the mean cannot be considered an appropriate statistic summarization value. We have also included boxplots with the MSE distributions in the test sets to give support to our analysis in Figure 2.1, and they include information about the best and worst solutions.

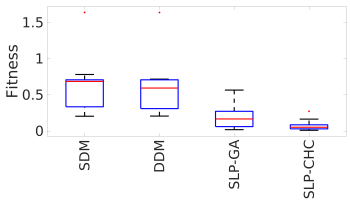
	SDM		**DDM**		**SLP-GA**		**SLP-CHC**	
	Test	**Time**	**Test**	**Time**	**Test**	**Time**	**Test**	**Time**
f_1	1.55 (3)	2.9	1.53 (3)	2.5	0.12 (2)	2.43	$4{,}14 \times 10^{-2}$ (1)	4.2
f_2	0.68 (3)	2.73	0.59 (3)	2.26	0.16 (2)	2.93	$5{,}33 \times 10^{-2}$ (1)	5.46
f_3	$7{,}93 \times 10^{6}$ (2)	3.03	$7{,}94 \times 10^{6}$ (2)	2.4	$4{,}35 \times 10^{6}$ (1)	2.36	$4{,}59 \times 10^{6}$ (1)	4.16
f_4	1.47 (3)	2.9	1.46 (3)	2.6	0.21 (2)	2.83	$6{,}27 \times 10^{-2}$ (1)	5.3
f_5	8.01 (3)	2.96	7.42 (3)	2.93	1.9 (2)	2.7	0.89 (1)	4.6
f_6	2.25 (3)	3	2.25 (3)	3	0.18 (2)	2.5	$8{,}22 \times 10^{-2}$ (1)	4.9
f_7	$1{,}44 \times 10^{-2}$ (3)	3	$1{,}51 \times 10^{-2}$ (3)	2.9	$1{,}39 \times 10^{-2}$ (2)	2.73	$1{,}33 \times 10^{-2}$ (1)	4.93
f_8	$2{,}49 \times 10^{-2}$ (3)	2.96	$2{,}49 \times 10^{-2}$ (3)	2.93	$6{,}92 \times 10^{-3}$ (2)	4	$1{,}8 \times 10^{-3}$ (1)	5.33
f_9	0.24 (3)	2.33	0.25 (3)	2.36	$8{,}21 \times 10^{-2}$ (2)	2.93	$1{,}99 \times 10^{-2}$ (1)	5.6
f_{10}	$3{,}45 \times 10^{-2}$ (3)	2.76	$3{,}45 \times 10^{-2}$ (3)	2.73	$1{,}47 \times 10^{-2}$ (2)	3.2	$3{,}99 \times 10^{-2}$ (1)	4.96
f_{11}	0.23 (3)	3	0.22 (3)	2.83	0.15 (2)	3	0.14 (1)	9.33
f_{12}	$3{,}99 \times 10^{4}$ (2)	2.96	$3{,}99 \times 10^{4}$ (2)	2.96	$1{,}113 \times 10^{4}$ (1)	2.5	$2{,}15 \times 10^{4}$ (1)	5.96
f_{13}	$4{,}91 \times 10^{2}$ (2)	3.03	$4{,}89 \times 10^{2}$ (2)	3	$1{,}25 \times 10^{2}$ (1)	2.73	$1{,}19 \times 10^{2}$ (1)	6.33
f_{14}	$6{,}99 \times 10^{7}$ (3)	2.96	$6{,}99 \times 10^{7}$ (3)	2.96	$1{,}65 \times 10^{7}$ (1)	2.63	$2{,}6 \times 10^{7}$ (2)	5.1
f_{15}	$3{,}77 \times 10^{4}$ (3)	2.76	$3{,}8 \times 10^{4}$ (3)	2.86	$2{,}28 \times 10^{4}$ (2)	2.43	$8{,}53 \times 10^{3}$ (1)	5.43
f_{16}	$8{,}81 \times 10^{2}$ (1)	3	$8{,}82 \times 10^{2}$ (1)	2.96	$9{,}42 \times 10^{2}$ (1)	2.53	$8{,}84 \times 10^{2}$ (1)	5.16
f_{17}	$4{,}76 \times 10^{-3}$ (2)	2.2	$4{,}76 \times 10^{-3}$ (2)	2.73	$2{,}57 \times 10^{-3}$ (1)	2.63	$1{,}79 \times 10^{-3}$ (1)	5.6
f_{18}	0.6 (3)	2.9	0.55 (3)	2.86	1.42 (2)	2.53	1.26 (1)	4.63
f_{19}	0.86 (3)	2.96	0.86 (3)	2.7	0.81 (2)	2.9	0.73 (1)	4.66

Tabla 2.2: Results of **SDM**, **DDM**, **SLP-GA** and **SLP-CHC** in benchmark algebraic expressions

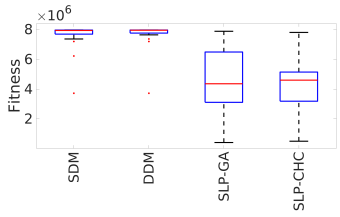
The Shapiro-Wilk test has been applied to check if the results obtained for each approach follow normality conditions. As the fitness distribution results did not follow a normal distribution, we performed a non-parametric Kruskal-Wallis test (KW) with a 95 % of confidence level to validate if there are significant differences between each approach statistically. The results of the KW test are presented together with the median fitness result of each approach in Columns



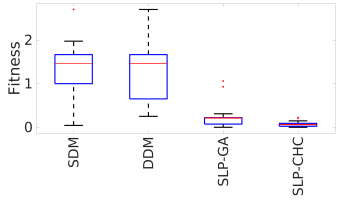
(a) Benchmark algebraic expression f_1



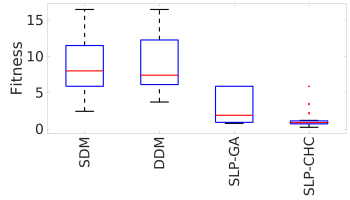
(b) Benchmark algebraic expression f_2



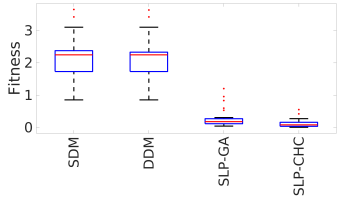
(c) Benchmark algebraic expression f_3



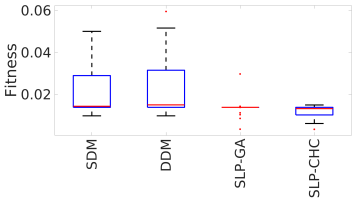
(d) Benchmark algebraic expression f_4



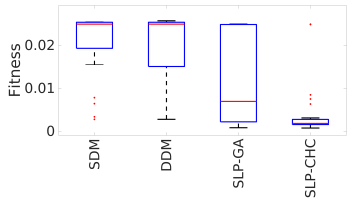
(e) Benchmark algebraic expression f_5



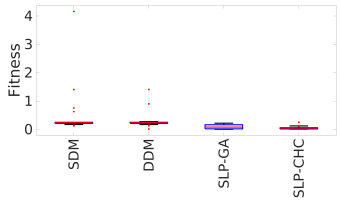
(f) Benchmark algebraic expression f_6



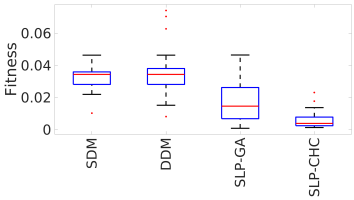
(g) Benchmark algebraic expression f_7



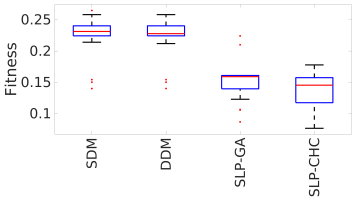
(h) Benchmark algebraic expression f_8



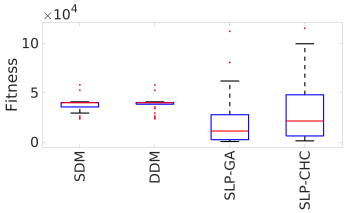
(i) Benchmark algebraic expression f_9



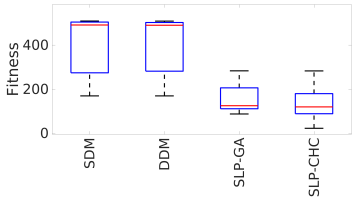
(j) Benchmark algebraic expression f_{10}



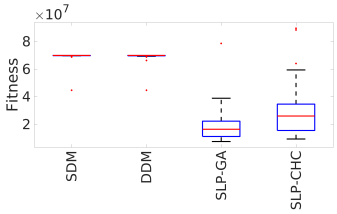
(k) Benchmark algebraic expression f_{11}



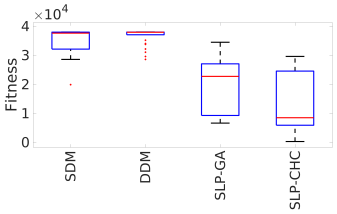
(l) Benchmark algebraic expression f_{12}



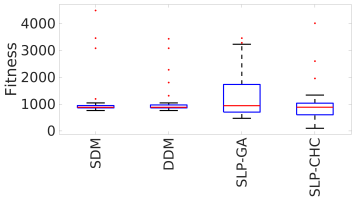
(m) Benchmark algebraic expression f_{13}



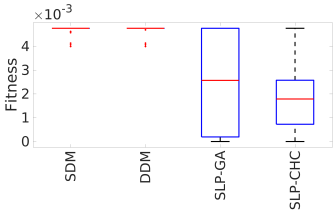
(n) Benchmark algebraic expression f_{14}



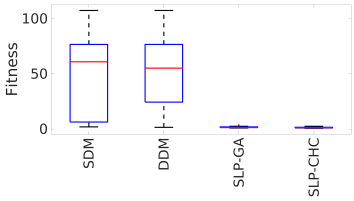
(ñ) Benchmark algebraic expression f_{15}



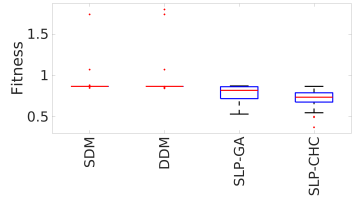
(o) Benchmark algebraic expression f_{16}



(p) Benchmark algebraic expression f_{17}



(q) Benchmark algebraic expression f_{18}



(r) Benchmark algebraic expression f_{19}

Figura 2.1: Boxplots of accuracy for each benchmark algebraic and approach

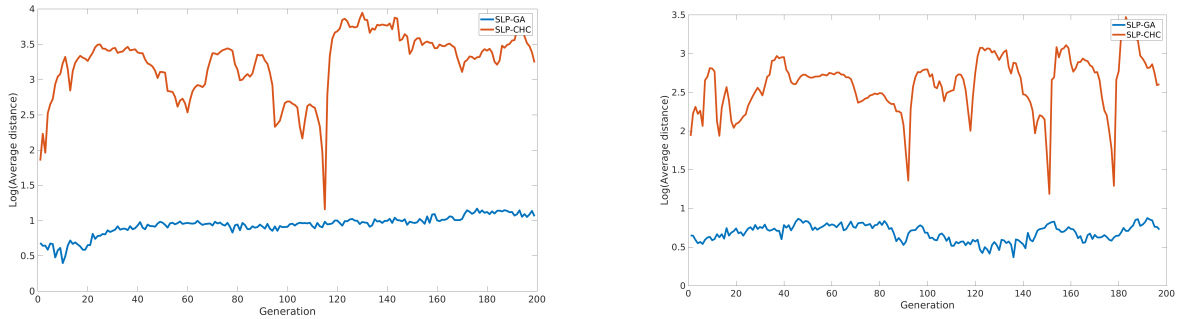


Figura 2.2: Diversity of SLP-GA (blue line) and SLP-CHC (red line) calculated as the average distance measure of the population

2, 4, 6 and 8 in Table 2.2, in brackets. More specifically, we performed a ranking between each approach and benchmark algebraic expression from 1 (the best approach) to 4 (worst algorithm results). If there are no significant differences between the two algorithms in a dataset, they are ranked with the same number.

In a first analysis of the results, we start with a comparison of the selected baseline methods (**SDM** and **DDM**). Results of the applied statistical test suggest that there are no differences regarding fitness in both approaches. This is consistent with the work of Ékart and Németh [EN02], where they argued that diversity is increased using their method, but they did not find improvement in the fitness of solutions.

In a previous work [Rue+19] we discussed how the SLP representation may overcome traditional limitations of tree representation. In the experimentation performed, this is validated and **SLP-GA** improves **SDM** and **DDM** in 18 of 19 problems studied, according to Table 2.2 and Figure 2.1. The remaining of the analysis focuses on the comparison of **SLP-CHC** and **SLP-GA**, consequently.

We continue our analysis by comparing the results of **SLP-GA** and **SLP-CHC** with the aim of verifying our initial hypothesis that the proposed metric together with the CHC algorithm helps to improve the control of diversity and convergence and also overcomes local optima. **SLP-CHC** has provided better results than **SLP-GA** in 13 cases, **SLP-CHC** and **SLP-GA** were equivalent in 5 cases, and **SLP-GA** was the best algorithm in 1 case. This confirms that the proposed metric can be used as a diversity measure in Genetic Programming to find a balance between exploration and exploitation.

Regarding the best solutions found by **SLP-GA** and **SLP-CHC**, Figure 2.1 shows that **SLP-CHC** provided the lowest fitness in 14 cases and **SLP-GA** achieved the lowest fitness in 5 experiments. With regards to the worst fitness value, **SLP-GA** provided the worst solution in 13 cases, meanwhile **SLP-CHC** did it in 4 experiments. In the remaining 2 cases, **SLP-GA** and **SLP-CHC** obtained the same worst solution.

Regarding the algorithm robustness, Figure 2.1 also shows that **SLP-CHC** is more robust than

SLP-GA, since the distance between the intermediate quartiles in the boxplots are lower in 14 cases for **SLP-CHC**, meaning that it is expected that a random execution of **SLP-CHC** will provide better results than **SLP-GA**.

The discussion follows with a diversity study of the results provided for both **SLP-GA** and **SLP-CHC**. We selected the executions that provided the best SLP for each approach, with the aim of analyzing the diversity behavior during the evolutionary process. Figure 2.2 describes a sample for the problems f_{14}, f_{16}, where the axis X stands for the generation of each approach and the axis Y for the average diversity in log scale, calculated as described in Equation 2.36. Blue lines represent the diversity measured for **SLP-GA** and red lines stand for the diversity of **SLP-CHC**.

We may see that the **SLP-GA** approach was able to preserve diversity during the evolutionary cycle, since the average distance was not decreasing during all generations. However, it was unable to explore the solution space enough to overcome local optima. Nevertheless, the **SLP-CHC** approach increased diversity in the population substantially, leading to a better exploration of the search space. In the cases when the population diversity decreased, the diverge procedure allowed to explore unknown areas since different individuals were included in the population. These facts suggest us that an increase of population diversity may help to reduce the premature convergence, allowing **SLP-CHC** to overcome the results of **SLP-GA**.

We conclude with this section with an analysis of the execution time. We can see in Table 2.2 that **SLP-CHC** was computationally more expensive than the remaining baseline approaches. This fact is a consequence of both representation schemes used to encode individuals and the similarity measure used in each approach. Whereas the metric used in **SDM** and **DDM** does not take into account commutative operations and only compare two trees node by node, our proposed similarity measure used in **SLP-GA** and **SLP-CHC** takes into account the grammar that generates an algebraic expression, as well as commutative property of operators. Thus, an increase in the execution time is expected, since for each pair of parents it is required to calculate the distance before crossover is applied. However, the increase in computational time observed could be palliated with the benefits of applying diversity control using the proposed

metric, depending on the researcher needs to avoid local optima.

5.2 Experimentation with Real Data

5.2.1 Data Acquisition and experimental settings

With the approaches validated in a controlled environment, this section describes the experimentation with real data. The data used in this experimentation were provided from a set of energy consumption time series of different buildings coming from the University of Granada from March 2013 to October 2015, hourly measured. More specifically, the buildings are two research centres and two faculties, which we named as B_1, B_2, B_3 and B_4 for confidentiality reasons. The energy consumption of these buildings are shown in Figure 2.3, where the axis X stands for the time and the axis Y for the energy consumption in kW/h. Before using the data, it must be preprocessed because could be missing data due to sensor failures or light cuts. The preprocessing step encompasses an interpolation of the missing values (around a 5 % of the data) and a time alignment to obtain the data in the same temporal range. Finally, we aggregate the energy (kW) consumed each 24 hours of the same day to get a final dataset. Our initial hypothesis in this experimentation is that the energy consumption of a weekday can be modelled as a combination of the energy consumption of the remaining working days in the same week. Consequently, we carried out a correlation analysis of the energy consumption of each weekday and building with the aim of understanding the energy consumption behavior. To that end, Figures 2.4a to 2.4d show the correlation plot matrices for each working day and building. In each correlation plot, the diagonal shows the histogram, which provides us information about the energy consumption distribution for each working day. Moreover, the text red in the remaining scatter plots gathers the correlation coefficient R, ranging from -1 to 1, between the days of the corresponding row and column of the plot matrices. The mentioned correlation coefficient denotes whether exists a positive or negative correlation between two variables. Values near to 1 denotes high positive correlation (respectively to -1 with high negative correlation), meanwhile

values closer to 0 mean low correlation. In this way, the mentioned figures verify that there is a high ($R > 0{,}7$) or medium ($0{,}3 \leq R \leq 0{,}7$) positive correlation between the energy consumption of the working days.

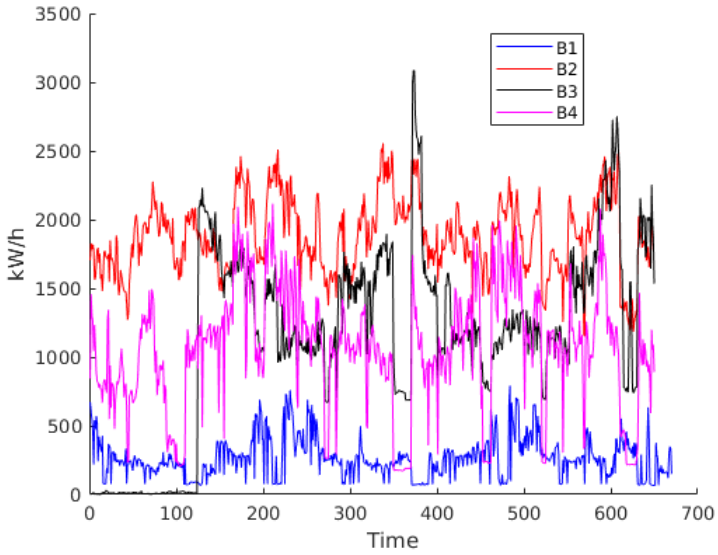
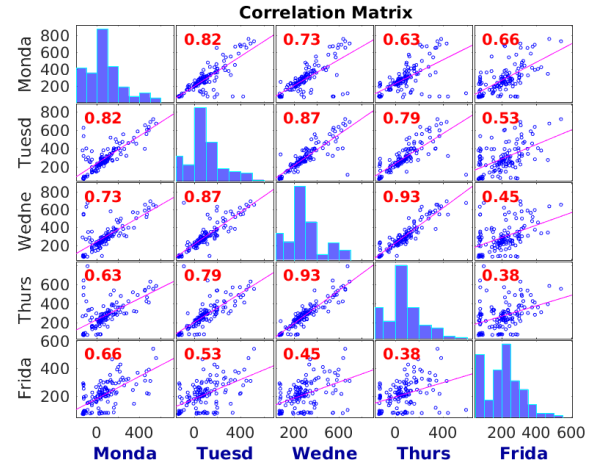
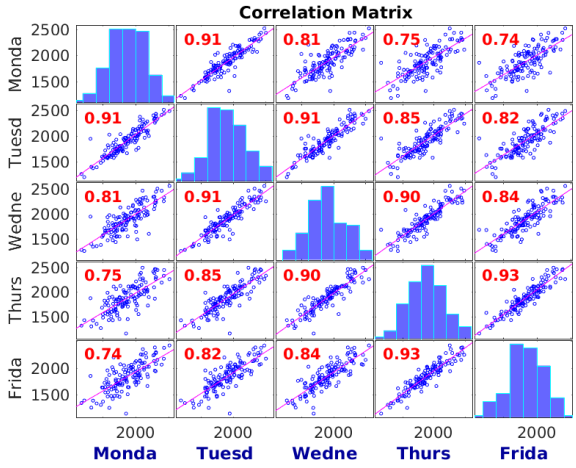


Figura 2.3: Energy consumption data series of buildings B_1, B_2, B_3 and B_4

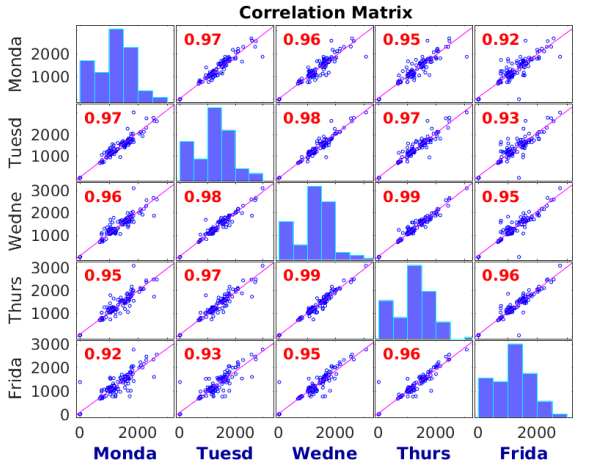
In the experiments, we divided each dataset into train (70 %) and test (30 %) data to avoid over-fitting. The train set was used to build each model of each approach and then it was tested over the test set to validate the quality of the solutions found. Moreover, as well as in synthetic data experiments, we performed 30 executions for each approach and problem with different seed to perform statistical tests. Although we kept the same configuration for symbolic regression (parameters $\bar{w}$ and mathematical operators), we performed a preliminary trial-and-error procedure to find the optimal parameter for each approach, and the best results were obtained with the following parameters: we allow a set of 31 rules for SLP and 31 nodes for trees. The population size was set to 200 individuals and the stopping criterion was 20000 solutions evaluated. Then, for *SLP-GA* we tuned the crossover and mutation probabilities to 90 % and 10 % respectively. For **SDM** and **DDM** we set the crossover and mutation probabilities to 80 % and 20 % respectively, and the niche size (σ) to 0.05. Regarding the **SLP-CHC** approach, the value used to compute the decrease rate was set to 0,3.



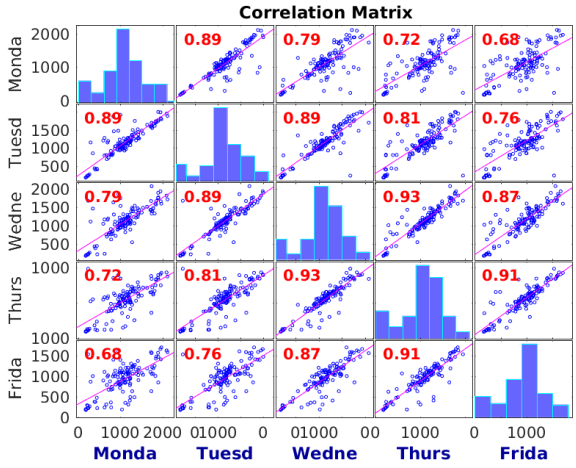
(a) Energy consumption correlation between working days for building *B1*



(b) Energy consumption correlation between working days for building *B2*



(c) Energy consumption correlation between working days for building *B3*



(d) Energy consumption correlation between working days for building *B4*

Figura 2.4: Correlation matrices of energy consumption for buildings B_1 to B_4, from Monday to Friday.

5.2.2 Results and Discussion

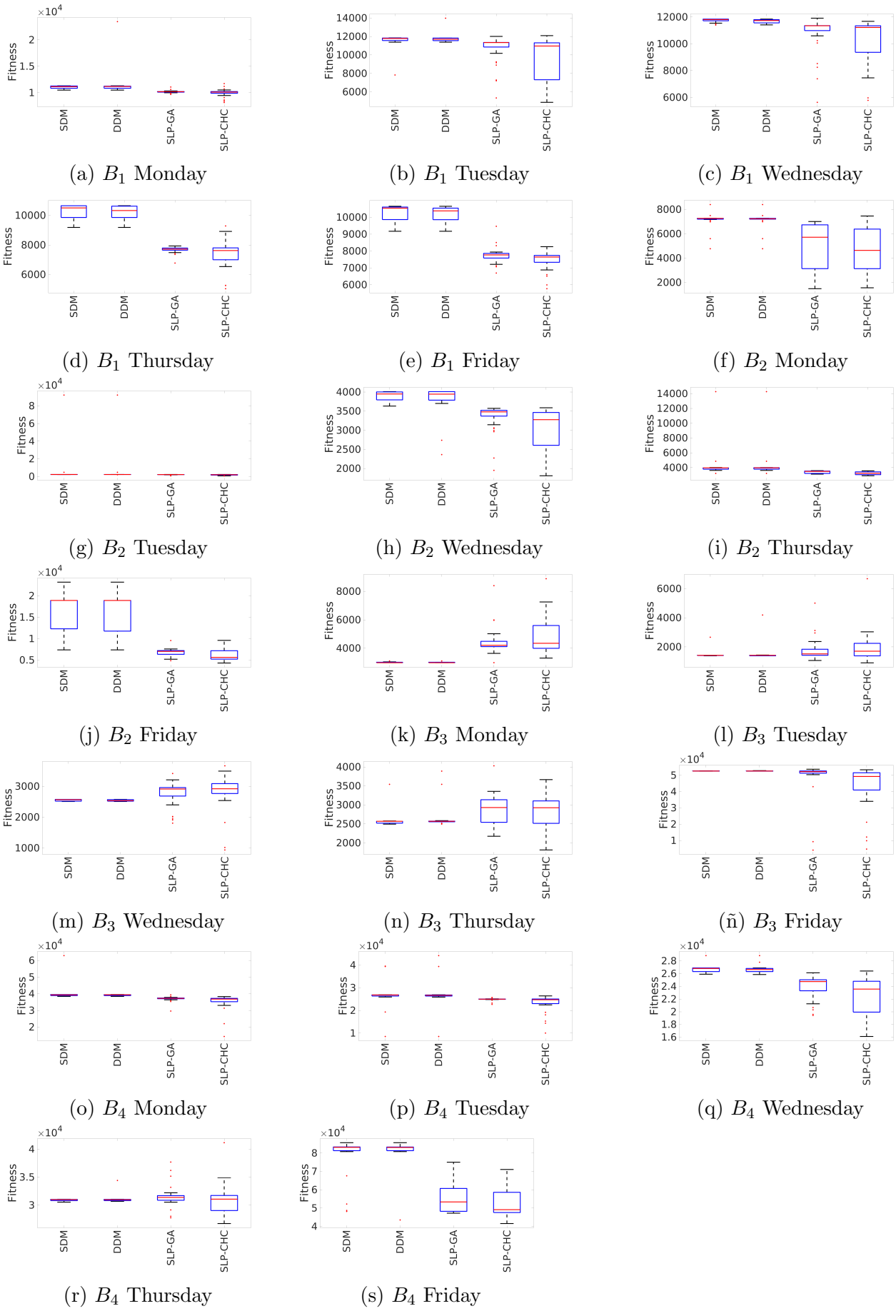
Table 2.3 shows the results obtained for each approach and problem in test data. The columns are organized in groups of two, for each algorithm analyzed (**SDM**, **DDM**, **SLP-GA** and **SLP-CHC**). For each approach, we focus on the measurements *Test* which contains the median MSE of 30 runnings and *Time* that gathers the average time spent by the algorithm in the 30 runnings, measured in milliseconds. Then, the rows are organized in groups of 5, where each group references the working day of each building (B_1 to B_4). On the other hand, to provide a

better analysis of the results, we included boxplots of the MSE distribution in all experiments in Figure 2.5. Each figure contains the boxplots of the MSE for the algorithms being compared, i.e. **SDM**, **DDM**, **SLP-GA** and **SLP-CHC**, for the same building and working day. Besides, we carried out a statistical test to empirically validate if one approach has potential over them. We first performed a normality test (Shapiro-Wilk test) to verify if the results provided for each approach follow a normal distribution. As the results did not follow a normal distribution, we performed a non-parametric Kruskal-Wallis test (KW) with a 95 % confidence level. The results of the KW test were presented together with the median results of each approach in brackets (Columns 2, 4, 6 and 8) as a sorted list that comes from 1 to 4, where algorithms marked to 1 mean that were better than the algorithms marked with 2, 3 and 4 respectively. If two approaches were marked with the same number meant that no significant differences were found between each approach.

Similarly to the experimentation in benchmark data and the results of the work [EN02], the baseline methods (**SDM** and **DDM**) did not present significant differences in the results over real energy consumption data in terms of accuracy. Moreover, regarding the results of Table 2.3 and the boxplots of Figure 2.5, SLP approaches overcame the results of **SDM** and **DDM** in 15 of 20 problems.

Therefore, we focus the analysis of this experimentation on the results of **SLP-GA** and **SLP-CHC**. Regarding the results of the median values of **SLP-GA** and **SLP-CHC** of Table 2.3 and the results of the second quartile of the boxplots in Figure 2.5, we may verify that **SLP-CHC** provided better results in 17 of 20 problems while **SLP-GA** did it in the remaining 3 experiments. With regards to the KW results, **SLP-CHC** proved to be significantly better in 10 of 20 experiments, meanwhile in the remaining 10 experiments no significant differences were found. Regarding the execution time, similar conclusion to Section 5.1 may be achieved. Although the proposed similarity measure together with the CHC adaptation helped to avoid local optima and performed better results, it caused an increase of the computational time of **SLP-CHC**, being more computational expensive than the remaining approaches.

On the other hand, we have also carried out an analysis of the diversity measured during the



(a) B_1 Monday (b) B_1 Tuesday (c) B_1 Wednesday

(d) B_1 Thursday (e) B_1 Friday (f) B_2 Monday

(g) B_2 Tuesday (h) B_2 Wednesday (i) B_2 Thursday

(j) B_2 Friday (k) B_3 Monday (l) B_3 Tuesday

(m) B_3 Wednesday (n) B_3 Thursday (ñ) B_3 Friday

(o) B_4 Monday (p) B_4 Tuesday (q) B_4 Wednesday

(r) B_4 Thursday (s) B_4 Friday

Figura 2.5: Boxplots of accuracy for SDM, DDM, SLP-GA and SLP-CHC for each building and working day

	SDM		DDM		SLP-GA		SLP-CHC	
	Test	**Time**	**Test**	**Time**	**Test**	**Time**	**Test**	**Time**
	Building B_1							
Monday	$1{,}11 \times 10^4$ (2)	$1{,}63 \times 10^3$	$1{,}12 \times 10^4$ (2)	$1{,}33 \times 10^3$	$1{,}02 \times 10^4$ (1)	$6{,}4 \times 10^2$	$1{,}01 \times 10^4$ (1)	$2{,}86 \times 10^3$
Tuesday	$1{,}18 \times 10^4$ (3)	$1{,}33 \times 10^3$	$1{,}17 \times 10^4$ (3)	$1{,}07 \times 10^3$	$1{,}13 \times 10^4$ (2)	$6{,}32 \times 10^2$	$1{,}1 \times 10^4$ (1)	$2{,}99 \times 10^3$
Wednesday	$1{,}18 \times 10^4$ (3)	$1{,}37 \times 10^3$	$1{,}17 \times 10^4$ (3)	$1{,}07 \times 10^3$	$1{,}13 \times 10^4$ (2)	$6{,}43 \times 10^2$	$1{,}12 \times 10^4$ (1)	$2{,}69 \times 10^3$
Thursday	$1{,}05 \times 10^4$ (2)	$1{,}43 \times 10^3$	$1{,}03 \times 10^4$ (2)	$1{,}03 \times 10^3$	$7{,}72 \times 10^3$ (1)	$6{,}51 \times 10^2$	$7{,}62 \times 10^3$ (1)	$2{,}85 \times 10^3$
Friday	$1{,}05 \times 10^4$ (3)	$1{,}2 \times 10^3$	$1{,}04 \times 10^4$ (3)	$1{,}1 \times 10^3$	$7{,}77 \times 10^3$ (2)	$6{,}59 \times 10^2$	$7{,}64 \times 10^3$ (1)	$2{,}75 \times 10^3$
	Building B_2							
Monday	$7{,}26 \times 10^3$ (2)	$1{,}53 \times 10^3$	$7{,}23 \times 10^3$ (2)	$1{,}4 \times 10^3$	$5{,}71 \times 10^3$ (1)	$6{,}34 \times 10^2$	$4{,}63 \times 10^3$ (1)	$3{,}19 \times 10^3$
Tuesday	$2{,}33 \times 10^3$ (2)	$1{,}4 \times 10^3$	$2{,}28 \times 10^3$ (2)	$1{,}33 \times 10^3$	$2{,}21 \times 10^3$ (1)	$6{,}81 \times 10^2$	$2{,}02 \times 10^3$ (1)	$2{,}68 \times 10^3$
Wednesday	$3{,}95 \times 10^3$ (3)	$1{,}33 \times 10^3$	$3{,}94 \times 10^3$ (3)	$1{,}27 \times 10^3$	$3{,}48 \times 10^3$ (2)	$6{,}46 \times 10^2$	$3{,}28 \times 10^3$ (1)	$2{,}85 \times 10^3$
Thursday	$3{,}97 \times 10^3$ (3)	$1{,}47 \times 10^3$	$3{,}97 \times 10^3$ (3)	$1{,}2 \times 10^3$	$3{,}49 \times 10^3$ (2)	$7{,}1 \times 10^2$	$3{,}22 \times 10^3$ (1)	$2{,}63 \times 10^3$
Friday	$1{,}89 \times 10^4$ (3)	$1{,}77 \times 10^3$	$1{,}89 \times 10^4$ (3)	$1{,}33 \times 10^3$	$7{,}04 \times 10^3$ (2)	$6{,}86 \times 10^2$	$5{,}63 \times 10^3$ (1)	$3{,}03 \times 10^3$
	Building B_3							
Monday	$2{,}98 \times 10^3$ (1)	$1{,}33 \times 10^3$	$2{,}98 \times 10^3$ (1)	$1{,}17 \times 10^3$	$4{,}2 \times 10^3$ (2)	$6{,}45 \times 10^2$	$4{,}34 \times 10^3$ (2)	$2{,}62 \times 10^3$
Tuesday	$1{,}42 \times 10^3$ (1)	$1{,}43 \times 10^3$	$1{,}42 \times 10^3$ (1)	$1{,}3 \times 10^3$	$1{,}52 \times 10^3$ (2)	$7{,}67 \times 10^2$	$1{,}7 \times 10^3$ (2)	$3{,}2 \times 10^3$
Wednesday	$2{,}56 \times 10^3$ (1)	$1{,}4 \times 10^3$	$2{,}53 \times 10^3$ (1)	$1{,}17 \times 10^3$	$2{,}91 \times 10^3$ (2)	$6{,}4 \times 10^2$	$2{,}92 \times 10^3$ (2)	$2{,}69 \times 10^3$
Thursday	$2{,}57 \times 10^3$ (1)	$1{,}23 \times 10^3$	$2{,}57 \times 10^3$ (1)	$1{,}13 \times 10^3$	$2{,}93 \times 10^3$ (2)	$6{,}47 \times 10^2$	$2{,}92 \times 10^3$ (2)	$2{,}51 \times 10^3$
Friday	$5{,}26 \times 10^4$ (3)	$1{,}47 \times 10^3$	$5{,}26 \times 10^4$ (3)	$1{,}03 \times 10^3$	$5{,}22 \times 10^4$ (2)	$7{,}05 \times 10^2$	$4{,}93 \times 10^3$ (1)	$2{,}93 \times 10^3$
	Building B_4							
Monday	$3{,}93 \times 10^4$ (3)	$1{,}47 \times 10^3$	$3{,}94 \times 10^4$ (3)	$1{,}33 \times 10^3$	$3{,}73 \times 10^4$ (2)	$6{,}4 \times 10^2$	$3{,}68 \times 10^4$ (1)	$2{,}91 \times 10^3$
Tuesday	$2{,}68 \times 10^4$ (3)	$1{,}43 \times 10^3$	$2{,}66 \times 10^4$ (3)	$1{,}3 \times 10^3$	$2{,}49 \times 10^4$ (2)	$6{,}17 \times 10^2$	$2{,}47 \times 10^4$ (1)	$2{,}79 \times 10^3$
Wednesday	$2{,}68 \times 10^4$ (3)	$1{,}43 \times 10^3$	$2{,}66 \times 10^4$ (3)	$1{,}37 \times 10^3$	$2{,}47 \times 10^4$ (2)	$6{,}62 \times 10^2$	$2{,}36 \times 10^4$ (1)	$2{,}86 \times 10^3$
Thursday	$3{,}1 \times 10^4$ (1)	$1{,}37 \times 10^3$	$3{,}1 \times 10^4$ (1)	$1{,}3 \times 10^3$	$3{,}14 \times 10^4$ (1)	$6{,}68 \times 10^2$	$3{,}11 \times 10^4$ (1)	$2{,}52 \times 10^3$
Friday	$8{,}31 \times 10^4$ (2)	$1{,}6 \times 10^3$	$8{,}3 \times 10^4$ (2)	$1{,}43 \times 10^3$	$5{,}33 \times 10^4$ (1)	$7{,}14 \times 10^2$	$4{,}91 \times 10^4$ (1)	$3{,}01 \times 10^3$

Tabla 2.3: Results of **SDM**, **DDM**, **SLP-GA** and **SLP-CHC** in real energy consumption data

genetic procedure of both **SLP-GA** and **SLP-CHC** approaches. Figure 2.6 shows the diversity registered (measured using Equation 2.36) during the training procedure for both **SLP-GA** and **SLP-CHC** in two scenarios. As can be seen, **SLP-CHC** helped to increase the diversity of the population and, consequently to achieve a better exploration of the search space. The results of Table 2.3 together with the diversity analysis suggest that **SLP-CHC** has potential over **SLP-GA**.

To conclude with the analysis of this experimentation, Figure 2.7 shows the original datasets and the results of the modelled data for each building. These results help us to conclude that **SLP-CHC** is a promising alternative for real applications of symbolic regression because although the results of Table 2.3 suggest a high MSE value, the plots verified that the modelled data fits correctly the real data.

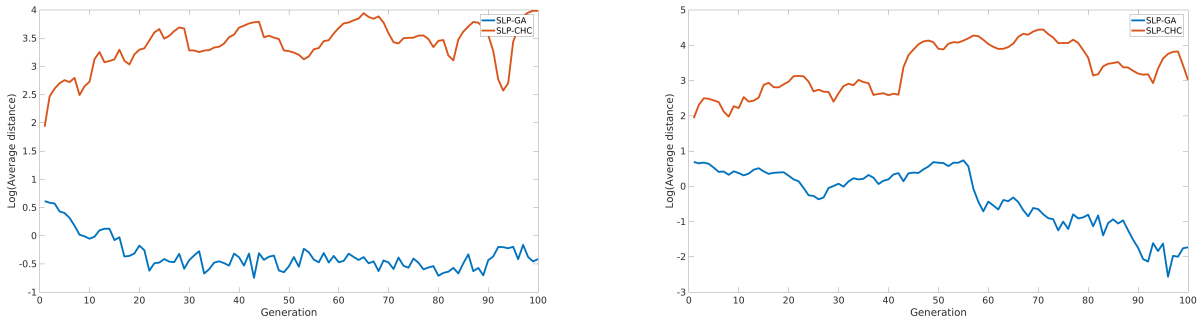
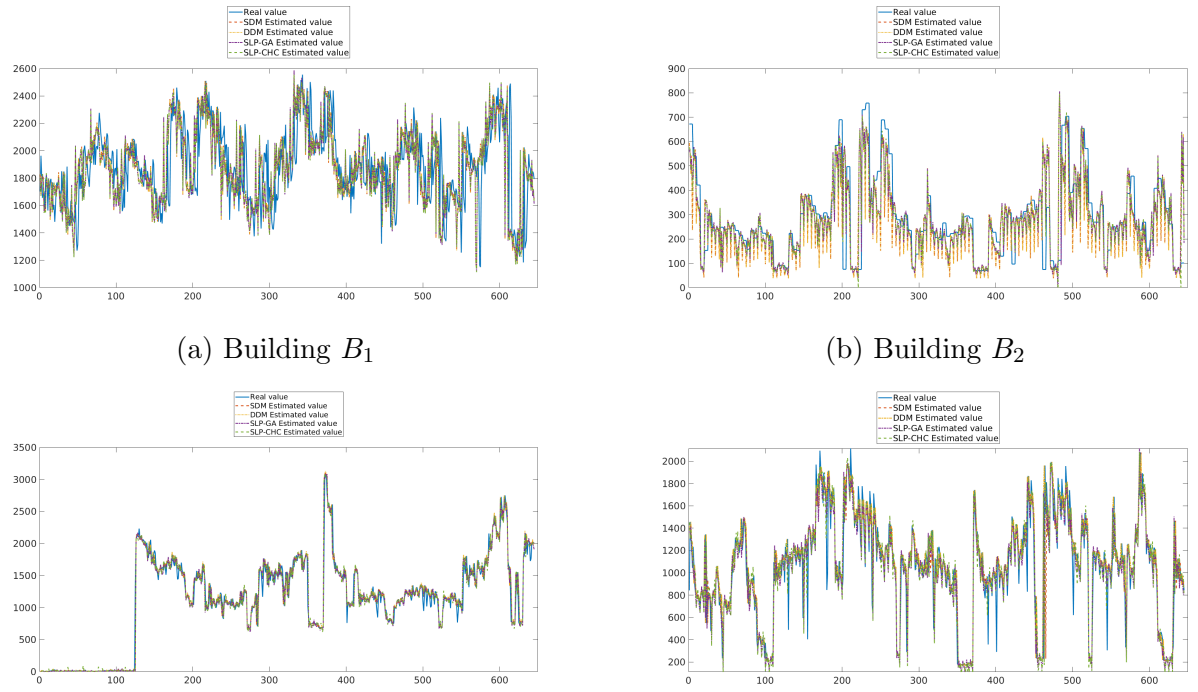


Figura 2.6: Diversity of SLP-GA (blue line) and SLP-CHC (red line) in energy consumption data on Monday of building B_2 (left) and on Thursday of building B_3 (right). The diversity was calculated as the average distance measure of the population.



(a) Building B_1

(b) Building B_2

(c) Building B_3

(d) Building B_4

Figura 2.7: Plots of real data (blue), SDM estimated data (red), DDM estimated data (yellow), SLP-GA estimated data (violet) and SLP-CHC estimated data (green) for the buildings B_1 to B_4

6 Conclusions

In this manuscript, we have developed a similarity measure to compare Straight Line Programs to solve symbolic regression problems, and we have proven that such measure is a metric. As an application of the proposed metric, we have faced the problem of controlling diversity and convergence in genetic programming using SLPs as representation model.

To do so, we adapted the classic CHC algorithm and included the proposed metric into the CHC's incest prevention mechanism to measure diversity in a population of SLPs. The remaining components of CHC allow us to obtain a balance in diversity and convergence, and the results obtained show that the inclusion of the developed diversity measure helps to overcome local optima with a small increase of the computational cost.

As no previous works define a similarity measure for SLPs, we have compared our approach with other tree-based representation genetic programming proposals existing in the literature. We validated our contribution in a set of 19 benchmark algebraic expressions and in a real problem about energy consumption modelling at the University of Granada, concluding that the proposal of this manuscript outperforms the results obtained by the baseline approaches.

The proposed metric considers the commutative property of algebraic operators to compute the similarity of SLPs, so that we can categorize our approach in the range between pure syntax-based similarity measures and semantic ones. Future works will be conducted to extend this metric considering semantic similarities between SLPs, and also to generalize the metric to further problems beyond symbolic regression.

Acknowledgements

This work was supported by the project TIN201564776-C3-1-R.

References

[Aff+17] Michael Affenzeller y col. «Dynamic Observation of Genotypic and Phenotypic Diversity for Different Symbolic Regression GP Variants». En: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '17. ACM, 2017, págs. 1553-1558. DOI: 10.1145/3067695.3082530.

[Alf+08] E. Alfaro-Cid y col. «Prune and Plant: A New Bloat Control Method for Genetic Programming». En: *2008 Eighth International Conference on Hybrid Intelligent Systems*. 2008, págs. 31-35. DOI: 10.1109/HIS.2008.127.

[Ang94] Peter John Angeline. «Genetic Programming and Emergent Intelligence». En: *Advances in Genetic Programming*. MIT Press, 1994, págs. 75-98. DOI: 10.5555/185984.185992.

[APM08] C. L. Alonso, J. Puente y J. L. Montaña. «Straight Line Programs: A New Linear Genetic Programming Approach». En: *2008 20th IEEE International Conference on Tools with Artificial Intelligence*. Vol. 2. 2008, págs. 517-524.

[AZN18] Muhammad Waqar Aslam, Zhechen Zhu y Asoke Kumar Nandi. «Diverse partner selection with brood recombination in genetic programming». En: *Applied Soft Computing* 67 (2018), págs. 558-566. DOI: https://doi.org/10.1016/j.asoc.2018.03.035.

[BB01] Markus Brameier y Wolfgang Banzhaf. «Evolving Teams of Predictors with Linear Genetic Programming». En: *Genetic Programming and Evolvable Machines* 2.4 (2001), págs. 381-407. DOI: 10.1023/A:1012978805372.

[BD02b] Lynne Billard y Edwin Diday. «Symbolic Regression Analysis». En: *Classification, Clustering, and Data Analysis*. Berlin, Heidelberg, 2002, págs. 281-288. DOI: 10.1007/978-3-642-56181-8_31.

[BGK04] E. K. Burke, S. Gustafson y G. Kendall. «Diversity in genetic programming: an analysis of measures and correlation with fitness». En: *IEEE Transactions on Evolutionary Computation* 8.1 (2004), págs. 47-62. DOI: 10.1109/TEVC.2003.819263.

[BK13a] Florian Benz y Timo Kötzing. «An Effective Heuristic for the Smallest Grammar Problem». En: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. GECCO '13. Amsterdam, The Netherlands, 2013, págs. 487-494. DOI: 10.1145/2463372.2463441.

[BP17] Armand R. Burks y William F. Punch. «An analysis of the genetic marker diversity algorithm for genetic programming». En: *Genetic Programming and Evolvable Machines* 18.2 (2017), págs. 213-245. DOI: 10.1007/s10710-016-9281-9.

[BR07] K.M.S. Badran y P.I. Rockett. «The roles of diversity preservation and mutation in preventing population collapse in multiobjective genetic programming». En: *Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference*. 2007, págs. 1551-1558. DOI: 10.1145/1276958.1277272.

[Bur+19] Bogdan Burlacu y col. «Parsimony Measures in Multi-objective Genetic Programming for Symbolic Regression». En: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '19. Prague, Czech Republic: ACM, 2019, págs. 338-339. DOI: 10.1145/3319619.3322087.

[Cas+15] Mauro Castelli y col. «Prediction of energy performance of residential buildings: A genetic programming approach». En: *Energy and Buildings* 102 (2015), págs. 67-74.

[CDS06] O. Cordón, S. Damas y J. Santamaría. «Feature-based image registration by means of the CHC evolutionary algorithm». En: *Image and Vision Computing* 24.5 (2006), págs. 525-533. DOI: https://doi.org/10.1016/j.imavis.2006.02.002.

[ČLM13] Matej Črepinšek, Shih-Hsi Liu y Marjan Mernik. «Exploration and Exploitation in Evolutionary Algorithms: A Survey». En: *ACM Comput. Surv.* 45.3 (2013), págs. 1-33. ISSN: 0360-0300. DOI: 10.1145/2480741.2480752.

[CLY09] G. Chen, C. P. Low y Z. Yang. «Preserving and Exploiting Genetic Diversity in Evolutionary Programming Algorithms». En: *IEEE Transactions on Evolutionary Computation* 13.3 (2009), págs. 661-673. DOI: 10.1109/TEVC.2008.2011742.

[CN08] Francisco Claude y Gonzalo Navarro. *Indexing Straight-Line Programs*. 2008.

[EN00] Anikó Ekárt y S. Z. Németh. «A Metric for Genetic Programs and Fitness Sharing». En: *Genetic Programming*. Berlin, Heidelberg, 2000, págs. 259-270. DOI: 10.1007/978-3-540-46239-2_19.

[EN02] Anikó Ekárt y Sandor Z. Németh. «Maintaining the Diversity of Genetic Programs». En: *Genetic Programming*. Berlin, Heidelberg, 2002, págs. 162-171. DOI: 10.13140/RG.2.1.2876.0167.

[ES93] Larry J. Eshelman y J. David Schaffer. «Real-Coded Genetic Algorithms and Interval-Schemata». En: *Foundations of Genetic Algorithms*. Vol. 2. Foundations of Genetic Algorithms. Elsevier, 1993, págs. 187-202. DOI: https://doi.org/10.1016/B978-0-08-094832-4.50018-0.

[Esh91] Larry J. Eshelman. «The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination». En: *Foundations of Genetic Algorithms* 1 (1991), págs. 265-283. DOI: https://doi.org/10.1016/B978-0-08-050684-5.50020-3.

[Fer+17] Adel Ferdjoukh y col. «Measuring Differences to Compare sets of Models and Improve Diversity in MDE». En: oct. de 2017.

[GGS87] J.R. Giles, J.R. Giles y Australian Mathematical Society. *Introduction to the Analysis of Metric Spaces*. Australian Mathematical Society Lecture Series. Cambridge University Press, 1987.

[Har15] F.E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer Series in Statistics. Springer-Verlag New York, 2015. DOI: 10.1007/978-3-319-19425-7.

[HB15] Torsten Hildebrandt y Jürgen Branke. «On Using Surrogates with Genetic Programming». En: *Evolutionary Computation* 23.3 (2015), págs. 343-367. DOI: 10.1162/EVCO_a_00133.

[Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[JWP01] Edwin D. de Jong, Richard A. Watson y Jordan B. Pollack. «Reducing Bloat and Promoting Diversity Using Multi-objective Methods». En: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. San Francisco, California, 2001, págs. 11-18. DOI: 10.5555/2955239.2955241.

[KHO19] Jonathan Kelly, Erik Hemberg y Una-May O'Reilly. «Improving Genetic Programming with Novel Exploration - Exploitation Control». En: *Genetic Programming*. Springer International Publishing, 2019, págs. 64-80. DOI: 10.1007/978-3-030-16670-0_5.

[Koz92] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992. DOI: 10.5555/138936.

[KRK15] R. Kalkreuth, G. Rudolph y J. Krone. «Improving Convergence in Cartesian Genetic Programming Using Adaptive Crossover, Mutation and Selection». En: *2015 IEEE Symposium Series on Computational Intelligence*. 2015, págs. 1415-1422. DOI: 10.1109/SSCI.2015.201.

[KS17] A.S. Kulunchakov y V.V. Strijov. «Generation of simple structured information retrieval functions by genetic algorithm without stagnation». En: *Expert Systems with Applications* 85 (2017), págs. 221-230. DOI: https://doi.org/10.1016/j.eswa.2017.05.019.

[LCY16] Q. Li, H. Cheng y M. Yao. «Adaptive Multi-phenotype Based Gene Expression Programming Algorithm». En: *Chinese Journal of Electronics* 25.5 (2016), págs. 807-816. DOI: 10.1049/cje.2016.08.041.

[LHC08] Manuel Lozano, Francisco Herrera y José Ramón Cano. «Replacement strategies to preserve useful diversity in steady-state genetic algorithms». En: *Information Sciences* 178.23 (2008), págs. 4421-4433. DOI: https://doi.org/10.1016/j.ins.2008.07.031.

[Li+08] G. Li y col. «Instruction-matrix-based genetic programming». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 38.4 (2008), págs. 1036-1049. DOI: 10.1109/TSMCB.2008.922054.

[Liu+07] L. Liu y col. «RLGP: An Efficient Method to Avoid Code Bloating on Genetic Programming». En: *2007 International Conference on Mechatronics and Automation*. 2007, págs. 2945-2950. DOI: 10.1109/ICMA.2007.4304028.

[LRW16b] Qiang Lu, Jun Ren y Zhiguang Wang. «Using Genetic Programming with Prior Formula Knowledge to Solve Symbolic Regression Problem». En: *Intell. Neuroscience* 1021378 (2016), 1:1-1:1. DOI: 10.1155/2016/1021378.

[Mar+16] D. Martín y col. «NICGAR: A Niching Genetic Algorithm to mine a diverse set of interesting quantitative association rules». En: *Information Sciences* 355-356 (2016), págs. 208-228. DOI: https://doi.org/10.1016/j.ins.2016.03.039.

[Mc +11] B. Mc Ginley y col. «Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation, and Selection». En: *IEEE Transactions on Evolutionary Computation* 15.5 (2011), págs. 692-714. DOI: 10.1109/TEVC.2010.2046173.

[MKJ12] Alberto Moraglio, Krzysztof Krawiec y Colin G. Johnson. «Geometric Semantic Genetic Programming». En: *Parallel Problem Solving from Nature - PPSN XII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 21-31.

[MT00] Julian F. Miller y Peter Thomson. «Cartesian Genetic Programming». En: *Genetic Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, págs. 121-132. ISBN: 978-3-540-46239-2.

[MWB95] B. McKay, M. J. Willis y G. W. Barton. «Using a tree structured genetic algorithm to perform symbolic regression». En: *First International Conference on*

Genetic Algorithms in Engineering Systems: Innovations and Applications. 1995, págs. 487-492.

[Nic+15] M. Nicolau y col. «Guidelines for defining benchmark problems in Genetic Programming». En: *2015 IEEE Congress on Evolutionary Computation (CEC)*. 2015, págs. 1152-1159. DOI: 10.1109/CEC.2015.7257019.

[OR01] M. O'Neill y C. Ryan. «Grammatical evolution». En: *IEEE Transactions on Evolutionary Computation* 5.4 (2001), págs. 349-358. DOI: 10.1109/4235.942529.

[PA16] Mateusz Pawlik y Nikolaus Augsten. «Tree edit distance: Robust and memory-efficient». En: *Information Systems* 56 (2016), págs. 157-173. DOI: https://doi.org/10.1016/j.is.2015.08.004.

[PM16] Léo Françoso dal Piccol Sotto y Vinícius Veloso de Melo. «Studying bloat control and maintenance of effective code in linear genetic programming for symbolic regression». En: *Neurocomputing* 180 (2016), págs. 79-93. DOI: https://doi.org/10.1016/j.neucom.2015.10.109.

[QHL15] L. Qu, C. Hongbing y H. X. Lin. «Edit distance based crossover operator in gene expression programming». En: *2015 8th International Conference on Biomedical Engineering and Informatics (BMEI)*. 2015, págs. 468-472. DOI: 10.1109/BMEI.2015.7401550.

[Rue+19] R. Rueda y col. «Straight line programs for energy consumption modelling». En: *Applied Soft Computing* 80 (2019), págs. 310-328. DOI: https://doi.org/10.1016/j.asoc.2019.04.001.

[RY98] E. S. Ristad y P. N. Yianilos. «Learning string-edit distance». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.5 (1998), págs. 522-532. DOI: 10.1109/34.682181.

[SE10] Selmar K. Smit y Agoston E. Eiben. «Using Entropy for Parameter Analysis of Evolutionary Algorithms». En: *Experimental Methods for the Analysis of Optimization Algorithms*. Berlin, Heidelberg: Springer, 2010, págs. 287-310. DOI: 10.1007/978-3-642-02538-9_12.

[Seg+17] C. Segura y col. «Improving Diversity in Evolutionary Algorithms: New Best Solutions for Frequency Assignment». En: *IEEE Transactions on Evolutionary Computation* 21.4 (2017), págs. 539-553. DOI: 10.1109/TEVC.2016.2641477.

[Urs02] Rasmus K. Ursem. «Diversity-Guided Evolutionary Algorithms». En: *Parallel Problem Solving from Nature — PPSN VII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, págs. 462-471. DOI: 10.1007/3-540-45712-7_45.

[Uy+10] Nguyen Quang Uy y col. «Semantic Similarity Based Crossover in GP: The Case for Real-Valued Function Regression». En: *Artifical Evolution*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, págs. 170-181. DOI: 10.1007/978-3-642-14156-0_15.

[WSB76] M.S Waterman, T.F Smith y W.A Beyer. «Some biological sequence metrics». En: *Advances in Mathematics* 20.3 (1976), págs. 367-387. DOI: 10.1016/0001-8708(76)90202-4.

Bibliografía

[Age] International Energy Agency. *Buildings. A source of enormous untapped efficiency potential*. https://www.iea.org/topics/buildings.

[Age19a] International Energy Agency. *Tracking Buildings 2019*. https://www.iea.org/reports/tracking-buildings-2019. 2019.

[Age19b] International Renewable Energy Agency. *Global energy transformation: A roadmap to 2050*. https://www.irena.org/publications/2019/Apr/Global-energy-transformation-A-roadmap-to-2050-2019Edition. 2019.

[APM08] C. L. Alonso, J. Puente y J. L. Montaña. «Straight Line Programs: A New Linear Genetic Programming Approach». En: *2008 20th IEEE International Conference on Tools with Artificial Intelligence*. Vol. 2. 2008, págs. 517-524.

[BD16] Peter J. Brockwell y Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-29854-2.

[Ben+20] Domenico Benvenuto y col. «Application of the ARIMA model on the COVID-2019 epidemic dataset». En: *Data in Brief* 29 (2020), pág. 105340. DOI: https://doi.org/10.1016/j.dib.2020.105340.

[BK13a] Florian Benz y Timo Kötzing. «An Effective Heuristic for the Smallest Grammar Problem». En: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. GECCO '13. Amsterdam, The Netherlands, 2013, págs. 487-494. DOI: 10.1145/2463372.2463441.

[BRJ94] George E. P. Box, Gregory C. Reinsel y Gwilym M. Jenkins. *Time series analysis : forecasting and control*. Englewood Cliffs, NJ, 1994.

[BS84] L. Babai y E. Szemeredi. «On The Complexity Of Matrix Group Problems I». En: *25th Annual Symposium onFoundations of Computer Science, 1984.* 1984, págs. 229-240.

[CH12] Samprit Chatterjee y Alis S. Hadi. *Regression Analysis by Example*. 2012. ISBN: 978-0-470-90584-5.

[Cha16] United Nations Climate Change. *The Paris Agreement*. https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement. Nov. de 2016.

[Cho56] N. Chomsky. «Three models for the description of language». En: *IRE Transactions on Information Theory* 2.3 (1956), págs. 113-124.

[Com] European Commission. *A new Horizon for Europe*. https://op.europa.eu/en/publication-detail/-/publication/00d78651-a037-11e8-99ee-01aa75ed71a1/language-en/format-PDF/source-77975709.

[ES03] A. Eiben y Jim Smith. *Introduction To Evolutionary Computing*. Vol. 45. Ene. de 2003. DOI: 10.1007/978-3-662-05094-1.

[GKB17] Ali Ghahramani, Simin Ahmadi Karvigh y Burcin Becerik-Gerber. «HVAC system energy optimization using an adaptive hybrid metaheuristic». En: *Energy and Buildings* 152 (2017), págs. 149-161. DOI: https://doi.org/10.1016/j.enbuild.2017.07.053.

[Goc+14] Snezhana Georgieva Gocheva-Ilieva y col. «Time series analysis and forecasting for air pollution in small urban area: an SARIMA and factor analysis approach». En: *Stochastic Environmental Research and Risk Assessment* 28 (2014), págs. 1045-1060. DOI: 10.1007/s00477-013-0800-4.

[Gre18] Cosima Gretton. «Trust and Transparency in Machine Learning-Based Clinical Decision Support». En: *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*. Ed. por Jianlong Zhou y Fang Chen. 2018, págs. 279-292. DOI: 10.1007/978-3-319-90403-0_14.

[Gui+18] Riccardo Guidotti y col. «A Survey of Methods for Explaining Black Box Models». En: *ACM Comput. Surv.* 51.5 (2018). DOI: 10.1145/3236009.

[Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[Jin+19] Ying Jin y col. *Bayesian Symbolic Regression*. 2019. arXiv: 1910.08892 [stat.ME].

[Kar+12] Dervis Karaboga y col. «Artificial bee colony programming for symbolic regression». En: *Information Sciences* 209 (2012), págs. 1-15. DOI: https://doi.org/10.1016/j.ins.2012.05.002.

[KC19] Tae-Young Kim y Sung-Bae Cho. «Predicting residential energy consumption using CNN-LSTM neural networks». En: *Energy* 182 (2019), págs. 72-81. DOI: https://doi.org/10.1016/j.energy.2019.05.230.

[Koz92] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992. DOI: 10.5555/138936.

[KV98] M.A. Kaboudan y M.K. Vance. «Statistical Evaluation of Symbolic Regression Forecasting of Time-Series». En: *IFAC Proceedings Volumes* 31.16 (1998), págs. 275-279. DOI: https://doi.org/10.1016/S1474-6670(17)40494-0.

[Li+08] G. Li y col. «Instruction-matrix-based genetic programming». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 38.4 (2008), págs. 1036-1049. DOI: 10.1109/TSMCB.2008.922054.

[Li+17] Chengdong Li y col. «Building Energy Consumption Prediction: An Extreme Deep Learning Approach». En: *Energies* 10.10 (2017). DOI: 10.3390/en10101525.

[Liu+19] Tao Liu y col. «A novel deep reinforcement learning based methodology for short-term HVAC system energy consumption prediction». En: *International Journal of Refrigeration* 107 (2019), págs. 39-51. DOI: https://doi.org/10.1016/j.ijrefrig.2019.07.018.

[MCG18] Michael MCGough. *How bad is Sacramento's air, exactly? Google results appear at odds with reality, some say*. https://www.sacbee.com/news/california/fires/article216227775.html. Ago. de 2018.

[McK+10] Robert I. McKay y col. «Grammar-based Genetic Programming: a survey». En: *Genetic Programming and Evolvable Machines* 11.3 (2010), págs. 365-396. ISSN: 1573-7632. DOI: 10.1007/s10710-010-9109-y.

[Mol+17] Miguel Molina-Solana y col. «Data science for building energy management: A review». En: *Renewable and Sustainable Energy Reviews* 70 (2017), págs. 598-609. DOI: https://doi.org/10.1016/j.rser.2016.11.132.

[MT00] Julian F. Miller y Peter Thomson. «Cartesian Genetic Programming». En: *Genetic Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, págs. 121-132. ISBN: 978-3-540-46239-2.

[PLM08b] Riccardo Poli, William Langdon y Nicholas Mcphee. *A Field Guide to Genetic Programming*. Ene. de 2008. ISBN: 978-1-4092-0073-4.

[Rud19] Cynthia Rudin. «Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead». En: *Nature Machine Intelligence* 1.5 (2019), págs. 206-215. DOI: 10.1038/s42256-019-0048-x.

[Rui+16] L.G.B. Ruiz y col. «An Application of Non-Linear Autoregressive Neural Networks to Predict Energy Consumption in Public Buildings». En: *Energies* 9 (ago. de 2016), pág. 684. DOI: 10.3390/en9090684.

[Rui+18] L.G.B. Ruiz y col. «Energy consumption forecasting based on Elman neural networks with evolutive optimization». En: *Expert Systems with Applications* 92 (2018), págs. 380-389. DOI: https://doi.org/10.1016/j.eswa.2017.09.059.

[SC09] Sara Silva y Ernesto Costa. «Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories». En: *Genetic Programming and Evolvable Machines* 10.2 (2009), págs. 141-179. DOI: 10.1007/s10710-008-9075-9.

[SC14] Andrew Stevenson y James R. Cordy. «A survey of grammatical inference in software engineering». En: *Science of Computer Programming* 96 (2014), págs. 444-459. DOI: https://doi.org/10.1016/j.scico.2014.05.008.

[Soa+18] Eduardo Soares y col. «Ensemble of evolving data clouds and fuzzy models for weather time series prediction». En: *Applied Soft Computing* 64 (2018), págs. 445-453. DOI: https://doi.org/10.1016/j.asoc.2017.12.032.

[Sur+15] N. K. Suryadevara y col. «WSN-Based Smart Sensors and Actuator for Power Management in Intelligent Buildings». En: *IEEE/ASME Transactions on Mechatronics* 20.2 (2015), págs. 564-571.

[Tra14] Ministerio Para la Transición Ecológica y el Reto Demográfico. *Hoja de ruta de los sectores difusos a 2020*. 2014.

[VBC18] Effy Vayena, Alessandro Blasimme e I. Glenn Cohen. «Machine learning in medicine: Addressing ethical challenges». En: *PLoS medicine* 15.11 (2018), e1002689-e1002689. ISSN: 1549-1676. DOI: 10.1371/journal.pmed.1002689.

[Whi95] P.A. Whigham. «Grammatically-based Genetic Programming». En: *Workshop on GP: From Theory to Real-World Applications*. 1995, págs. 33-41.

[YM00] Robert Yaffee y Monnie McGee. *Introduction to Time Series Analysis and Forecasting: With Applications of SAS and SPSS*. Ene. de 2000.

[Zaf+17] A. Zafar y col. «A Meta-Heuristic Home Energy Management System». En: *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. 2017, págs. 244-250.