UNIVERSIDAD NACIONAL DE COLOMBIA

Planteamiento de una metodología de medición inferencial mejorada a partir del formalismo de regresión simbólica, un método heurístico de búsqueda.

Ing., Soleyda Manrique Naranjo.

Universidad Nacional de Colombia
Facultad de Ingeniería
Bogotá, Colombia
2015

Planteamiento de una metodología de medición inferencial mejorada a partir del formalismo de regresión simbólica, un método heurístico de búsqueda.

Ing., Soleyda Manrique Naranjo.

Tesis presentada como requisito parcial para optar al título
de:
Magister en Ingeniería - Automatización Industrial

Directora
Ph.D., Maria Alejandra Guzmán Pardo.

Línea de Investigación:
Sensórica inferencial y computación evolutiva.

Universidad Nacional de Colombia
Facultad de Ingeniería
Bogotá, Colombia
2015

Dedicado a
mi familia

... y al verdadero
autor

Resumen

Mediante la combinación de conceptos de computación evolutiva y regresión no lineal haciendo uso de métodos de kernel, se logra proponer un algoritmo híbrido de regresión simbólica, útil en la construcción de modelos de inferencia. El presente trabajo describe el camino de investigación recorrido hasta llegar al planteamiento de un algoritmo que, inspirado en el funcionamiento del algoritmo de regresión simbólica original, genera modelos matemáticos que se adaptan a datos experimentales de forma satisfactoria. Con el fin de mostrar la utilidad de esta técnica de modelado, se introduce el concepto de sensórica inferencial y se evalúa el desempeño de la proposición en este contexto. Los modelos matemáticos generados por el algoritmo propuesto muestran una reducción significativa en el valor del error de estimación respecto a modelos obtenidos mediante la ejecución del algoritmo de regresión simbólica original.

Palabras clave: inferencia, programación genética, regresión simbólica, regresión lineal y no lineal, métodos de kernel.

Abstract

By combining concepts from evolutionary computing and nonlinear regression using kernel methods, it was possible to propose a hybrid symbolic regression algorithm, useful in the construction of inference models. This document describes the whole procedure followed in order to propose, from the original symbolic regression algorithm operation, an algorithm that generates mathematical models which satisfactorily fit experimental data. The concept of inferential measurements is introduced to corroborate the utility of this modeling technique, and the performance of the proposed algorithm is evaluated in this context. Mathematical models generated by the proposed algorithm show a significant reduction in the estimation error, when compared to those obtained by running the original symbolic regression algorithm.

Key words: inference, genetic programing, symbolic regression, linear and nonlinear regression, kernel methods.

Índice general

Índice de figuras

Índice de cuadros

Lista de símbolos

$\hat{y}$	Valor estimado de una variable de salida en el proceso de inferencia.
y	Valor conocido de una variable de salida en el proceso de inferencia. Valor objetivo de la estimación.
$\hat{y}$	Vector de valores estimados de una variable de salida del sistema de inferencia.
y	Vector de valores objetivo de una variable de salida del sistema de inferencia.
$\hat{Y}$	Matriz de datos estimados de salida del sistema de inferencia.
Y	Matriz de valores objetivo de salida del sistema de inferencia.
x_i	Valor de variable de entrada al sistema de estimación.
x	Vector de valores de variables de entrada al sistema de estimación.
X	Matriz de valores de variables de entrada al sistema de estimación.
α_i,β_i	Valores de parámetros o coeficientes dentro del proceso de inferencia.
α,β	Vectores de parámetros o coeficientes dentro del proceso de estimación
w_{ij}	Valor de peso de conexión entre el nodo i y el nodo j en una red neuronal.
w	Vector de valores de peso de conexión entre nodos de una red neuronal.
C, ξ, ξ^*	Término de concesión y variables de holgura contempladas por la técnica de máquinas de vector de soporte.
$f(i,t)$	Valor de aptitud de la solución i perteneciente a la generación t en el proceso de regresión simbólica.

$P(i,t)$ Probabilidad de selección del i-ésimo individuo en la t-ésima generación en el proceso de regresión simbpolica.

$rank(i)$ Rango del i-ésimo individuo dentro del conjunto de elementos en la generación actual.

PS Presión de selección, valor utilizado en el cálculo del rango de un individuo en el proceso de selección por ranking

G Matriz Gram o de producto interno.

$\phi(\cdot)$ Función de mapeo o transporte.

Capítulo 1

Introducción: acerca del trabajo de investigación.

En las siguientes secciones se presenta el concepto y la utilidad de la *medición inferencial*, una prometedora herramienta para el sector industrial. Paso seguido, se describen las principales técnicas empleadas en el diseño de instrumentos para su aplicación, haciendo hincapié sobre el uso de algoritmos de *computación evolutiva*. A partir de allí, se exponen los motivos que llevaron a la propuesta de este proceso investigativo y se dejan claramente estípulados los objetivos a alcanzar con el desarrollo del mismo.

1.1. Contextualización: mediciones inferenciales.

Las mediciones inferenciales han ido convirtiéndose en una herramienta de valor incalculable para la industria. Su principal aplicación apunta a mejorar la calidad de procesos productivos cuando diversos tipos de limitaciones evitan contar con información suficiente del estado de los mismos. En las siguientes secciones se presenta el concepto de medición y sensor inferencial, se resaltan las situaciones en las que su utilidad sale a flote y se describen, brevemente, las principales técnicas utilizadas en su diseño.

1.1.1. Concepto y utilidad de las mediciones inferenciales.

Medir, conocer el valor cambiante de las variables de un proceso en un instante determinado de tiempo, constituye la base fundamental de su control. Por ello el hombre, quien de una u otra manera está en constante búsqueda de métodos que le permitan alcanzar el control de su entorno, ha desarrollado multitud de dispositivos encargados de realizar esta tarea como sensores de temperatura, presión, humedad entre muchos otros.

En el campo industrial, específicamente en el área de control y supervisión, los sensores constituyen una herramienta primordial, al encargarse de detectar magnitudes físicas o químicas y convertirlas en señales eléctricas [Webster & Eren (2014)], de forma tal que sea posible analizarlas, interpretarlas y tomar decisiones convenientes. Dentro de las industrias, suele contarse con una cantidad considerable de dispositivos encargados de realizar tareas de medición. Sin embargo, el rastreo y control de ciertas variables podría verse limitado por alguna de las siguientes razones:

- Inexistencia de dispositivos aptos para realizar la medición: es posible enfrentarse a la necesidad de monitorear valores de una variable, con alta influencia en el desempeño adecuado de un proceso, pero cuya medición no es habitual y por tanto no existe un dispositivo apto para su medida. Es éste el caso, por ejemplo, de quienes se dieron a la tarea de diseñar una herramienta que entrega valores del «número de fluidización», variable de suma importancia dentro del proceso de tratamiento de sólidos en lecho fluidizado ([Botero *et al.* (2009)]), pues a pesar de la relevancia de esta variable dentro de este proceso no era posible su control debido a la inexistencia de un sensor especializado. De forma similar, es posible hallarse en situaciones en las que el dispositivo

efectivamente existe pero el proceso llevado a cabo para realizar la medición es tedioso y no entrega resultados en tiempo real. Esta situación suele encontrarse relatada con frecuencia en la literatura al trabajarse con procesos en los cuales es posible medir cierta variable pero debe hacerse mediante análisis de laboratorio, es decir, fuera de línea ([Ibargüengoytia *et al.* (2013)], [Butler & Zhang. (2012)], [Wu & Luo. (2009)]). El tiempo que tarda la adquisición de las muestras y la ejecución del proceso de medida podría traer consecuencias para el proceso y la calidad del producto.

- Costos elevados de adquisición y mantenimiento de los dispositivos existentes: la búsqueda de un dispositivo de medición de una variable no habitual, si bien puede terminar en la determinación de la inexistencia del mismo, también puede hacerlo en la necesidad de una inversión considerable que no asegura un fin exitoso, pues al tratarse de una variable de baja frecuencia de medición, los procesos asociados a la construcción de los dispositivos podrían no haber sido perfeccionados. Esto conlleva no solo a una alta inversión inicial en la compra del equipo sino también, probablemente, a un gasto constante relacionado con su mantenimiento.[Ibargüengoytia *et al.* (2013)], por ejemplo, expresan su inconformidad con el alto costo de un sensor convencional de viscosidad, además de su bajo desempeño en aplicaciones de control de lazo cerrado.

- Dificultad de posicionamiento de un dispositivo físico en el área de acción de la variable: otro de los inconvenientes que imposibilita el monitoreo de algunas variables dentro de un proceso está relacionado con el medio de acción de la magnitud a medir. Medios adversos o de difícil acceso podrían dificultar el posicionamiento del sensor o bien generar daños en el mismo, lo cual conduciría nuevamente a hablar de inversiones ahora ligadas a reparaciones frecuentes. Precisamente ésta es una de las motivaciones del trabajo realizado por [Pereira *et al.* (2011)], quienes en búsqueda de un sensor para la conductividad del agua, aclaran que dispositivos físicos para esta variable deberían desempeñarse en medios hostiles afectados en gran medida por agentes contaminantes.

Las mediciones inferenciales, llevadas a cabo por sensores inferenciales, virtuales o de software nacen como una solución a estos inconvenientes y basan su funcionamiento en la idea de relación de las variables involucradas en un proceso. Éstos no son más que programas de computador con la capacidad de estimar el valor de cierta variable de proceso, haciendo uso de información generada por otras variables [Espinosa (2004)]. El principio de esta metodología sugiere que aunque cierta magnitud sea difícil de medir físicamente, deben existir

magnitudes medibles cuyo comportamiento se relacione con el de aquella no medible. Por tanto, si se logra descubrir la relación mencionada, es posible inferir el valor de la magnitud problema a partir de los de aquellas variables considerablemente más asequibles y accesibles.

1.1.2. Técnicas de diseño de sensores inferenciales.

De acuerdo con [Kadlec *et al.* (2009)], la idea base detrás de un sensor inferencial es la identificación o modelado de un sistema cuyas entradas son valores de variables de fácil medición dentro del proceso de interés y cuya salida es la variable a inferir. Hace más de dos décadas inició la puesta en marcha de esta práctica con la aparición de una vasta cantidad de trabajos que aglomeran técnicas y metodologías de diseño, proceso que, como un problema de identificación de sistemas, puede ser afrontado desde dos enfoques generales conocidos como *basado en modelo* y *basado en datos*. El diagrama mostrado en la figura 1.1 resume algunas de las principales técnicas, mencionadas en literatura académica y científica, utilizadas en el diseño de sensores inferenciales.

1.1.2.1. Técnicas de diseño basadas en modelo.

En general, el enfoque de identificación de sistemas basado en modelo o caja blanca, llamado así debido a la fácil interpretación de sus expresiones, basa su accionar en la búsqueda de la expresión matemática que relaciona las entradas y la salida de un proceso. Este tipo de modelado parte de leyes y principios fundamentales (físicos, químicos, biológicos y/o económicos), en los que ecuaciones y parámetros son determinados mediante modelamiento teórico [Nelles (2001)].

El diseño de un sensor virtual basado en modelo cuenta con dos interpretaciones diferentes. En la primera de ellas se busca una expresión matemática que presente la variable no medible como función de variables de fácil medición (ver figura 1.2). En la segunda interpretación se parte de una expresión matemática que describe el comportamiento del sistema, donde la salida no es necesariamente la variable a inferir, y a partir de él, mediante un estimador de estado, se infiere el estado de alguna de las variables involucradas (ver figura 1.3).

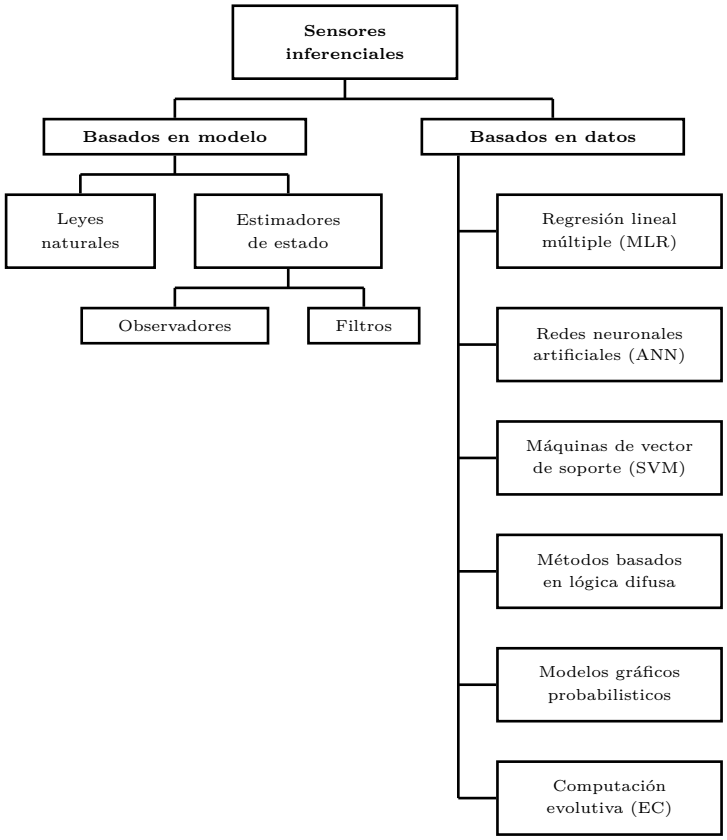


Figura 1.1: Cuadro de técnicas de diseño de sensores inferenciales.

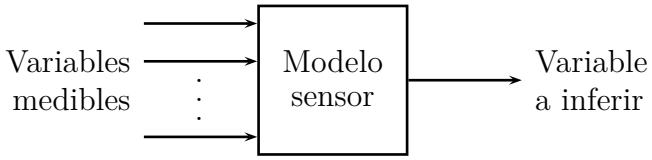


Figura 1.2: Estructura de sensor inferencial basado en modelo matemático.

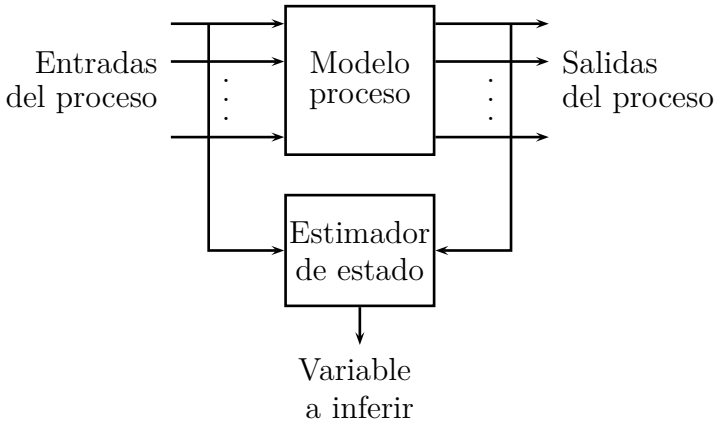


Figura 1.3: Estructura de sensor inferencial basado en estimador de estado.

Estimador de estado: definido por [Ogata (1998)], un estimador de estado es un dispositivo o programa de computador que infiere el valor de las variables de estado de un sistema con base en las mediciones de las variables de salida y de control. El diseño de un estimador de estado supone un conocimiento perfecto del sistema, es decir, se requiere que el modelo que se tiene del mismo refleje con precisión su relación entrada-salida [Colmenares (2006)]. El ideal es obtener, haciendo uso del modelo del proceso, un valor de estados estimado de forma tal que el error de estimación, definido como la diferencia entre el estado real y el estado estimado, tienda a cero en infinito, lo cual implicaría que el estado estimado converge al estado real de forma asintótica [Botero *et al.* (2009)].

Dos grandes grupos o clasificaciones desprenden de la concepción general de estimadores de estado de acuerdo con la teoría base detrás de ellos. Cuando se hace caso omiso de las señales estocásticas que actúan sobre el sistema (perturbaciones y ruidos de medición) se habla de observadores, cuyo líder es el propuesto por [Luenberger (1966)] y cuyo principio de operación yace en un modelo matemático del proceso que corre en paralelo con el proceso mismo. Si el modelo fuese perfecto, el valor de los estados entregados por éste serían idénticos a los estados reales. En práctica, sin embargo, existen errores del modelo que generan una diferencia entre los estados reales y estimados, diferencia que a su vez se refleja en la discrepancia entre la salida del proceso real y la salida del modelo. Esta diferencia es utilizada para actualizar el estimador mediante un valor de ganancia. Por supuesto, la aplicación de un observador da por sentado la observabilidad del sistema, es decir, la posibilidad de estimar sus estados a partir de sus entradas y salidas [Ogata (1998)].

Como alternativa a los observadores y creando la segunda gran clasificación de los estimadores de estado se encuentran los filtros, algoritmos de estimación basados en teoría estocástica. Su principal representante es el filtro de Kalman que produce una estimación de estado que contiene un nivel mínimo de ruido pues considera la presencia de perturbaciones y medidas de ruido aleatorias en el proceso. Este algoritmo, desarrollado por Rudolph E. Kalman en los años 60, es en esencia un conjunto de ecuaciones matemáticas que implementan un tipo de estimador predictor-corrector óptimo en el sentido que minimiza la covarianza del error estimado cuando algunas condiciones son alcanzadas [Bishop & Welch (2001)].

Tanto los observadores como los filtros cuentan con igual fin, encontrar una ganancia de retroalimentación del error que permita minimizar este último. Para los observadores, sin embargo, esta ganancia debe ser hallada manualmente mientras que el filtro de Kalman cuenta con un algoritmo que permite encontrar este valor de forma automática cuando se conoce la varianza de los ruidos que afectan el sistema.

Cabe mencionar que ambas clasificaciones de estimadores fueron creadas en principio para actuar sobre sistemas y modelos lineales. Muchos procesos, sin embargo, no cuentan con esta característica y de allí el nacimiento de modificaciones, o mejoras, que buscan sobrepasar estos inconvenientes y entregar una mejor estimación.

Respecto a las mejoras realizadas al filtro de Kalman, [Botero *et al.* (2009)] resaltan la creación del filtro de Kalman linealizado que hace uso de un modelo linealizado del sistema no lineal, el filtro de Kalman extendido que realiza la linealización alrededor del último valor de estado estimado, filtro de Kalman con restricciones y con modelo restringido que incluyen información valiosa para el proceso que puede llevar a una mejor estimación y el filtro Kalman no lineal con "Unscented Transformation" desarrollado para lidiar con no linealidades no suaves.

De igual forma, [Botero *et al.* (2009)] mencionan la existencia en la literatura de observadores asintóticos (OA), observadores adaptables (Oad), observadores de alta ganancia (OAG) y observadores de modos deslizantes (OMD) entre otros como variaciones a la versión llana del observador de estado.

1.1.2.2. Técnicas de diseño basadas en datos experimentales.

El modelado de sistemas basado en datos, por su parte, basa su funcionamiento en observaciones empíricas del proceso y mediciones de las variables involucradas. Tanto la estructura como los parámetros del modelo son determinados de forma experimental [Nelles (2001)]. La principal característica encontrada en la literatura de este tipo de modelado, y que lo hace tan cotizado, es la escasa o nula necesidad, a diferencia de las técnicas anteriores, de conocimiento a priori sobre la dinámica del sistema a modelar. Este tipo de modelado suele ser considerado de mayor grado de realismo pues al ser datos reales los que conforman la base de la técnica, en el modelo se incluyen dinámicas generalmente eliminadas o difícilmente describibles en modelos deterministas.

La tarea en el modelado basado en datos consiste, fundamentalmente, en aprender, a partir de un conjunto de datos experimentales de entrada-salida, la relación existente entre ellos, de forma tal que al exponer al modelo a nuevas mediciones de las variables de entrada, sea posible inferir el valor de la variable de salida.

(a) Regresión lineal múltiple (MLR - *Multiple linear regression*).

La regresión lineal y todas sus variantes como regresión de componentes principales, regresión por mínimos cuadrados parciales, etc., constituyen los primeros indicios de técnicas de creación de sensores inferenciales a partir de datos experimentales [Espinosa (2004)]. Esta metodología supone el valor de la variable a estimar como una combinación lineal de las señales de entrada o de excitación del sistema (variables medibles):

$$\hat{y} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_N x_N = \alpha^T \mathbf{x} \tag{1.1}$$

Siendo, en 1.1, $\hat{y}$ la salida estimada, $\mathbf{x}$ el vector de variables de entradas, α un vector de parámetros y N el número de variables de entrada, el problema se reduce al hallazgo de los valores del vector α que minimizan la norma del vector diferencia entre los valores históricos (experimentales) y los estimados de la variable. Este proceso es, popularmente,

realizado mediante la técnica de mínimos cuadrados que busca solucionar el siguiente caso de optimización:

$$\min_{\alpha} \frac{1}{2}(\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}) \tag{1.2}$$

El planteamiento mediante mínimos cuadrados permite llegar a la siguiente solución donde $\mathbf{X}$ y $\mathbf{Y}$ corresponden a matrices de datos de entrada y salida respectivamente.

$$\alpha = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{1.3}$$

Esta expresión, sin embargo, supone la no singularidad de la matriz $(\mathbf{X}^T\mathbf{X})$, situación no siempre probable debido a la posibilidad de contar con variables linealmente dependientes dentro del proceso. Es ésta la razón de la aparición de procesos variantes de modelado como regresión de componentes principales (PCR - *Principal component regression*) y regresión por mínimos cuadrados parciales (PLSR - *Partial least square regression*) que incluyen procesos de reducción de dimensionalidad y análisis de correlación.

Mientras que la regresión lineal múltiple y sus derivados cuentan con la simplicidad como ventaja más sobresaliente, el hecho de incluir la palabra lineal revela su mayor falencia pues con frecuencia la realidad no coincide con este nivel de perfección.

(b) Redes neuronales artificiciales (ANN - *Artificial neural networks*).

Inspiradas en la infinitamente compleja estructura cerebral de los seres humanos, cuyos primeros intentos de modelado fueron hechos por los neurólogos McCulloch y Pitts en 1943 [Marsland (2009)], las redes neuronales artificiales han sido las innegables ganadoras en popularidad en la elaboración de sensores inferenciales ([Gómez & Sanchez (2011)], [Pereira *et al.* (2011)], [Richter *et al.* (2010)]).

Una red neuronal artificial, específicamente aquella estructura conocida como perceptrón multicapa (MLP - *Multilayer perceptron*) propuesta por Rumelhart, Hinton y McCle-

lland [Marsland (2009)], consiste en un conjunto de capas interconectadas de nodos, neurodos o perceptrones que se activan ante la superación de un umbral determinado por una función de activación. Estas estructuras son entrenadas para responder de una forma u otra ante ciertos estímulos de entrada. El entrenamiento consiste en el hallazgo de los pesos de interconexión de los neurodos que generan la respuesta deseada.

El proceso de diseño de un sensor inferencial mediante redes neuronales consiste en el entrenamiento, a partir de datos de entrada y salida conocidos del sistema, de cierta estructura de perceptrones interconectados, de forma tal que al llegar un nuevo dato de entrada, la red sepa cómo responder y entregue un valor de variable estimada.

Generalmente, la estructura utilizada en la implementación de redes neuronales en la identificación de sistemas y por ende en el diseño de sensores inferenciales consiste en una capa intermedia de perceptrones con funciones de activación sigmoidales o tangenciales y un único perceptrón en la capa de salida con función de activación lineal [Marsland (2009)]. La figura 1.4 presenta la estructura de red general mencionada.

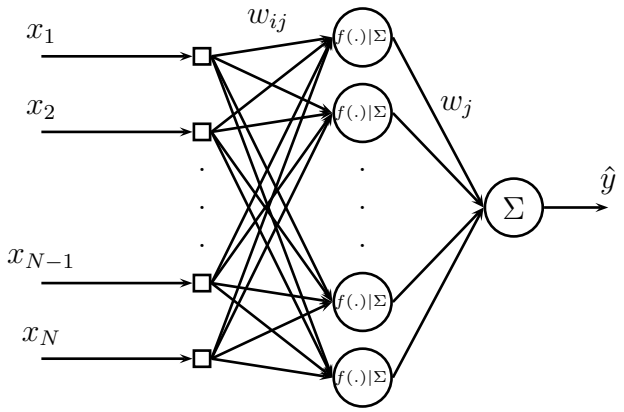


Figura 1.4: Estructura de sensor inferencial basado en red neuronal artificial.

A partir de la estructura presentada en la figura 1.4 puede definirse la salida de la red neuronal como sigue, siendo α un sesgo considerado, N el número de variables medibles o datos de entrada y M el número de nodos en la capa oculta:

$$\hat{y} = \sum_{j=1}^{M} w_j . f(\sum_{i=1}^{N} w_{ij} \cdot x_i) + \alpha \tag{1.4}$$

El objetivo de entrenamiento de una red neuronal tipo perceptrón multicapa, cuyo proceso es descrito en [Babuska (2004)], es encontrar los pesos w_{ij} y w_j que minimizan la siguiente función de costo correspondiente al error de aproximación (diferencia entre el valor deseado en la salida (y_k) y el valor obtenido ($\hat{y}_k$)), siendo k cada conjunto de datos de entrenamiento y K el número total de estos últimos:

$$J(w) = \frac{1}{2} \sum_{k=1}^{K} (y_k - \hat{y}_k)^2 \tag{1.5}$$

(c) Máquinas de vector de soporte (SVM - *Support vector machines*).

El objetivo inicial de la regresión mediante máquinas de vectores de soporte propuesta por [Vapnik (1995)], es el hallazgo de una función $f(x)$ con la siguiente estructura:

$$f(\mathbf{x}) = < \mathbf{w}, \mathbf{x} > + b \tag{1.6}$$

Siendo $\mathbf{x}$ el vector de datos de entrada.

Esta técnica pretende que la función $f(\mathbf{x})$ cuente con un valor admisible de desviación (ϵ) respecto a la función objetivo a aproximar:

$$|y - f(\mathbf{x})| = |y - < \mathbf{w}, \mathbf{x} > + b| < \epsilon \tag{1.7}$$

Siendo y el valor objetivo de la predicción. Lo anterior implica que en la regresión realizada por máquinas de vector de soporte, no son importantes los errores siempre y cuando estos sean menores a un valor considerado aceptable.

El entrenamiento de una SVM se convierte en un problema de optimización, en el que se buscan los valores de $\mathbf{w}$ y b que minimicen la norma de $\mathbf{w}$, valor directamente relacionado con la llanura de la función $f(\mathbf{x})$. Si este problema tiene solución, es decir,

existe una función $f(\mathbf{x})$ que aproxima los pares de datos de entrada salida (x_i, y_i) con un valor de precisión ϵ, se dice factible, de lo contrario, es necesario introducir en la función objetivo, un término de holgura que determina la concesión entre el grado de minimización de la función objetivo y el grado de desviación permitido [Donís Díaz *et al.* (2003)].

La función objetivo final a minimizar en el entrenamiento de una máquina de vector de soporte es, entonces:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{l} \xi_i + \xi_i^* \tag{1.8}$$

Sujeto a:

$$|y_i - < \mathbf{w}, \mathbf{x} > -b| \leq \epsilon + \xi_i \tag{1.9a}$$

$$\xi_i, \xi_i^* > 0 \tag{1.9b}$$

Donde ξ_i y ξ_i^* son variables de holgura y C el término de concesión.

Este problema de optimización suele ser resuelto mediante la definición del problema dual descrito en [Smola & Schölkopf (2004)] y el uso de multiplicadores de Lagrange.

(d) Sistemas difusos.

Basados en la interpretación y tratamiento lingüístico de los datos, los sistemas difusos, al igual que las redes neuronales artificiales, tienen como fundamento la aproximación al funcionar de la actividad humana. La lógica difusa, introducida por [Zadeh (1965)] y raíz del modelado difuso, parte de la idea de relatividad de las circunstancias. Relatividad que es representada mediante el uso de conjuntos difusos caracterizados por contar con límites solapados [Babuska (2004)]. Mientras que en la lógica convencional los elementos de un conjunto pertenecen o no a él, en la lógica difusa estos mismos elementos

pueden pertenecer con diferente nivel de compromiso a un grupo en particular.

El proceso de aproximación o identificación de sistemas mediante lógica difusa parte de un proceso de inferencia en el que mediante relaciones descritas por reglas de la forma ***si*** *antecedente* ***entonces*** *consecuente* se posicionan las entradas definidas en el antecedente en una parte particular del espacio de salida.

Dentro del proceso de inferencia a partir de lógica difusa, sobresalen dos enfoques principales, un enfoque lingüístico, liderado por [Mamdani (1977)], en el que tanto el antecedente como el consecuente son representados mediante la asociación de una etiqueta lingüística y uno más, conocido como Takagi-Sugeno (TS) en el que al consecuente le es asociada una expresión matemática [Takagi & Sugeno (1985)].

Ejemplo de una expresión con estructura Mamdani es la siguiente:

$$\textbf{\textit{If}}\ x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } + \ldots + \text{ and } x_N \text{ is } A_N \quad \textbf{\textit{then}}\ y \text{ is } B$$

Por su parte, una expresión tipo Takagi-Sugeno tomaría la siguiente estructura:

$$\textbf{\textit{If}}\ x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } + \ldots + \text{ and } x_N \text{ is } A_N\ \textbf{\textit{then}}\ y \text{ is}$$
$$\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_N x_N$$

Esta última expresión presenta un alto nivel de similitud a la regresión multivariable pues este concepto no se aleja de estas raíces, un proceso de modelado de funciones mediante enfoque Takagi-Sugeno no es más que un proceso de regresión separado por regiones delimitadas por la base de reglas.

Un sistema de inferencia difuso es constituido por tres bloques principales. Un bloque de fuzzificación, en el que la(s) entrada(s) toman una representación simbólica mediante su evaluación en funciones de pertenencia, un bloque de inferencia en el que el grado de cumplimiento de una base de reglas es calculado entregando una salida en particular y un bloque de defuzzificación en el que esta salida es convertida de su representación simbólica a un valor conciso.

(e) Métodos gráficos.

Esta técnica de modelado consiste en el uso de grafos, nodos unidos mediante arcos, que codifican distribuciones de probabilidad. Cada nodo representa una variable aleatoria, generalmente discreta, unida a otros nodos (padres e hijos) mediante arcos que representan relaciones de dependencia. Las redes bayesianas, gráficos dirigidos acíclicos a los que le son introducidos tablas de distribución de probabilidad, conforman el modelo gráfico más general [Marsland (2009)].

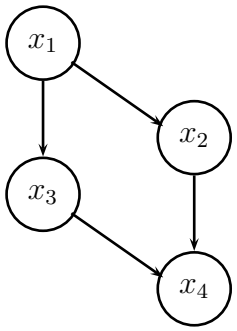


Figura 1.5: Modelo gráfico dirigido.

Cuando se implementan redes bayesianas se asume que un nodo o variable depende exclusivamente de sus padres y cada nodo se asocia a una tabla de probabilidad condicionada que define su probabilidad de estado de acuerdo con el estado de sus predecesores [Duda *et al.* (2000)]. De allí, el que las redes bayesianas entreguen, tanto una descripción cualitativa al mostrar mediante el grafo las relaciones entre variables, como cuantitativa al introducir valores de probabilidad.

Las redes bayesianas y en general los modelos gráficos probabilísticos son usados con frecuencia en la realización de inferencia a partir de observaciones, al obtener la probabilidad de ocurrencia de un hecho en particular. Este proceso de inferencia consiste en la propagación de evidencia (estado de algunas de las variables) a través de una red para, a partir de las relaciones entre variables y los valores de probabilidad condicionada, hallar la probabilidad a posteriori de la variable a inferir [Sucar (2006)].

(f) Computación evolutiva.

En las últimas décadas, el concepto de *computación evolutiva* ha sido incluido en el área de sensórica o medición inferencial. Este término envuelve un conjunto de algoritmos que constituyen una abstracción de la teoría de la evolución, propuesta por Charles Darwin, aplicada a la búsqueda de soluciones óptimas a problemas. Aunque sus primeros indicios radican en los años 30's no fue sino hasta 1960 cuando el aumento en la disponibilidad de computadores digitales permitió su uso como herramienta de modelado y simulación y activó el interés de estudiosos en diferentes esquinas del mundo quienes presentaron sus interpretaciones del concepto [Langdon *et al.* (2008)]. Es así como nacen los tres grandes componentes de la computación evolutiva. En la Universidad Técnica de Berlín, Rechenberg y Schwefel formulan ideas acerca de la posibilidad de utilizar procesos evolutivos en la solución de problemas de optimización paramétrica [Rechenberg (1971)]. Éstas son las raíces de un conjunto de algoritmos conocidos como «estrategias evolutivas». De forma paralela, [Fogel *et al.* (1966)] proponen la «programación evolutiva» como una forma de alcanzar objetivos de la inteligencia artificial mediante la aplicación de técnicas evolutivas y [Holland (1975)] en la Universidad de Michigan desarrolla las bases del algoritmo genético simple en búsqueda de una forma de diseñar e implementar sistemas adaptativos robustos.

A pesar de ser $1950 - 1960$ la década inicial de la computación evolutiva, no fue sino hasta 1980 que se obtuvieron resultados satisfactorios [Coello & Zacatenco (2004)]. Uno de estos resultados es la *programación genética (GP - Genetic programming)* propuesta por [Koza (1989)]. Técnicamente, la programación genética es un algoritmo evolutivo que considera una población de programas de computador que de forma iterativa, generación tras generación, es transformada en otras poblaciones de programas mediante la aplicación de operadores genéticos [Langdon *et al.* (2008)].

- ***Computación evolutiva y programación genética en el diseño de sensores inferenciales.***

Los conceptos de técnicas evolutivas descritos han sido aplicados en las últimas

décadas en el diseño de sensores inferenciales gracias a una aplicación específica de la programación genética conocida como regresión simbólica [Koza (1992)]. A diferencia de los métodos de regresión convencional que suponen un modelo determinado y se enfocan en el hallazgo de valores de parámetros, la regresión simbólica trabaja por hallar, no sólo los mejores valores de parámetros, sino también la mejor expresión matemática que presente una variable dependiente como función de otras variables independientes. Esto quiere decir que no es necesario una suposición previa de la forma de la función. En síntesis, la regresión simbólica pretende encontrar una expresión matemática que provea una buena, mejor o perfecta aproximación entre un número finito de valores muestreados de variables independientes y los valores asociados de la variable dependiente.

La regresión simbólica es una técnica evolutiva o iterativa de regresión, dirigida por el error de aproximación [Kumar *et al.* (2014)]. En ella se parte de un conjunto o población aleatoria de expresiones matemáticas o individuos codificadas en «genes» con estructuras de árbol y se crean nuevas expresiones en cada iteración que, en teoría, se desenvuelven mejor en la tarea de ajustarse a una función original. Estas expresiones son generadas a partir de expresiones «padres» mediante la aplicación de operadores genéticos que cuentan, además, con cierta probabilidad de ocurrencia asociada.

1.2. Motivación.

Los instrumentos de medición inferencial constituyen una herramienta prometedora en la mejora de la calidad de procesos productivos pues contribuyen a la consecución de mayor volumen de información relevante en la toma de decisiones. Si bien es posible trabajar con una amplia variedad de técnicas en el diseño de estas herramientas, la popularidad de los algoritmos pertenecientes al paradigma de computación evolutiva ha ido en aumento gracias a resultados exitosos en la solución de problemas complejos ([Kumar *et al.* (2014)], [Faris & Sheta (2013)], [Gondro & Kinghorn (2008)]). Estos algoritmos, además, cuentan con una amplia cantidad de características y peculiaridades con las que es posible interactuar en búsqueda de otros comportamientos. El algoritmo de regresión simbólica, concebido con fines específicos de modelado, no es la excepción en este sentido y a pesar de dar

muestra de buenos resultados en su labor ([Sharma & Tambe (2014)], [Bolshakov (2013)], [Byington *et al.* (2012)]), dichas peculiaridades han llevado a investigadores a proponer variaciones sobre el concepto inicial de este algoritmo en búsqueda de resultados aún más satisfactorios ([Kumar *et al.* (2014)], [Gonçalves & Silva (2013)], [Kommenda *et al.* (2013)], [Martínez *et al.* (2011)], [Searson *et al.* (2010)], [Lopes & Weinert (2004)]).

El presente trabajo es motivado por la posibilidad de converger, mediante un proceso investigativo, a una modificación al algoritmo de regresión simbólica original que genere un mejor comportamiento de la herramienta en el hallazgo de modelos útiles en tareas de medición inferencial. Un mejor comportamiento implica ya sea un algoritmo con mayor nivel de autonomía debido a un menor número de parámetros a sintonizar o bien un algoritmo que genere modelos que se ajusten con mayor precisión a los datos experimentales tratados, o modelos que generalicen en mayor medida y no entreguen resultados de inferencia satisfactorios tan solo con los datos experimentales a partir de los cuales son construidos.

1.3. Objetivos de la tesis.

El objetivo general de esta investigación consiste en el desarrollo de un algoritmo, basado en el algoritmo de regresión simbólica original, que mejore los resultados obtenidos por este último en el ejercicio de modelado, con fines aplicativos al proceso de medición inferencial, de acuerdo con criterios como precisión, capacidad de generalización, confiabilidad o autonomía.

Especificamente se pretende:

1. Comprender el funcionamiento del algoritmo original de regresión simbólica y establecer la influencia de sus parámetros en los resultados de inferencia que se obtienen al aplicarlo.

2. Plantear e implementar modificaciones al algoritmo original de regresión simbólica y proponer un algoritmo final de regresión que mejore su desempeño en procesos de estimación ya sea en términos de precisión, generalización, confiabilidad o autonomía.

3. Comparar el desempeño del algoritmo final de regresión propuesto respecto al algoritmo de regresión simbólica original en la estimación de una variable de proceso.

1.4. Organización del documento.

Este documento recopila el proceso de investigación llevado a cabo en búsqueda del cumplimiento de los objetivos mencionados en la sección anterior y se estructura en cinco capítulos.

- Capítulo 1. Introducción: con fines de contextualización, se presenta el concepto de *medición inferencial*, su utilidad y principales técnicas de diseño de herramientas encargadas de efectuar esta tarea. Dentro de estas técnicas se realza la noción de *computación evolutiva* y a partir de allí se establecen los motivos para llevar a cabo el proceso investigativo y se aclaran los objetivos a alcanzar al final del mismo.

- Capítulo 2. Fundamentación: en esta sección se extiende la descripción del paradigma de computación evolutiva y de su rama *programación genética* enfatizando la aplicación de esta última en tareas de medición inferencial a través del concepto de *regresión simbólica*. De igual forma se presenta el funcionamiento del algoritmo encargado de ejecutar este tipo de regresión junto con la totalidad de parámetros involucrados y se establece la influencia de los mismos en el proceso de modelado.

- Capítulo 3. Modificación: el alto número de parámetros con que cuenta el algoritmo de regresión simbólica lo hacen un algoritmo flexible y apto para estudiar variaciones potenciales que mejoren su desempeño. En esta sección se presentan, de forma general, algunas modificaciones contempladas por otros autores. A partir de una de estas modificaciones, surge un interrogante cuya respuesta se convierte en el foco central de esta investigación. Este capítulo contiene, además, la matemática necesaria para comprender, paso seguido, la variación o modificación propuesta.

- Capítulo 4. Verificación: este capítulo resume los resultados de la aplicación, tanto del algoritmo original de regresión simbólica como del algoritmo modificado propuesto, dentro de algunos procesos de modelado. Lo anterior con la finalidad de verificar la consecución de resultados satisfactorios, en mayor grado, al considerar la idea descrita en el capítulo precedente. De igual forma se compara el funcionamiento del algoritmo

concebido respecto a otro algoritmo de regresión considerado dentro de las técnicas populares de diseño de sensores inferenciales.

- Capitulo 5. Conclusión: en esta sección se discute acerca de los resultados obtenidos en el proceso investigativo y presentados en secciones anteriores. Nacen, además, ciertos interrogantes e inquietudes que llevan a plantear posibles temas para trabajo futuro.

Capítulo 2

Fundamentación: computación evolutiva y el algoritmo original de regresión simbólica en el proceso de inferencia de medidas.

Este capítulo presenta la descripción del funcionamiento del algoritmo de regresión simbólica original propuesto por [Koza (1992)], enfocado a desarrollar tareas de medición inferencial. Su implementación en MATLAB permite argumentar, a partir de experiencias de interacción con el mismo, acerca de la influencia de los parámetros involucrados. En circunstancias pertinentes, ejercicios sencillos son desarrollados con la intensión de mostrar al lector dicha influencia. Estos ejercicios son desarrollados con datos sintéticos que no corresponden, imperativamente, a variables de proceso reales. El contexto de los ejercicios es creado intuitivamente, a partir de resultados obtenidos mediante pruebas experimentales y valores de parámetros utilizados por el autor del algoritmo en el desarrollo de sus propios ejemplos.

2.1. La computación evolutiva en el proceso de inferencia.

Incurrir en el proceso de medición inferencial supone arribar al hallazgo de un modelo, o expresión matemática, que se ajuste a la definición del valor de una variable dependiente a partir de su correspondiente conjunto de valores de variables independientes, todas estas actuando dentro de un sistema en particular. Una vez dicho modelo ha sido hallado para ciertos valores de datos experimentales, éste puede ser usado en la inferencia de valores futuros de la variable dependiente del sistema [Koza (1992)].

Al ser imperativo el hallazgo de un modelo que presente la relación entre las variables involucradas dentro de un sistema y que permita inferir el estado de una de ellas a partir del estado de las restantes, lo es, de igual forma, la búsqueda del mismo. Esta búsqueda ha sido enfrentada, en algunos casos, mediante la inclusión de técnicas heurísticas ([Sharma & Tambe (2014)],[Byington *et al.* (2012)],[Smits & Kordon (2008)]), entendiéndose esto como técnicas de búsqueda que reúnen «criterios, métodos o principios prácticos que llevan a la toma de una decisión, entre múltiples alternativas, que promete ser más efectiva en alcanzar un objetivo» [Pearl (1984)].

Para el proceso de medición inferencial o, simplemente, para el proceso de búsqueda de un modelo matemático, la decisión a tomar es precisamente qué modelo entre un conjunto de modelos potenciales es más efectivo en el ajuste de ciertos datos experimentales u observaciones de las variables involucradas.

Como un constituyente de estas técnicas heurísticas de búsqueda han sido considerados los algoritmos enmarcados bajo el concepto de *computación evolutiva* que, contando con teorías basadas en el principio de evolución de las especies y de sobrevivencia de los ejemplares con mayor habilidad de adaptación promulgada por Charles Darwin en 1859, proponen una metodología en la que una población inicial de soluciones potenciales, en este caso específico modelos potenciales, es evolucionada y modificada «genéticamente» hasta obtener una solución que cumpla en un grado satisfactorio con los requisitos de búsqueda en cuestión, en este caso un bajo error de estimación.

Dentro de este paradigma de búsqueda de soluciones mediante soluciones potenciales en evolución, métodos como los algoritmos genéticos [Holland (1975)], la programación evolutiva [Fogel *et al.* (1966)] y la programación genética [Koza (1989)] resaltan como los más conocidos [Michalewicz (1996)]. Estos algoritmos, cada uno con sus particularidades, cuentan con las siguientes características comunes:

- Una representación genética de las soluciones candidatas.
- Una forma de crear una población inicial de soluciones candidatas.
- Una función de evaluación que permita calificar las soluciones de acuerdo con su grado de aptitud como solución.
- Operadores que alteren la información genética contenida en las representaciones.
- Valores para ciertos parámetros como tamaño de la población, probabilidades de acción de cada operador, etc.

2.2. La programación genética en el proceso de inferencia.

En la búsqueda metódica de soluciones dentro de un espacio de soluciones potenciales, el paradigma de *programación genética* propone considerar cada posible solución como un programa jerárquico de computador cuyo tamaño, forma y contenido dependerán del dominio del problema que se enfrenta [Koza (1989)].

La descripción dada por su autor [Koza (1989)], plantea el inicio del proceso generacional de búsqueda a partir de un conjunto aleatorio de programas de computador compuestos por funciones y terminales acordes al problema tratado. Cada programa es, posteriormente, medido en términos de su aptitud en la solución del problema en cuestión para, a partir de dicha medida e introduciendo el principio Darwiniano de reproducción y supervivencia de aquel con mayor capacidad de adaptación y el operador genético de recombinación sexual o cruce, crear una nueva población de programas mediante la manipulación de la población

en la generación actual.

La operación de reproducción mencionada consiste en la selección, de forma proporcional a la aptitud, de los mejores individuos de la población actual para ser transportados, sin alteración alguna, a la población que constituirá la siguiente generación. Por su parte, el proceso de recombinación o cruce es usado para crear, a partir de dos soluciones o programas de computador «padres» seleccionados de acuerdo con su medida de aptitud y con longitudes y formas generalmente diferentes, programas «hijos» o descendientes compuestos a partir de subexpresiones de sus padres. Esto con la esperanza de obtener nuevos programas con mayor nivel de aptitud en la solución del problema planteado.

Después de creada una nueva población mediante la aplicación de estos dos operadores genéticos a programas de la población actual, esta última es reemplazada por los constituyentes de la nueva generación quienes, de igual forma, serán medidos y calificados de acuerdo con su habilidad en la solución del problema considerado, permitiendo así, realizar este proceso de forma cíclica hasta llegar a la creación de un programa de computador que realice la tarea de forma suficientemente satisfactoria.

Los pasos a seguir en el desarrollo de todos los algoritmos pertenecientes al paradigma de la programación genética pueden ser resumidos de la siguiente manera:

1. Generar, de forma aleatoria, una población inicial que contenga las funciones y terminales involucrados en el problema.

2. De forma iterativa realizar los siguientes pasos hasta alcanzar un criterio de terminación determinado por el diseñador :

    a) Ejecutar cada uno de los programas de la población inicial y asignar a ellos un valor de aptitud de acuerdo con su desempeño en el desarrollo de la tarea considerada.

b) Construir una nueva población de programas de computador mediante la aplicación de las dos operaciones genéticas básicas. Dichas operaciones son aplicadas a programas seleccionados de acuerdo con su alta calificación de desempeño.

1) Copiar en la nueva población algunos individuos de la población actual con buen desempeño (reproducción).

2) Construir nuevos programas de computador mediante la recombinación genética de segmentos, elegidos de forma aleatoria, de dos programas existentes (cruce).

3. Designar uno de los individuos como la solución al problema tratado.

Gráficamente, el paradigma de programación genética es presentado por su autor de acuerdo con el diagrama de flujo mostrado en la figura 2.1.

Ahora bien, gracias a un algoritmo en particular, perteneciente al paradigma de la programación genética, diversos autores han podido incluir conceptos de búsqueda poblacional al enfrentarse a tareas de estimación [Kaydani *et al.* (2014)], [Bolshakov (2013)], [Kordon *et al.* (2004)].

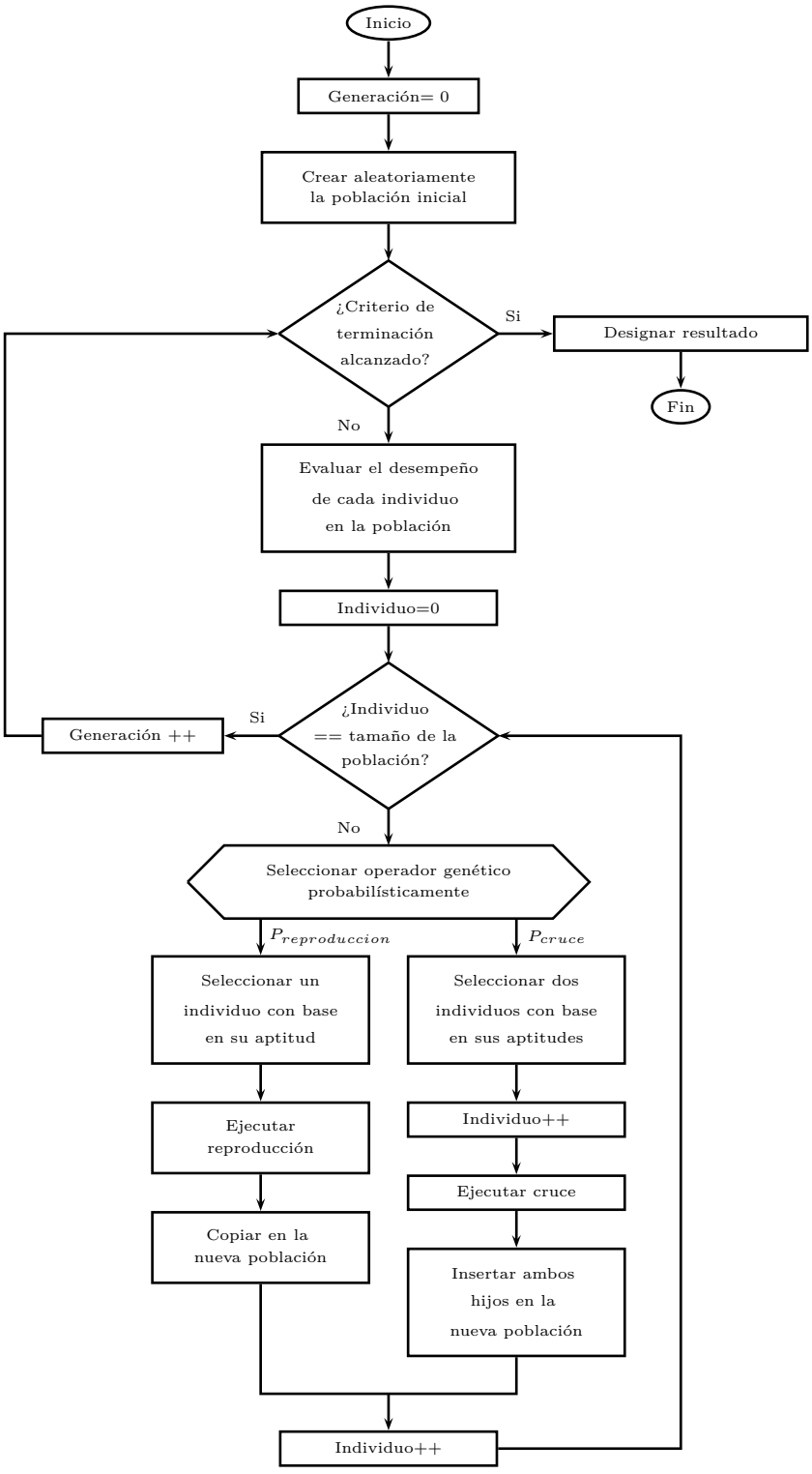


Figura 2.1: Diagrama de flujo del paradigma de programación genética. Tomado de [Koza (1992)].

2.3. Algoritmo de regresión simbólica: un componente de la programación genética.

Propuesto y descrito por [Koza (1992)] en su documento titulado «Genetic programming: on the programming of computers by means of natural selection», el algoritmo de regresión simbólica constituye una aplicación del paradigma de la computación evolutiva en la que se pretende descubrir la forma funcional correcta de una expresión matemática, junto con sus coeficientes numéricos, que se adapte en mayor medida a un conjunto finito de datos contemplado [Koza (1992)].

Lo anterior significa, en el contexto de las mediciones inferenciales, que al contar con un conjunto o base de datos experimentales de variables involucradas en un sistema, este algoritmo se encargaría de hallar una expresión matemática que se ajuste a los mismos en la mayor medida posible, permitiendo descubrir la relación entre las variables, específicamente entre una variable dependiente y las restantes.

2.3.1. Componentes del algoritmo de regresión simbólica.

Siendo un descendiente de la computación evolutiva, la regresión simbólica cuenta con cada una de las cinco características de esta práctica.

2.3.1.1. Una representación genética de las soluciones candidatas.

Como su nombre lo indica, el algoritmo de regresión simbólica no es más que una técnica de regresión. Ésta, sin embargo, propone representar cada posible modelo o expresión matemática mediante estructuras tipo árbol. Esta representación permite actuar en la búsqueda tanto de los coeficientes envueltos en la expresión como de la estructura o forma funcional de la expresión misma [Koza (1992)]. Es precisamente esta característica lo que diferencia este tipo de regresión de conceptos de regresión convencionales en los que se plantea un modelo con estructura fija y se buscan tan solo los valores de los coeficientes que generan un mayor grado de acople de los datos tratados.

La regresión lineal, por ejemplo, propone un modelo con estructura fija de la forma $y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n$ en el que, siendo y la variable dependiente y $\{x_1, x_2, \ldots, x_n\}$ el conjunto de variables independientes, se buscan los valores de los coeficientes $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$ que generen un valor de $\hat{y}$ inferida con la menor discrepancia posible del valor real.

La regresión simbólica, por su parte, propone iniciar la búsqueda con una población de soluciones candidatas codificadas en estructura de árbol como la mostrada en la figura 2.2 correspondiente a la expresión $y = \alpha_1 x_1 + \alpha_2 x_2 + cos(x_3)$. Estructuras de esta forma, compuestas por funciones, operaciones, o nodos internos (Ej. $+, -, \times, \div, sen(), cos(), exp()$) y terminales o nodos externos (Ej. valores constantes y variables $x_1, x_2 \ldots x_n$) consideradas por el diseñador, interactúan mediante la aplicación de operadores genéticos durante un número determinado de generaciones hasta obtener una expresión matemática codificada que pueda ser utilizada satisfactoriamente en la inferencia de valores de la variable dependiente.

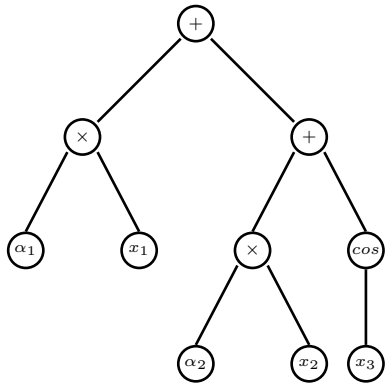


Figura 2.2: Ejemplo de codificación de expresiones matemáticas en forma de árbol.

2.3.1.2. Una forma de crear una población inicial de soluciones candidatas.

Uno de los pasos iniciales en el algoritmo de regresión simbólica, al igual que en cualquier técnica evolutiva, consiste en la construcción de una población inicial de soluciones o modelos candidatos que compondrán la generación cero. Para ello, en este algoritmo orientado

a procesos de modelado, debe iniciarse con la construcción de un número de estructuras en forma de árbol determinado por el diseñador (tamaño de la población), a partir de la combinación de elementos de los conjuntos de funciones y terminales considerados en el punto anterior. La elección de los elementos que compondrán estos conjuntos se convierte, por lo tanto, en un proceso de sumo cuidado en la tarea de diseño.

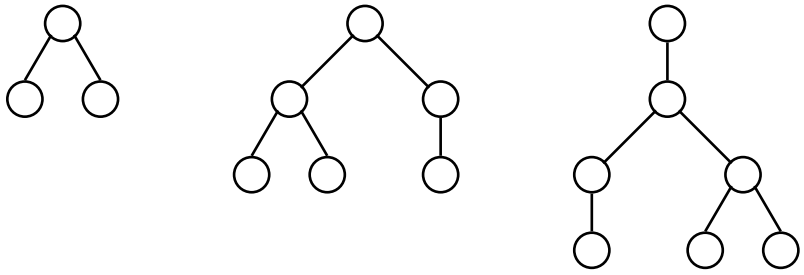
La construcción de una estructura que codifique expresiones matemáticas inicia con la selección, de forma aleatoria, de un elemento del conjunto de funciones. Este elemento constituirá la raíz del árbol. El número de ramas que desprenden de dicha raíz dependerá del número de argumentos que reciba la función seleccionada. Posteriormente, para cada rama es asignado un elemento, seleccionado de forma aleatoria, bien sea del conjunto de funciones o del conjunto de terminales. Si el elemento seleccionado es una función, el proceso de generación continúa de forma recursiva. Si, por el contrario, dicho elemento pertenece al conjunto terminal, allí se constituye el final de dicha rama.

El autor del algoritmo [Koza (1992)] hace distinción de tres tipos de procesos generativos que pueden ser implementados a la hora de construir las estructuras. Cada uno de ellos genera árboles con características, formas y tamaños diferentes.

- Método «full»: en el primero de ellos, llamado el método «full», la longitud de cada rama del árbol medida desde la raíz hasta cada punto terminal es idéntica e igual a un valor de profundidad máximo definido por el diseñador.

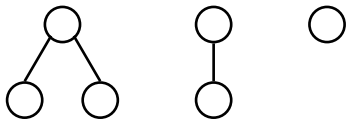
- Método «grow»: el segundo método es llamado el método «grow», consistente en la creación de árboles cuyas ramas no exceden un valor de profundidad máximo pero no todas las ramas son de igual longitud.

- Método «ramped half and half»: un tercer método de generación considerado es llamado por el autor método «ramped half and half». Éste involucra tanto el método «full» como el «grow». Consiste en la creación de un número idéntico de árboles considerando desde profundidad 2 hasta el valor máximo permitido. Es decir, para una población de 100 individuos y una profundidad máxima de 5 niveles, 25 individuos

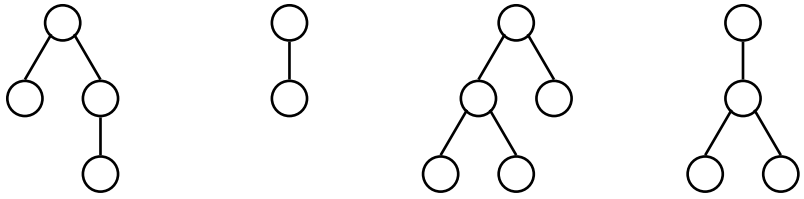


Longitud máxima: 2 Longitud máxima: 3 Longitud máxima: 4

Figura 2.3: Ejemplo de estructuras de árbol generadas mediante método «full».



Longitud máxima: 2



Longitud máxima: 3

Figura 2.4: Ejemplo de posibles estructuras de árbol resultantes mediante el método «grow».

tendrán longitud máxima 2, 25 longitud máxima 3, 25 profundidad máxima 4 y los 25 restantes profundidad máxima 5. Además, para cada conjunto el 50 % de los individuos serán construidos mediante el método «full» y el 50 % restante mediante el método «grow».

[Koza (1992)] resalta, además, la improductividad y por tanto la necesidad de evitar la

creación de individuos duplicados y la inclusión de individuos con alto nivel de aptitud de forma intencional pues estos podrían dominar sobre el resto de individuos arruinando por completo el proceso de exploración.

2.3.1.3. Una función de evaluación que permita calificar las soluciones de acuerdo con su grado de aptitud como solución.

El ideal en el proceso de inferencia es encontrar una expresión matemática que permita estimar el valor de una variable dependiente (y) a partir de datos de variables independientes ($\mathbf{x}$). Si se cuenta con un conjunto de l datos experimentales de estas variables $((\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_l}, y_l))$, es posible calificar el desempeño de una expresión en el proceso de estimación al evaluar la discrepancia entre el valor de la variable dependiente obtenido al evaluar en ella valores de las variables independientes ($\hat{y}$) y el valor real de la variable a inferir.

Esta calificación suele hacerse a partir del cálculo del error de aproximación o el cuadrado de este mismo ([Pereira *et al.* (2011)], [Liu (2007)]). Dicho error equivale a la sumatoria de la diferencia entre valores inferidos y valores reales de la variable de interés para un conjunto de l datos o casos considerados.

$$e = \sum_{i=1}^{l} y_i - \hat{y_i}$$

Un segundo paso considerado en el paradigma de programación evolutiva consiste, justamente, en la evaluación del desempeño o aptitud de cada programa, en este caso de cada estructura tipo árbol, para realizar la tarea planteada. Esta evaluación permite clasificar cada solución de acuerdo con su nivel de aptitud y dar preferencia en actividades genéticas a aquellas con mejores resultados, en el caso de mediciones inferenciales, aquellas expresiones matemáticas que, al ser evaluadas en valores de las variables independientes, generan un menor valor para el error de aproximación. Es precisamente ésta la razón por la cual la regresión simbólica se conoce como una técnica iterativa de regresión dirigida por el error de aproximación [Kumar *et al.* (2014)].

[Koza (1992)] define en su documento cuatro tipos de medidas de aptitud con posibilidad de consideración:

- «Raw fitness»: *«es la medida de aptitud obtenida a partir de la terminología natural del problema mismo»*. En el proceso de inferencia, este valor estaría dado por el error de aproximación, siendo éste la suma, o la suma al cuadrado, de la diferencia entre el valor predicho y el valor real de la variable dependiente para cada uno de los casos del conjunto de datos considerado.

- «Standardized fitness»: *«este valor reestructura el valor de aptitud bruto ("raw") de forma tal que un valor pequeño de aptitud sea siempre un mejor valor»*. El autor recomienda asignar cero al mejor valor de aptitud estandarizado. Considerando de nuevo el proceso de inferencia, en el que es plausible un valor bajo de error o aptitud bruto, el valor de la aptitud bruto y de la aptitud estandarizada son idénticos.

- «Adjusted fitness»: *«la medida de aptitud ajustada $a(i,t)$ para el i-ésimo individuo en la t-ésima generación es calculada a partir del valor de aptitud estandarizado $s(i,t)$ de la siguiente manera:»*

$$a(i,t) = \frac{1}{1 + s(i,t)}$$

«de forma tal que dicho valor fluctúa entre 0 *y* 1 *y es mayor para aquellos individuos con mejor desempeño dentro de la población»*.

De acuerdo con el autor, este tipo de puntuación enfatiza las pequeñas diferencias en el valor de aptitud estandarizado que permiten discernir entre una buena y una muy buena solución pues exagera la importancia de pequeñas diferencias en el valor de aptitud estandarizado cuando éste se acerca a su mejor valor.

- «Normalized fitness»: *«el valor de aptitud normalizado $n(i,t)$ para el i-ésimo individuo de una población de tamaño n en la t-ésima generación es calculado a partir del valor*

de aptitud ajustado de la siguiente manera:»

$$n(i,t) = \frac{a(i,t)}{\sum_{j=1}^{n} a(j,t)}$$

«Este valor se caracteriza por fluctuar entre 0 *y* 1*, ser mayor para aquellos individuos con mejor desempeño y generar un valor de suma de aptitudes de los individuos de una generación igual a la unidad.»*

2.3.1.4. Operadores que alteren la información genética contenida en las representaciones.

El paradigma de programación genética considera dos tipos de operadores genéticos básicos, reproducción o paso de un programa de una generación a otra sin alteración y cruce o recombinación de segmentos de dos programas seleccionados de acuerdo con su nivel de aptitud.

Sin embargo, la amplia descripción de los algoritmos evolutivos permite definir, a gusto del diseñador, el funcionamiento de estos operadores. Es por ello que diversos autores se han dado a la tarea de construir métodos para cumplir con los procesos de selección, reproducción y cruce contemplados dentro de los algoritmos evolutivos [Jebari & Madiafi (2011)].

- Operaciones de selección y reproducción.

  De acuerdo con [Koza (1992)], el método más popular de seleccionar, en este caso una estructura tipo árbol, para interactuar en operaciones genéticas es aquel, descrito por [Holland (1975)], proporcional a la calificación de desempeño. Haciendo uso de este tipo de selección, también conocida como selección proporcional por ruleta, en la operación de reproducción, la probabilidad de que un individuo i de la t-ésima generación sea mantenido en la generación futura, para una población de n individuos o estructuras, es calculada mediante la utilización del valor de aptitud $f(i,t)$ de la siguiente manera:

$$P(i,t) = \frac{f(i,t)}{\sum_{j=1}^{n} f(j,t)}$$

[Goldberg *et al.* (1991)] presentan, además, dos métodos de selección alternativos conocidos como selección por ranking y selección por torneo.

- Selección por ranking: trabajar con este tipo de selección supone la necesidad de organizar la población de acuerdo con el valor numérico que caracteriza sus desempeños. La medida de aptitud asignada dependerá solo de la posición de cada individuo dentro de la escala creada y no de la medida de desempeño misma.

  Considerando este esquema, la probabilidad de selección del i-ésimo individuo es calculada mediante la siguiente expresión, siendo n el tamaño de la población y $SP \in [1.0, 2.0]$ el valor de presión de selección equivalente a la probabilidad del mejor individuo de ser seleccionado comparado con la probabilidad promedio de selección de todos los individuos. El valor de ranking irá desde 1 para el individuo con peor desempeño hasta n para el mejor ejemplar [Pohlheim (2000)].

  $$P(i) = 2 - SP + 2 \times (SP - 1) \times \frac{rank(i) - 1}{n(n-1)}$$

  Este tipo de selección puede verse, al igual que la selección proporcional, como un proceso de sorteo por ruleta, en la que una porción de la misma es asignada a cada individuo. Al ejecutar el algoritmo la ruleta gira y aquellos individuos con mayor área asignada tienen una probabilidad mayor de ser elegidos. La repartición del área de la ruleta, sin embargo, difiere de un algoritmo a otro. Mientras que, por ejemplo, en la selección proporcional se asigna al mejor individuo una porción de acuerdo con su desempeño, en la selección por ranking se le asigna siempre la misma porción. Las figuras 2.5 y 2.6 muestran una posible distribución del área de la ruleta para ambos algoritmos.

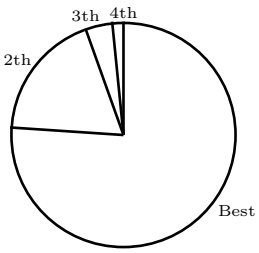


Figura 2.5: Ruleta segmentada por el método de selección proporcional.

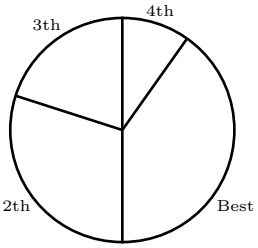


Figura 2.6: Ruleta segmentada por el método de selección por ranking.

- Selección por torneo: esta metodología considera, en principio, la selección aleatoria de un conjunto de k individuos, generalmente 2, que al ser organizados de acuerdo con su nivel de aptitud permiten la selección del individuo con mejor desempeño [Jebari & Madiafi (2011)].

Cabe mencionar que aquellos individuos que han sido seleccionados para contribuir en operaciones de reproducción y cruce continúan haciendo parte de la población actual y por tanto pueden ser seleccionados más de una vez (re-selección) para realizar estas tareas dentro de los procesos generativos en la generación actual.

La operación de selección puede ser realizada de forma independiente, sobre la población actual, cada vez que sea necesario escoger un individuo para participar en alguna operación. Sin embargo, también es posible, en un solo paso al inicio de la construcción de una nueva población, componer un «mating-pool» del tamaño de la misma constituido por individuos seleccionados probabilísticamente de acuerdo con su nivel de aptitud. Cuando un individuo es requerido para participar en operaciones genéticas, éste es tomado, en orden descendente, directamente del «mating-pool».

Además, aunque [Koza (1992)] no lo considera dentro de su algoritmo original, es posible contemplar una estrategia elitista que permita a un porcentaje reducido de los individuos con mejor calificación pasar de una generación a otra sin alteración.

- Operación de cruce.

La operación de cruce o recombinación sexual como es nombrada por el autor del algorítmo permite la creación de variaciones a partir de la combinación de segmentos de dos individuos padres seleccionados debido a su desempeño de características plausibles. El autor del algoritmo trabajado propone la selección de estos individuos mediante idéntico método al utilizado en la selección con fines reproductivos.

La operación de cruce considerada dentro del proceso de regresión simbólica inicia con la selección aleatoria (probabilidad de distribución uniforme) de un punto o nodo de cruce en cada uno de los padres. Los elementos contenidos en las ramas generadas a partir de dicho punto constituyen el fragmento de cruce de cada padre. El primer hijo es generado al reemplazar en uno de los padres su fragmento de cruce por el fragmento de cruce del segundo padre, mientras que el segundo hijo es generado al reemplazar, en el segundo padre, su fragmento de cruce por el fragmento de cruce del primer padre.

La longitud máxima de las estructuras de árbol generadas debe ser limitada para prevenir la generación de expresiones excesivamente complejas, por lo tanto, si el resultado de una operación de cruce genera una estructura que sobrepasa una longitud límite establecida por el diseñador, la operación es abortada y el primero de los padres es escogido arbitrariamente para hacer parte de la nueva población. Si ambas operaciones de cruce son interrumpidas, ambos padres pasarán a hacer parte de la generación futura.

Es aquí donde la forma de representación de la información dada por la programación genética genera una ventaja frente a los algoritmos genéticos convencionales. Debido a procesos de reproducción y cruce incestuosos (de un individuo consigo mismo), en algoritmos genéticos es posible que una solución mediocre pero ligeramente más apta que las demás termine dominando a la población pues el resultado de dichas operaciones será, a su vez, hijos totalmente idénticos al padre. En el algoritmo de regresión simbólica y en general en aquellos pertenecientes al paradigma de computación evolu-

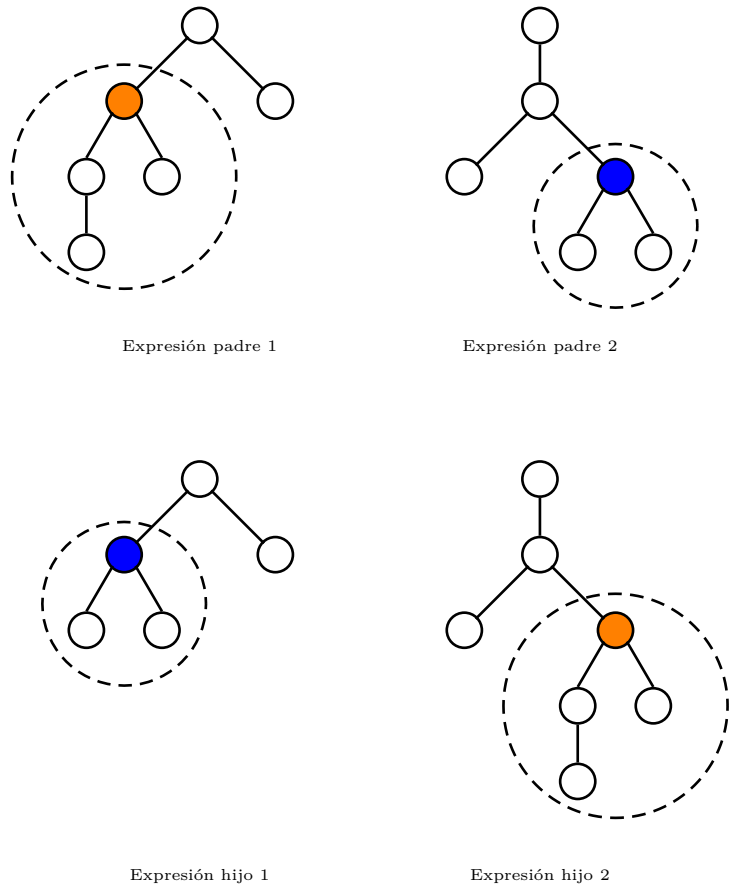


Figura 2.7: Ejemplo de operación de cruce de dos estructuras padre.

tiva, gracias a la complejidad de la estructura que codifica la información perteneciente a cada individuo y a la posibilidad de seleccionar un nodo diferente aunque se trate del mismo padre, los hijos generados de un cruce incestuoso no son necesariamente idénticos a su antecesor y por tanto se reduce el riesgo de llegar a una convergencia prematura [Koza (1992)].

- Operaciones complementarias.

Como se menciona en párrafos anteriores, los algoritmos originales de programación genética consideran tan solo operaciones de reproducción y cruce. Sin embargo, en su

documento, [Koza (1992)] menciona la posibilidad de incluir cinco operaciones complementarias que, él considera, no generan resultados diferenciadores en el proceso:

- Mutación: es un operador asexual que introduce cambios aleatorios en la estructura de una sola expresión original. Esta operación inicia con la selección aleatoria de un punto o nodo de la estructura manipulada para remover el segmento de él desprendido y reemplazarlo por una nueva estructura de árbol generada aleatoriamente (ver figura 2.8). Al igual que en el proceso de cruce, un valor de longitud máxima, generalmente equivalente a la longitud máxima de los individuos de la población inicial, limita el proceso de mutación.

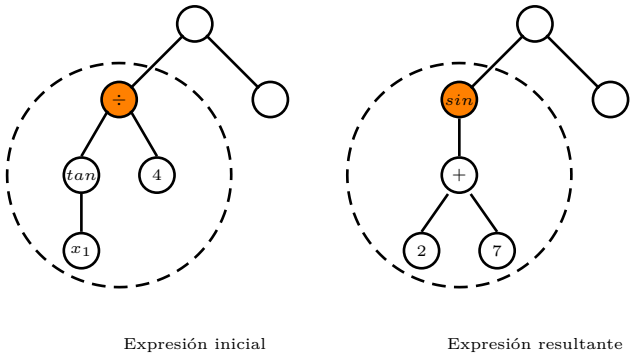


Figura 2.8: Ejemplo de operación de mutación de una estructura.

- Permutación: es una operación asexual pues involucra a un solo individuo o expresión seleccionado mediante método idéntico al utilizado en operaciones de reproducción y cruce. Esta operación inicia con la selección aleatoria de un nodo interno de la estructura (una función). De acuerdo con el número de argumentos que reciba dicha función se genera un conjunto de posibles permutaciones de las cuales una es seleccionada de forma aleatoria para ser ejecutada. Si la función seleccionada es conmutativa no se genera efecto alguno en la evaluación de la expresión.

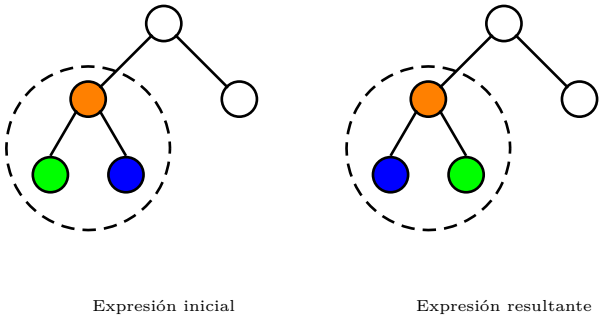


Figura 2.9: Ejemplo de operación de permutación de una estructura.

- Edición: operación asexual que considera que, si cierta función tiene solo valores constantes como argumentos, la expresión puede ser evaluada y reemplazada por el resultado de la evaluación (ver figura 2.10). Al realizarse, esta operación se ejecuta para cada uno de los individuos de la población simultáneamente y es controlada por un parámetro de frecuencia generacional determinado por el diseñador.

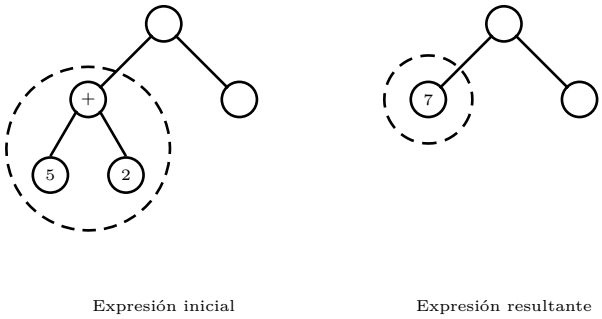


Figura 2.10: Ejemplo de operación de edición de una estructura.

- Encapsulación: operación consistente en la identificación de forma automática de un segmento de árbol potencialmente útil que pueda ser referenciado posteriormente mediante la asignación de un nombre específico (ver figura 2.11). El

individuo o expresión a manipular es seleccionado mediante método idéntico al utilizado en los procesos de reproducción y cruce. Un punto o nodo interno (función) es posteriormente elegido de forma aleatoria. El segmento desprendido de dicho nodo es removido y reemplazado por la llamada a una función que referencia a dicho segmento. El conjunto de funciones considerado es entonces agrandado al incluir la función creada.

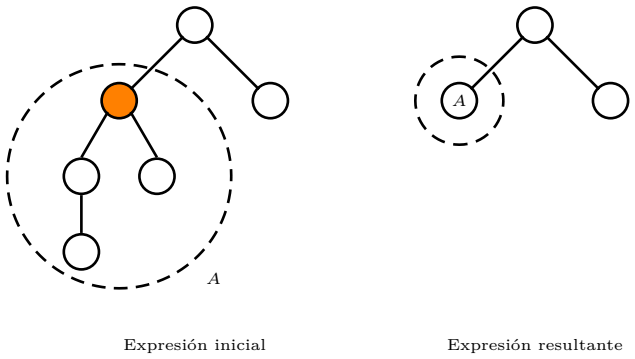


Figura 2.11: Ejemplo de operación de encapsulación de una estructura.

- Diezmado: esta operación es realizada, generalmente, después de construida y evaluada la población inicial. Es controlada por un parámetro que especifica el porcentaje afectado de la población. Consiste en la eliminación de cierta cantidad de individuos de acuerdo con el cálculo de los valores de aptitud. Su implementación se justifica en situaciones en las que la poblacion inicial contiene un alto número de integrantes con muy pobre desempeño debido a la complejidad del problema tratado. Eliminar algunos de estos integrantes reduce el riesgo de dominancia de aquellos individuos con desempeño superior [Koza (1992)].

2.3.1.5. Valores para ciertos parámetros como tamaño de la población, probabilidades de acción de cada operador etc.

Un proceso fundamental previo al inicio del algoritmo es la configuración de parámetros. Este algoritmo en particular cuenta con los siguientes parámetros básicos:

- Tamaño de la población.
- Número de generaciones.
- Probabilidades de cruce y reproducción.
- Longitudes máximas permitidas de las estructuras.
- Probabilidades y frecuencia de ocurrencia de los operadores complementarios.
- Método generativo de las estructuras de árbol, siendo «ramped half and half» el preferido por el autor.
- Método de selección de candidatos para operaciones genéticas siendo el proporcional el considerado en el algoritmo simple.
- Tipo de medida del nivel de aptitud a utilizar, siendo la aptitud ajustada la utilizada en el algoritmo simple.

Debe, además, determinarse qué elementos compondrán los conjuntos de funciones y terminales involucrados en la construcción de las estructuras. La selección de dichos elementos depende, en gran medida, de la naturaleza del problema a abarcar al igual que del conocimiento que el diseñador tenga, y desee incorporar, del sistema a modelar. Con frecuencia en la industria, la relación entre variables presenta comportamientos no lineales ([Sharma & Tambe (2014)], [Fan & Xu (2007)], [Zhang *et al.* (2004)]) y por tanto puede ser beneficioso considerar operaciones, dentro del conjunto de funciones, que inserten no linealidades .

[Koza (1992)] resalta en su documento, además, la importancia de crear o trabajar con conjuntos de funciones y operaciones que cumplan con condiciones de «cierre» (funciones bien

definidas ante cualquier combinación de argumentos que puedan recibir) y «suficiencia» (funciones y terminales capaces de expresar la solución al problema contemplado).

Respecto al conjunto de terminales, el problema de inferencia supone que éste sea compuesto por las variables independientes consideradas junto con un posible rango de valores constantes.

2.3.2. Descripción del algoritmo de regresión simbólica.

En síntesis, el flujo de actividades a desarrollar dentro del algoritmo de regresión simbólica original es el siguiente:

1. Configurar y definir parámetros involucrados.

2. Construir una población inicial de estructuras tipo árbol de acuerdo con el método seleccionado y los elementos contemplados en los conjuntos de funciones y terminales.

3. Realizar de forma iterativa hasta alcanzar en número máximo de generaciones configuradas o hasta cumplir con algún otro criterio de terminación impuesto por el diseñador:

   a) Evaluar los datos experimentales en cada expresión matemática y hallar el error de aproximación que permita asignar valores de desempeño y organizar la población de acuerdo con éste.

   b) Aplicar operadores genéticos sobre individuos de la población, eligiendo expresiones mediante la técnica de selección escogida (proporcional al valor de aptitud para el caso del algoritmo simple).

4. Designar la solución o modelo obtenido.

Luego de realizado el proceso iterativo hasta alcanzar un criterio de terminación determinado por el diseñador, consistente generalmente en un número máximo de generaciones operadas, la selección de la solución obtenida es realizada mediante una de dos posibles opciones:

- El mejor hasta ahora: en este método se elige el individuo, expresión o árbol cuyo resultado de la evaluación de los datos experimentales de las variables tratadas haya generado el menor error de aproximación dentro de todas las generaciones consideradas. Este tipo de selección supone la necesidad de guardar el mejor individuo en cada generación.

- El mejor de la última generación: esta metodología propone como solución final al mejor individuo de la última generación.

De acuerdo con [Koza (1992)] el resultado al utilizar ambos métodos es generalmente el mismo pues el mejor individuo suele estar siempre presente en la última generación pues pudo haber sido generado en generaciones precedentes pero debido a su alto desempeño es copiado mediante reproducción hasta la última generación o efectivamente fue creado en la última generación.

Es posible considerar, además, una tercera estrategia en la que un conjunto de individuos seleccionados de forma proporcional a su valor de aptitud es considerado para ser la solución al problema. En este caso el resultado de la inferencia resultaría del promedio de los resultados arrojados por dichos individuos.

2.4. Evaluación de impacto de los parámetros involucrados en el proceso de regresión simbólica.

El algoritmo de regresión simbólica cuenta con una cantidad considerable de parámetros a sintonizar de acuerdo con el problema enfrentado. El autor en su documento [Koza (1992)], presenta ciertas preferencias y recomendaciones. En los siguientes apartados, se describe, de forma general, la influencia de cada uno de los parámetros considerados, teniendo en cuenta tanto la experiencia de interacción con el algoritmo como los comentarios del autor y de otros investigadores, entre ellos, [Noraini & Geraghty (2011)], [Julstrom (1999)], [Blickle & Thiele (1995)] y [Goldberg *et al.* (1991)].

La sintonización de estos parámetros suele ser experimental y particular para cada problema enfrentado y es por esto que, a pesar de las recomendaciones, no existen indicaciones exactas que aseguren obtener un resultado provechoso.

2.4.1. Tamaño de la población.

El valor correspondiente a este parámetro se encuentra estrechamente relacionado con el área de la zona o espacio de búsqueda abarcado. Un número grande significa una mayor densidad de expresiones cubriendo dicha zona y por tanto una exploración del espacio con mayor detalle. [Koza (1992)], menciona que si se contara con una población lo suficientemente grande, la solución al problema tratado podría ser encontrada mediante una búsqueda aleatoria a ciegas en la primera generación. Sin embargo, contemplar un número considerable de expresiones o individuos en la población inicial implica un alto consumo, tanto computacional como temporal al momento de evaluar el desempeño de cada uno de ellos y entregar una calificación. De acuerdo con el autor, una población de máximo 500 individuos es suficiente en dos tercios de los casos.

2.4.2. Número de generaciones.

Al igual que el tamaño de la población, el número de generaciones es dependiente de la complejidad del problema tratado. Trabajar con un número bajo de iteraciones puede evi-

tar que se dé lugar a un proceso de convergencia a una solución satisfactoria pues es este número, generalmente, el criterio de terminación considerado. Por el contrario, un número excesivamente alto puede terminar en gasto de recursos computacionales innecesarios.

La figura 2.12 muestra, por ejemplo, cómo el algoritmo se desenvuelve rápidamente en encontrar solución a un sistema simple de una entrada (x) y una salida y, tal que $y = x^4 \times x^3 \times x^2 \times x$. Con una población inicial de 100 individuos e iterando durante 51 generaciones. El algoritmo llega a la solución esperada en la quinta generación, por lo que las restantes conforman procesamiento extra e innecesario en el que es posible que se construyan expresiones que, aunque equivalentes, posean estructuras significativamente complejas respecto a las encontradas en generaciones iniciales.

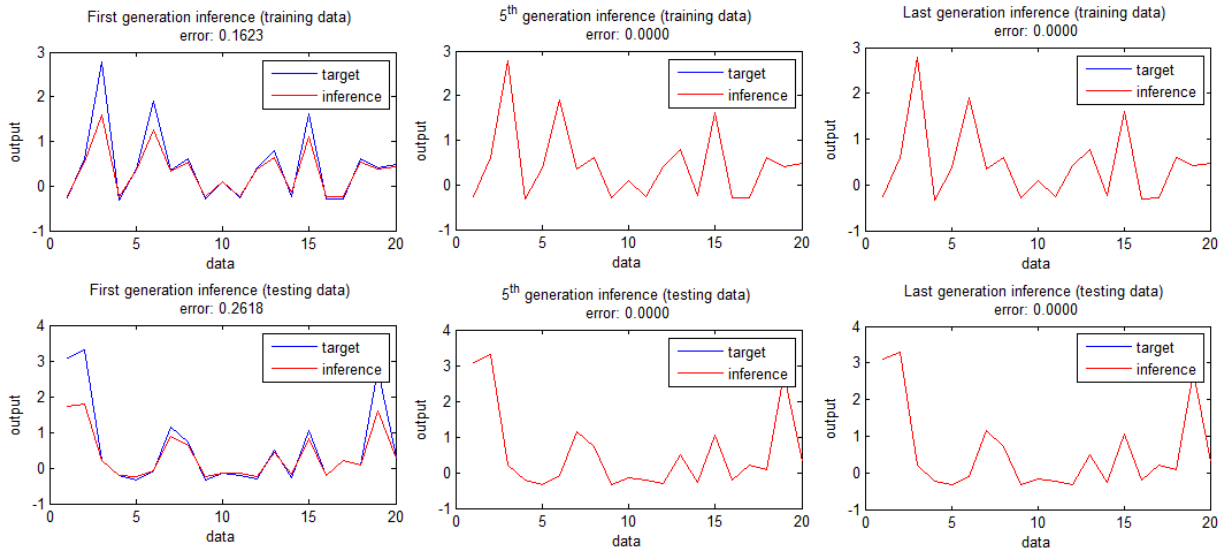


Figura 2.12: Evolución del proceso de búsqueda de la expresión $y = x^4 \times x^3 \times x^2 \times x$.

2.4.3. Método generativo de las estructuras de árbol.

La creación de la población inicial suele ser un proceso aleatorio en el que se construyen tantas estructuras tipo árbol diferentes como lo determine el tamaño de la población considerado por el diseñador. No se aconseja insertar individuos o expresiones con buena medida de desempeño, conocida a priori, dentro de la población inicial. Hacerlo puede terminar en la dominancia de esta expresión sobre las demás y estropear el proceso de búsqueda hacia

otros horizontes. El autor aconseja, en caso de querer incluir conocimiento experto en el proceso de búsqueda, que la totalidad (100 %) de los individuos sean generados con desempeño similar a partir del conocimiento que se tiene del sistema y no de forma aleatoria como se hace de forma convencional.

«Full», «grow» y «ramped half-and-half » son los tres métodos propuestos por [Koza (1992)] para generar las estructuras de árbol que codifican los modelos matemáticos candidatos en el proceso de inferencia. Para árboles creados mediante el método «full» con una longitud máxima establecida, todos los caminos desde la raíz hasta los terminales tienen la misma longitud y por tanto los árboles de una población contienen la misma forma. Por el contrario, árboles creados mediante el método «grow» generan una población diversa en estructuras. Si se relaciona la diversidad con el grado de exploración del espacio de búsqueda, es preferible trabajar con poblaciones con estructuras variadas. Es, tal vez, ésta la razón por la cual el autor resalta las características del método «ramped half-and-half» que crea árboles con una amplia variedad de tamaños y formas.

2.4.4. Operadores a considerar en el conjunto de funciones.

De acuerdo con el autor del algoritmo [Koza (1992)], el conjunto de funciones suele estar compuesto, al menos, por las cuatro operaciones aritméticas ordinarias. En el proceso de inferencia, sin embargo, suele ser necesario la inclusión de otro tipo de operadores que ayuden a describir las no linealidades que, con frecuencia, hacen presencia en los sistemas. La elección de este conjunto debe hacerse a partir de pruebas sobre los datos experimentales considerados. Podría creerse apropiado incluir el mayor número posible de operadores, sin embargo, es posible que esto resulte en estructuras complejas que no aseguren un desempeño significativamente mejor. Además, un conjunto de funciones grande debe reflejarse en una población grande que permita contener el mayor número de combinaciones posibles. Esto se traduce en complejidad y consumo de recursos computacionales.

2.4.5. Longitudes máximas permitidas de las estructuras.

El ideal, al final del proceso de búsqueda de una expresión que permita realizar inferencia, es obtener una estructura que genere el menor error de aproximación posible pero que a su

vez sea relativamente simple. Estos pueden considerarse dos objetivos opuestos pues es razonable pensar que una estructura compleja podrá contener mayor número de combinaciones de funciones y terminales que lleven a un mejor ajuste de los datos experimentales considerados. Sin embargo, en situaciones en las que se permiten grandes longitudes, el algoritmo podría terminar generando expresiones considerablemente complejas sin mejora significativa en precisión respecto a estructuras más simples.

Por el contrario, restringir la longitud de las expresiones a un valor demasiado bajo puede provocar dificultad en la creación de una población inicial sin individuos repetidos.

2.4.6. Probabilidades y frecuencia de ocurrencia de los operadores genéticos.

2.4.6.1. Operación de reproducción.

El operador de reproducción permite a un porcentaje de los individuos de la población actual hacer parte de la población futura sin modificación alguna. La selección de estos individuos típicamente responde a aquellos con desempeño sobresaliente aunque no de forma imperativa pues el método de selección suele ser incluyente con todos los candidatos.

El proceso de reproducción admite que un individuo sea seleccionado y tenido en cuenta para la formación de una nueva población más de una vez. Este hecho puede generar que luego de un proceso evolutivo en el que se considera una probabilidad alta de reproducción, un alto porcentaje de individuos de la población futura sean individuos repetidos pertenecientes a la población actual. Lo anterior lleva a pensar en la posibilidad de generar un proceso de convergencia prematura en el que debido a la ausencia, o poca presencia, de mecanismos que involucren diversidad (operadores de cruce, mutación, ... etc.) la exploración del espacio de búsqueda sea superficial, regida solo por los individuos pertenecientes a la población inicial.

La figura 2.13, muestra el resultado de promediar, por 30 experimentaciones, el número de individuos diferentes en cada una de 100 generaciones o iteraciones del algoritmo de regresión simbólica original considerando una población de 50 individuos y diversos valores de

probabilidad de reproducción y cruce. Es notorio que al establecer una probabilidad alta de reproducción el algoritmo converge rápidamente. Esta convergencia prematura se ve reflejada en los resultados de error de aproximación por generación mostrados en la figura 2.14. En el caso en el que se considera probabilidad alta de reproducción, la taza de reducción del error es menor que en aquel en el que se da algo más de preferencia al operador de cruce.

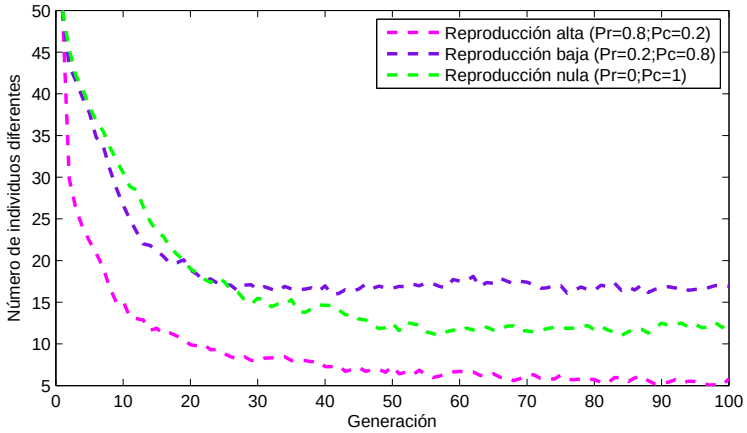


Figura 2.13: Evolución poblacional generada al contemplar diversas probabilidades de ocurrencia de operadores genéticos.

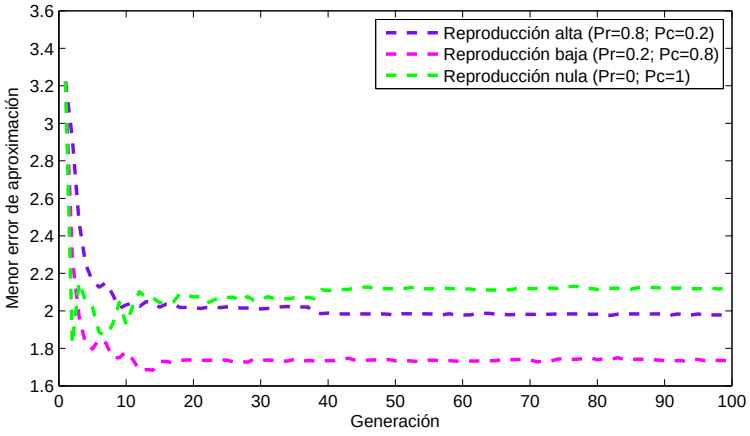


Figura 2.14: Evolución del error de aproximación generada al contemplar diversas probabilidades de ocurrencia de operadores genéticos.

2.4.6.2. Operación de cruce.

La función de la operación de cruce consiste en crear nuevos individuos prometedores dentro del espacio de búsqueda. Este operador se encarga de fomentar la exploración y mantener la diversidad de la población. A pesar de ser el ideal crear individuos con mejor desempeño a partir de la recombinación genética de individuos existentes, en algunos casos el operador de cruce puede ser el culpable de la degradación de algunas buenas expresiones.

[Koza (1992)], describe en su documento las situaciones que pueden presentarse en el momento de cruzar dos estructuras de árbol a partir de un nodo, en cada una de ellas, elegido de forma aleatoria:

- Si en uno de los progenitores el nodo elegido es un nodo terminal (e.j. variable o constante) éste es removido y reemplazado por un segmento de árbol proveniente del otro progenitor. Esta situación suele dar lugar a una expresión de longitud considerable.
- Si en ambos progenitores el nodo elegido es un nodo terminal, estos son simplemente canjeados y por tanto, la longitud de los individuos nuevos es idéntica a la de sus padres.
- Si en uno de los progenitores, la raíz es el nodo elegido, la expresión entera será insertada en el punto o nodo de cruce del segundo progenitor dando como resultado, probablemente, una expresión de longitud considerable. El segmento de árbol del segundo progenitor se convertirá en el segundo nuevo individuo creado.
- Si en ambos progenitores la raíz es el nodo elegido para realizar el cruce, el resultado son dos individuos totalmente idénticos a sus padres.
- Cuando un individuo es cruzado consigo mismo o dos individuos idénticos son cruzados, las dos estructuras o individuos generados serán, generalmente, diferentes entre sí y diferentes de sus progenitores. Lo anterior debido a la posibilidad de realizar el cruce a partir de nodos diferentes en ambos padres.

Esta última característica es sumamente importante y distintiva en la programación genética pues no siendo imperativo que al ocurrir un cruce incestuoso se generen individuos idénticos, se tiende a conservar la diversidad y, por tanto, a continuar con la promoción del

proceso exploratorio.

Diversos autores aconsejan trabajar con una probabilidad de cruce mayor respecto a la probabilidad de reproducción [Noraini & Geraghty (2011)]. Esto debido, principalmente, a la función de promoción de la exploración de dicho operador. Sin embargo, omitir por completo la operación de reproducción para dar prioridad a la operación de cruce podría resultar en una búsqueda aleatoria en la que se combinan individuos esperando obtener individuos mejores pero sin un punto de referencia y comparación con individuos de poblaciones anteriores. Estas circunstancias son visibles, igualmente, en las figuras 2.13 y 2.14.

2.4.6.3. Operación de mutación.

La función principal del operador de mutación consiste en introducir diversidad especialmente en generaciones avanzadas en las que el proceso de convergencia (reducción del número de individuos diferentes dentro de la población) es mayor. Este operador podría ayudar a explorar áreas del espacio de búsqueda diferentes a las explotadas por los individuos de la población actual o a reintroducir o recrear información que, generación tras generación, se ha ido perdiendo o degenerando debido a la aplicación del resto de operadores. Sin embargo, el autor del algoritmo [Koza (1992)], considera esta operación como secundaria y sin efecto contundente sobre la búsqueda. Él argumenta que debido a la, generalmente, poca cantidad de funciones y terminales, contemplados en un problema de regresión, sería extraño que alguna de ellas desapareciera totalmente en el proceso de búsqueda.

Un segundo argumento propone que, en cierta medida, el operador de cruce involucra comportamientos de mutación pues al igual que en ésta, al cruzar dos individuos se está removiendo parte de ellos e introduciendo nueva información proveniente de su compañero.

A pesar de estos argumentos, la figura 2.15 muestra que, en efecto, la inclusión de este operador al proceso de manipulación de información genética podría traer beneficios reflejados en la creación de estructuras, a partir de la adición a éstas de unas nuevas, que presenten comportamientos mejores en el ajuste de los datos. La figura 2.15 ilustra la evolución del error de aproximación mínimo producido generación tras generación. El contexto consiste en una población de 50 individuos que interactúan durante 100 generaciones. Estos lo hacen,

inicialmente, con probabilidad nula de mutación, probabilidad de cruce de 0.8 y probabilidad de reproducción de 0.2. En un segundo caso se contempla probabilidad de mutación de 0.1 y la probabilidad de cruce se reduce a 0.7.

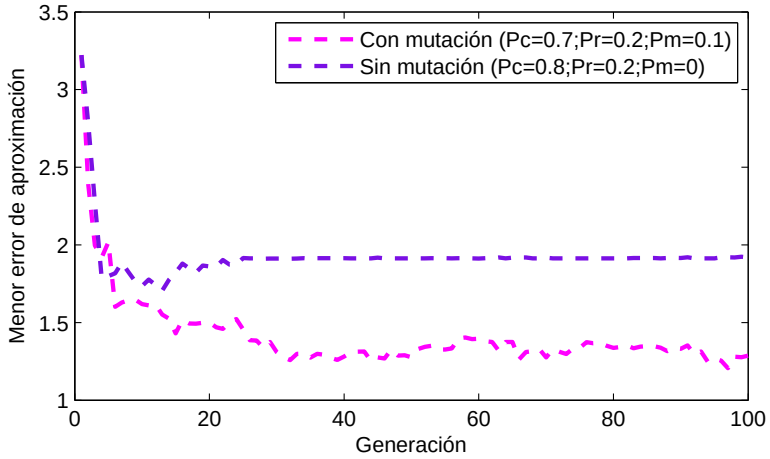


Figura 2.15: Evolución del error de aproximación generada al contemplar y no contemplar la operación de mutación.

En este contexto, en particular, es preciso notar el efecto de la mutación en la reducción del error de aproximación mínimo promedio.

Debe tenerse en cuenta, sin embargo, que permitir la acción del operador de mutación significa, también, exponerse a la probabilidad de degradación de buenas soluciones. El individuo, resultado de una mutación, al igual que el de ninguna de las operaciones contempladas en las técnicas evolutivas, es imperativamente mejor a su predecesor en términos de cercanía a la solución óptima. Probablemente es ésta la razón por la cual la probabilidad de mutación suele considerarse baja en algoritmos de computación evolutiva. Además, abusar de este operador, al igual que del operador de cruce, puede llevar al uso de la estrategia evolutiva como una simple búsqueda aleatoria.

2.4.6.4. Operación de permutación.

El punto clave a tener en cuenta al evaluar el efecto del operador de permutación consiste en analizar la propiedad conmutativa de las funciones y operaciones matemáticas contem-

pladas en el proceso de búsqueda pues dado el caso en el que la función contenida en el nodo seleccionado para ser permutado responda a dicha propiedad, no se producirá efecto inmediato en el valor producido por la expresión matemática contenida en la estructura resultante.

El autor considera este operador una técnica secundaria de manipulación de la información genética y sin efecto contundente sobre el desempeño del algoritmo de búsqueda. La figura 2.16 muestra el efecto en la evolución generacional del error de aproximación al incluir la acción del operador de permutación. El contexto contemplado es idéntico al creado para analizar el efecto de la mutación. Las funciones consideradas incluyen la operación de substracción que no responde a la propiedad conmutativa. En efecto, en esta experimentación en particular, no es notoria la acción, ni benéfica ni perjudicial, de la inclusión de este operador.

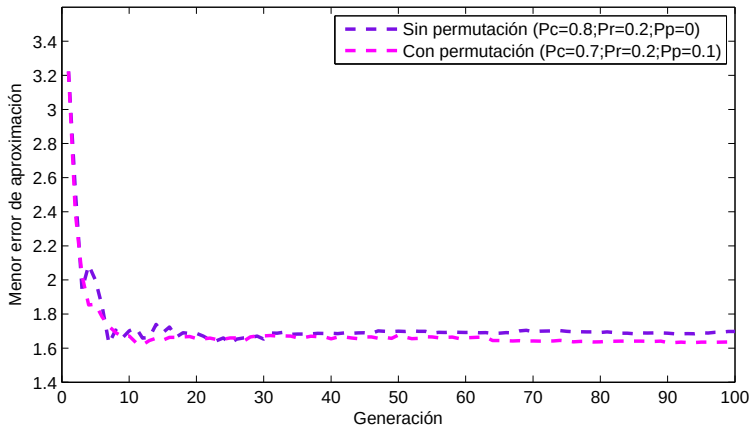


Figura 2.16: Evolución del error de aproximación generada al contemplar y no contemplar la operación de permutación.

2.4.6.5. Operación de encapsulamiento.

El encapsulamiento consiste en tomar cierto segmento de una estructura en particular y establecerlo como una nueva función a tener en cuenta en operaciones futuras. Al igual que la permutación, [Koza (1992)] considera ésta como una operación secundaria cuyo efecto no ha sido estrictamente definido. Sin embargo, su inclusión exige la disposición de memoria

que permita almacenar cada una de las nuevas expresiones convertidas en funciones. Aplicar esta operación con mucha frecuencia implica el crecimiento considerable del conjunto de funciones, hecho que de acuerdo con lo dicho en la sección *«Operadores a considerar en el conjunto de funciones»* no es necesariamente benéfico.

2.4.6.6. Operacion de edición.

El operador de edición se incluye como una operación secundaria que pretende simplificar las expresiones obtenidas cada cierto número de generaciones. Este operador toma sentido solo para aquellas expresiones que contengan nodos que actúan sobre valores de contantes que puedan ser evaluados y reducidos. La figura 2.17 pretende mostrar el efecto de la inclusión de este operador dentro de un proceso de búsqueda consistente en 50 individuos que interactúan genéticamente durante 100 generaciones. El parámetro de frecuencia de edición se configura para aplicar esta operación cada 10 generaciones. En la figura es posible notar una reducción significativa del número de nodos promedio de las poblaciones gracias a la inclusión del operador. Es de esperarse que este hecho se vea reflejado en soluciones finales con estructuras menos complejas.

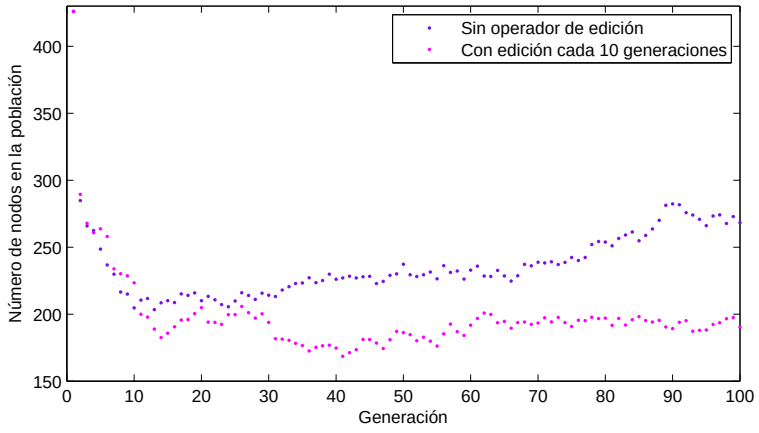


Figura 2.17: Evolución de la complejidad de las estructuras implicadas en el algoritmo utilizado y no utilizando la operación de edición.

Debe considerarse, sin embargo, que la reducción de la complejidad de las estructuras puede alterar el curso normal de la búsqueda, pues se reduce el número de nodos en los que

los operadores genéticos como cruce y mutación pueden actuar. Debe tenerse en cuenta, además, que la aplicación de esta operación aumenta el tiempo de procesamiento y por tanto frecuencias altas no serían recomendables.

2.4.6.7. Operación de diezmado.

El diezmado es, igualmente, una operación secundaria pensada para problemas complejos con poblaciones iniciales considerablemente grandes [Koza (1992)]. Su pretensión es iniciar el algoritmo de búsqueda con una población con calificaciones homogéneas en la que un individuo no domine rápidamente sobre los demás debido al amplio bache entre su desempeño sobresaliente y el de aquellos con desempeño deficiente. Aplicar diezmado en poblaciones pequeñas lleva a despoblar el espacio de búsqueda debido a la reducción del número de individuos en la población inicial y complica sustancialmente el proceso de búsqueda.

2.4.7. Método de selección de candidatos para operaciones genéticas.

La operación de selección permite explorar el espacio de búsqueda en vista de escoger a los individuos que participarán en la construcción de una generación futura. Al aplicar esta operación se espera una fijación mayor en aquellas áreas en las que se ubican expresiones que generan un menor error de aproximación en el proceso de inferencia. Sin embargo, ninguna de las posibles estrategias utilizadas se concentra única y exclusivamente en el punto que alberga la expresión con mejor desempeño. Las tres metodologías de selección descritas consideran la posibilidad de elegir, con cierta probabilidad, a cualquiera de los individuos constituyentes de la población. Esto permite conservar la diversidad de la misma que a la vez se traduce en un mayor grado de exploración.

La implementación de una u otra de las metodologías de selección contempladas afectan el desarrollo del proceso de búsqueda, especialmente, en términos de complejidad de implementación y consumo de recursos computacionales, velocidad de convergencia de la población hacia una solución, área del espacio de búsqueda explorada y precisión de la solución hallada. Diversos autores, como [Noraini & Geraghty (2011)], [Blickle & Thiele (1995)] y

[Goldberg *et al.* (1991)], se han tomado la tarea de escribir acerca de características y funcionamientos propios de cada metodología de selección en técnicas evolutivas. [Julstrom (1999)], por ejemplo, compara la complejidad computacional de la implementación de la técnica de selección por ranking y aquella por torneo y concluye sobre la efectividad y rapidez del algoritmo de selección por torneo debido a la ausencia de procesos de organización.

Algunas de las características estudiadas por los autores citados son:

(a) *Velocidad de convergencia:* en un proceso de búsqueda suele quererse llegar de forma rápida a una solución. Sin embargo, una velocidad de convergencia extremadamente alta puede significar falencias o superficialidad en la exploración del espacio.

Para medir esta velocidad en un proceso de búsqueda evolutivo que aplica cierta técnica de selección, [Goldberg *et al.* (1991)], por ejemplo, proponen encontrar el tiempo o número de generaciones que le toma a un individuo de una población inicial dominar sobre los restantes cuando el método de selección es utilizado exclusivamente en el proceso de reproducción. Los autores llaman a esta medida «take-over».

Otra posible forma de estudiar el comportamiento de una metodología de selección en términos de convergencia consiste en analizar la variación poblacional después de cierto número de generaciones, esta vez considerando tanto el operador de reproducción como el de cruce. Podría pensarse que un algoritmo que al final del proceso recursivo genera una población con el 90 % de los individuos idénticos cuenta con mayor velocidad de convergencia que aquel que finaliza con tan solo el 50 % de ellos.

(b) *Exploración del espacio de búsqueda:* iniciar el proceso de búsqueda mediante una amplia cobertura del espacio (exploración) para ir limitando dicha área poco a poco hasta concentrar esfuerzos en un espacio menor (explotación) es el esperado de una técnica de indagación. Cuando al aplicar un método de selección sobre una población se genera una nueva población con un alto número de individuos idénticos, el espacio de búsqueda se limita bruscamente traduciéndose en posibles falencias en la etapa de exploración. Este fenómeno es definido por [Blickle & Thiele (1995)] como «diverstity loss». El porcentaje de pérdida de diversidad generado por un algoritmo en particular es calculado por ellos

como la proporción de los individuos que no son escogidos para pasar de una generación a otra tras aplicar un proceso de selección exclusivo para reproducción.

(c) *Calidad de la solución hallada:* el desempeño de un algoritmo de selección puede ser medido, también, a partir de la calidad de la solución hallada al final del proceso de búsqueda llevado a cabo bajo dicha metodología de discriminación. Más aún, los autores [Blickle & Thiele (1995)] definen una medida por ellos nombrada «selection intensity» equivalente a la diferencia entre el desempeño promedio de los individuos de una población y el desempeño promedio de los individuos de la población generada mediante la aplicación de la metodología de selección para fines reproductivos. Se espera que un buen algoritmo de selección produzca un mejoramiento considerable del desempeño de una generación a otra.

El cuadro 2.1 muestra los resultados obtenidos al analizar el comportamiento de cada uno de los tres algoritmos de selección contemplados (selección por torneo con 2 y 10 candidatos, selección por ranking con presión de selección 1.5 y 2 y selección proporcional). El contexto de la experimentación consiste en una población de 50 individuos, probabilidad de cruce de 0.8, probabilidad de reproducción de 0.2 y 100 generaciones o iteraciones del algoritmo.

	Proporcional	**Torneo candidatos = 2**	**Torneo candidatos = 10**	**Ranking presión = 1.5**	**Ranking presión = 2**
Porcentaje de individuos diferentes en la última generación (%)	42	42	17.6	48	13.87
Error de aproximación de la solución hallada	0.8872	1.2506	1.2729	1.2406	1.3823

Cuadro 2.1: Indicadores de velocidad de convergencia y calidad de la solución hallada

.

¿Qué indican los resultados mostrados en el cuadro 2.1?

- *Indicador de convergencia:* entre mayor sea el porcentaje de individuos diferentes en la última generación, menor es el grado de convergencia generado por el algoritmo.
- *Indicador de calidad:* el error de aproximación es la medida natural del problema de medición inferencial abarcado. Entre menor sea dicho error mejor es el desempeño de la solución hallada.

Bajo idéntico contexto, la figura 2.18 presenta un ejemplo de la evolución poblacional del algoritmo de regresión simbólica al aplicar cada una de las cinco estrategias de selección contempladas.

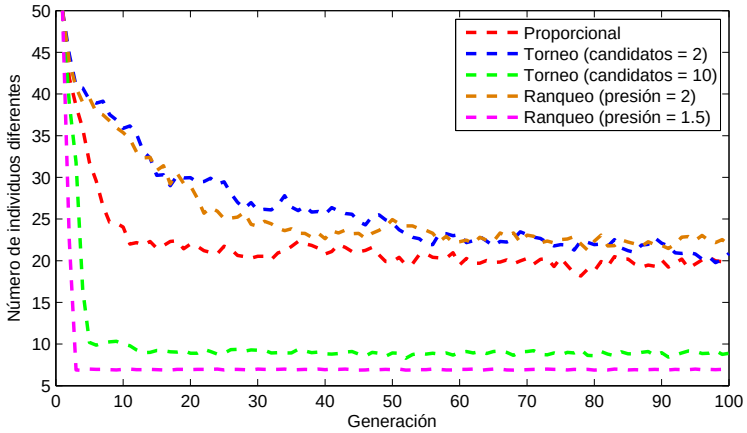


Figura 2.18: Evolución poblacional generada por cada estrategia de selección: número de individuos diferentes por generación.

En esta figura se aprecia la similitud en el comportamiento de la selección proporcional, selección por torneo con dos candidatos y selección por ranking con presión de selección igual a 2. Las tres técnicas, al pasar las 100 generaciones, reducen la búsqueda a cerca de la mitad de individuos considerados en la población inicial. Por su parte, la selección por torneo con diez candidatos y por ranking con presión de selección de 1.5 muestran una reducción brusca (en muy pocas generaciones) de la cantidad de individuos involucrados en la búsqueda. Esta última situación puede considerarse una convergencia prematura y un proceso de búsqueda ineficiente que termina en los altos valores de error de aproximación

consignados en el cuadro 2.1.

El cuadro 2.2, por su parte, muestra el resultado de experimentar sobre los algoritmos de selección, esta vez considerando tan solo una iteración y probabilidad unitaria para el operador de reproducción.

	Proporcional	**Torneo candidatos = 2**	**Torneo candidatos = 10**	**Ranking presión = 1.5**	**Ranking presión = 2**
«Take-over»	5.8	27.0333	1.9333	11.6333	1
«Density-loss» (%)	49.44	42.13	72.73	44.13	96
«Selection intensity»	4.8702×10^3	4.7973×10^3	4.8766×10^3	4.8001×10^3	4.8714×10^3

Cuadro 2.2: Indicadores de velocidad de convergencia, pérdida de diversidad y calidad de la solución hallada calculados en el paso de una sola generación considerando tan solo la aplicación del operador de reproducción.

¿Qué indican los resultados mostrados en el cuadro 2.2?

- *Indicadores de convergencia:* entre mayor sea el valor de «take-over», menor es la velocidad de convergencia del algoritmo.
- *Indicadores de diversidad:* el indicador de «diversity loss» toma valores entre cero y cien por ciento. Un valor mayor implica un mayor número de individuos de la población actual que no son contemplados en la formación de una población futura.
- *Indicadores de calidad:* «selection intensity» muestra qué tan eficiente es un algoritmo en construir una población con desempeño superior a la de sus antecesores. Un valor mayor en este índice significa un mayor grado de mejoramiento del desempeño promedio de la población manipulada.

¿Qué se espera de un buen algoritmo de selección?

A pesar de desearse un proceso de búsqueda que entregue una solución rápidamente, un valor alto del indicador de convergencia no puede considerarse una señal de buen desempeño, pues

puede ser resultado, por el contrario, de una exploración ineficiente. El ideal es obtener un equilibrio que asegure velocidad de convergencia, poca pérdida de diversidad y alta calidad de la solución hallada.

2.4.7.1. Selección proporcional.

Al aplicar esta técnica de selección se asigna a cada individuo un valor de probabilidad de ser elegido proporcional a su desempeño, lo que asegura que ningún elemento de la población es descartado. Así, el peor de los individuos, aunque con una probabilidad menor a la de sus compañeros, es candidato a ser seleccionado. Por esta razón es posible afirmar que con este método, en una población balanceada (con individuos cuyo desempeño no discrepa considerablemente uno del otro), se mantiene la diversidad y por tanto se explora convenientemente el espacio de búsqueda. Prueba de ello es el bajo índice de pérdida de diversidad consignado para esta estrategia de selección en el cuadro 2.2. Sin embargo, de acuerdo con [Noraini & Geraghty (2011)], es posible hallarse en situaciones en las que la población inicial contenga uno que otro individuo con un muy buen desempeño alejado del de los demás y en estas circunstancias es probable que se dé privilegio de selección a dichos individuos, generando una rápida e ineficiente convergencia.

Los resultados mostrados en los cuadros 2.1 y 2.2 muestran un buen equilibrio entre velocidad de convergencia y conservación de la diversidad para este algoritmo. El número de individuos diferentes en la centésima generación, generado por esta metodología, equivale a menos de la mitad de los individuos de la población inicial por lo que no es posible renegar sobre su capacidad de convergencia.

El método de selección proporcional es el favorecido por [Koza (1992)] en el planteamiento de su algoritmo. Es posible que esto se deba al equilibrio mencionado y a los resultados obtenidos para los índices de calidad de la solución hallada (superiores respecto a las técnicas restantes). En un proceso de inferencia, obtener un error de aproximación menor es mucho más significativo que llegar a la expresión que lo genera rápidamente.

2.4.7.2. Selección por torneo.

De acuerdo con [Noraini & Geraghty (2011)], este método de selección es el utilizado con mayor frecuencia en tareas desarrolladas mediante técnicas evolutivas. Esto, tal vez, debido a la baja complejidad de implementación ligada a la ausencia de procesos de organización de los individuos de acuerdo con su medida de desempeño. Al ser aleatoria la selección de los participantes en el torneo, esta técnica es incluyente con todos y cada uno de los individuos de la población promoviendo así la diversidad y evitando la dominancia de algunas expresiones. Sin embargo, esta conservación de la diversidad podría afectar, de acuerdo con [Noraini & Geraghty (2011)] la velocidad de convergencia. Esto puede verse reflejado en los resultados mostrados en el cuadro 2.2 para un torneo que considera dos candidatos. La pérdida de diversidad es baja respecto a la calculada para el resto de técnicas de selección. Sin embargo, la medida de «take-over» es considerablemente mayor respecto a la de los restantes.

El parámetro a considerar en este caso es el número de candidatos por torneo. Un mayor número de candidatos tiende a dar privilegio de escogencia a las mejores soluciones. Prueba de ello son los resultados mostrados en el cuadro 2.2 para el índice de pérdida de diversidad generado por el algoritmo de selección por torneo que considera diez candidatos. Este valor es considerablemente mayor al obtenido en la competencia entre tan solo dos pues existe mayor probabilidad de seleccionar consecutivamente al mejor de los individuos. Esta pérdida de diversidad se refleja, igualmente, en el bajo número de generaciones que le toma a un individuo, seleccionado mediante esta técnica, dominar sobre la población y puede traducirse en una exploración bastante trivial del espacio de búsqueda.

2.4.7.3. Selección por ranking.

De acuerdo con [Noraini & Geraghty (2011)], la selección basada en ranking puede sobreponerse a los problemas de convergencia prematura generados por una población inicial que contiene individuos altamente sobresalientes respecto a los demás, pues a diferencia de la selección proporcional, la probabilidad de escogencia de uno u otro individuo no depende de su desempeño sino de la ubicación obtenida al ser organizados de acuerdo con éste. Este proceso de organización, sin embargo, se traduce en términos de mayor costo computacional

y complejidad de implementación.

El parámetro influyente en este algoritmo es la presión de selección, valor contenido en el rango $[1, 2]$. Un valor de presión de selección cercano a uno da a los individuos con mejor desempeño una probabilidad mayor de selección. Ésta es la razón por la cual, en los cuadros 2.1 y 2.2 es notoria la alta velocidad de convergencia y pérdida de diversidad del algoritmo de selección por ranking que considera presión de selección igual a 1.5 respecto a aquel que considera dicho valor igual a 2. Al igual que en el proceso de selección por torneo con gran número de participantes, dar un amplio privilegio en la selección a los mejores individuos conlleva a la dominancia por parte de estos.

2.4.8. Estrategia elitista.

Aunque en la descripción del algoritmo original se menciona el concepto de estrategia elitista, ésta no es frecuentemente utilizada por su autor. Sin embargo, el permitir el paso de una pequeña porción de los mejores individuos generación tras generación, asegura que la expresión que conformará la solución al problema de búsqueda estará presente en la última generación, pues la expresión con mejor desempeño durante el proceso es rescatada y no corre el peligro de degenerarse debido a la aplicación de operadores genéticos sobre ella.

Claro está que considerar un alto porcentaje de individuos en la estrategia elitista estropea totalmente el proceso de búsqueda que se convertirá en una simple búsqueda aleatoria a partir de las expresiones generadas para constituir la población inicial.

2.4.9. Tipo de medida del nivel de aptitud a utilizar.

1. «Raw fitness»: esta medida de desempeño, en el contexto de medición inferencial, es equivalente al error de aproximación calculado ya sea como la suma, sobre todos los casos de la base de datos considerada, de la diferencia entre el valor estimado y el valor real de la variable a inferir o como la suma, sobre todos los casos de la base de datos considerada, de la diferencia al cuadrado entre el valor estimado y el valor real de la variable a inferir. Esta última forma de expresar la discrepancia entre el valor esperado y el valor obtenido enfatiza la influencia de divergencias mayores. Hacer uso de este

tipo de error permite castigar en mayor medida, asignando una baja calificación de desempeño, a aquellas expresiones que generan valores considerablemente más alejados de los valores esperados.

2. «Standarized fitness»: esta medida de aptitud pretende asignar un valor bajo a aquellas expresiones con mejor desempeño, siendo el ideal obtener una expresión que genere un valor de aptitud estandarizado igual a cero. Al ser el valor ideal del error de aproximación considerado en el proceso de medición inferencial igual a cero, la medida de aptitud estandarizada toma idéntico valor a la medida bruta o raza («raw») y por tanto no hay diferencia alguna en el uso de una u otra.

3. «Adjusted fitness»: este valor fluctúa entre cero y uno. El autor del algoritmo lo prefiere debido a su capacidad de exagerar la importancia de pequeñas diferencias en el cálculo de la medida estandarizada. Lo anterior permite diferenciar, con mayor facilidad, entre un buen individuo y uno muy bueno.

4. «Normalized fitness»: consiste en un concepto especialmente necesario cuando se considera selección proporcional o por ruleta. Esta forma de calcular el desempeño asegura que la suma de las calificaciones de la población entera sea igual a la unidad, hecho que facilita el cálculo de la probabilidad de selección.

La elección de uso de uno u otro tipo de medida de aptitud dependerá de las necesidades propias del momento. Algunos operadores genéticos, por ejemplo, hacen uso de un tipo de medida particular.

Capítulo 3

Modificación: variaciones contempladas por otros autores y propuesta para mejorar el desempeño del algoritmo de regresión simbólica original en el proceso de inferencia.

Debido a su diversidad en características, el algoritmo de regresión simbólica se ha prestado para que múltiples autores presenten sus modificaciones en búsqueda de mejores características de precisión, generalización, complejidad, autonomía, etc. El ideal es obtener una herramienta que halle un modelo matemático, relativamente simple, que se ajuste lo mejor posible a los datos experimentales u observaciones con los que es realizada la búsqueda, evitando caer en fenómenos de sobre-entrenamiento que reduzcan la capacidad de generalización del mismo. Las ideas de estos autores pueden llegar a convertirse en el primer escalón para proponer una nueva modificación que lleve al algoritmo a entregar mejores resultados.

3.1. Modificaciones implementadas sobre el algoritmo de regresión simbólica original.

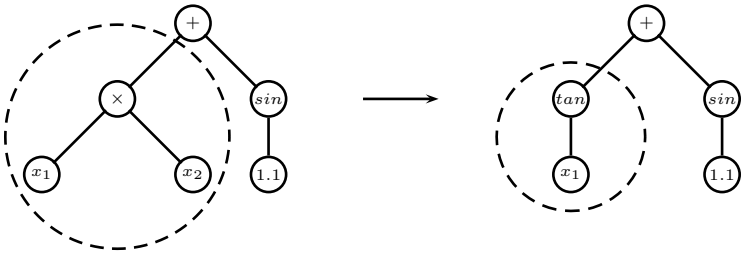
Si bien, el algoritmo de regresión simbólica original ha dado muestra de buen funcionamiento en la construcción de relaciones y modelos matemáticos ([Sharma & Tambe (2014)], [Bolshakov (2013)]), siempre es posible plantear hipótesis acerca de modificaciones que generen resultados con mejores características. Es así como surgen, por ejemplo, propuestas de modificación del funcionamiento de los operadores genéticos, de la forma de representar las soluciones potenciales y de la forma de construir la solución final, entre otras.

3.1.1. Modificación de operadores.

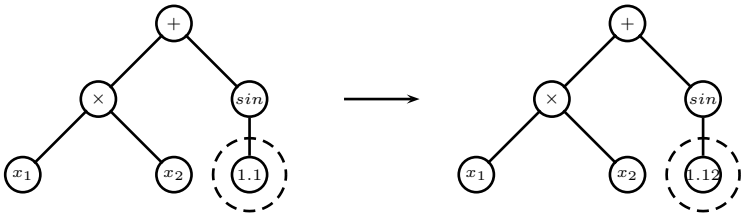
La versatilidad del algoritmo de regresión simbólica y la diversidad de características que contempla ha permitido a muchos interesados jugar con el funcionamiento de los operadores genéticos considerados en búsqueda de mejores comportamientos.

Conservando el sentido del operador de mutación, que pretende diversificar el proceso de búsqueda, autores como [Hii *et al.* (2011)], [Searson *et al.* (2010)] y [Lopes & Weinert (2004)] han propuesto, por ejemplo, incluir tipos de mutación adicionales al considerado en el algoritmo de regresión original. Mientras que en éste se considera tan solo la mutación que, en un individuo, remueve el sub-árbol que desprende a partir de un nodo elegido de forma aleatoria y lo reemplaza por otro, estos autores proponen mutaciones que consideran realizar leves variaciones a constantes, enviar constantes a cero o a la unidad, cambiar una variable por otra elegida al azar, etc. (ver figura 3.1).

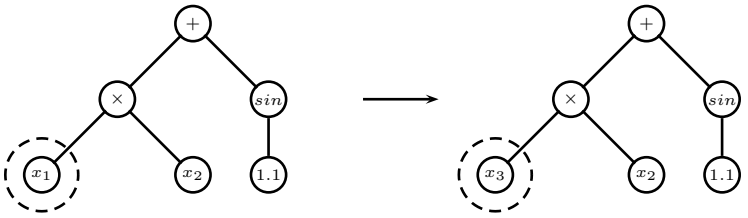
Variaciones en las operaciones de selección y cruce también se hacen presentes en trabajos como los de [Martínez *et al.* (2011)] y [Ferreira (2001)]) incluyendo, por ejemplo, conceptos en los que se da prevalencia en operaciones de selección a aquellos modelos que, dentro de una generación, se ajustan mejor a la mayor cantidad de instancias posibles de la base de datos.



a) Operación de mutación original: cambio aleatorio de un sub-árbol.



b) Operación de mutación: variación leve de una constante.



c) Operación de mutación: cambio de variable.

Figura 3.1: Ejemplo de operaciones de mutación consideradas por diversos autores.

3.1.2. Mejoras en precisión.

3.1.2.1. Sintonización de constantes.

Pretendiendo realizar una búsqueda local o un proceso de explotación del espacio, se han propuesto estrategias para sincronizar los valores de las constantes de un modelo en particular. El procedimiento descrito y trabajado por [Lopes & Weinert (2004)], por ejemplo, consiste en evaluar el desempeño de un individuo y conservarlo en la memoria para poste-

riormente comparar con la reevaluación al modificar, en cierto porcentaje, el valor de las constantes involucradas en la expresión. [Kommenda *et al.* (2013)], por su parte, hacen uso del algoritmo de Levenberg-Marquard, un algoritmo de optimización no lineal, para hallar los valores de constantes que hacen que un modelo hallado al final del proceso de búsqueda se ajuste en mayor medida a los datos experimentales. Aunque, de acuerdo con los autores, ambos procesos generan cierta mejora en el modelo final establecido, la inclusión de estas técnicas acarrean aumento en los costos computacionales del procedimiento.

3.1.2.2. Variaciones en la representación.

El algoritmo original de regresión simbólica propone representar cada posible modelo solución como una sola estructura de árbol cuyas hojas o nodos contienen funciones, operadores, variables y constantes que se conectan entre sí para dar sentido a una expresión matemática. Esta estructura es entendida en la mayoría de documentos como un «gen» debido a la analogía del proceso de regresión simbólica con procesos genéticos y de evolución.

Por su parte, autores como [Kumar *et al.* (2014)], [Hii *et al.* (2011)], [Searson *et al.* (2010)] y [Ferreira (2001)] se han aventurado a considerar en sus trabajos la construcción de estructuras compuestas por dos o más genes que, de forma análoga a la biología, constituyen un cromosoma. Esta propuesta nace con el objetivo de expandir el número de posibles combinaciones de operadores y operandos considerados en las expresiones y con la esperanza de obtener estimaciones más ajustadas a los datos. Los autores coinciden en afirmar que el algoritmo de regresión simbólica multigen, como es por ellos nombrado, puede ser más exacto y eficiente computacionalmente que el enfoque estándar.

Por su parte, [Lopes & Weinert (2004)] trabajan con la propuesta de representación hecha por [Ferreira (2001)]. Este enfoque propone codificar cada expresión mediante genes concatenados. Visualmente, este tipo de representación (ver figura 3.2) y el considerado en el algoritmo original son idénticos. Ambos son estructuras de árbol con una longitud determinada. Sin embargo, en la nueva codificación, sub-árboles de la estructura inicial son considerados como genes y por tanto nuevas operaciones pueden ser definidas entre ellos. [Lopes & Weinert (2004)], por ejemplo, incluyen en su aplicación operaciones de cruce a uno y dos puntos en los que el cromosoma o la estructura total es dividida en la raíz de uno o

dos genes que constituirán el límite de entrelazado.

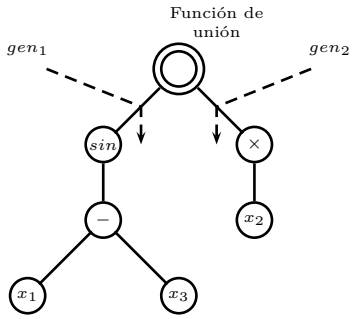
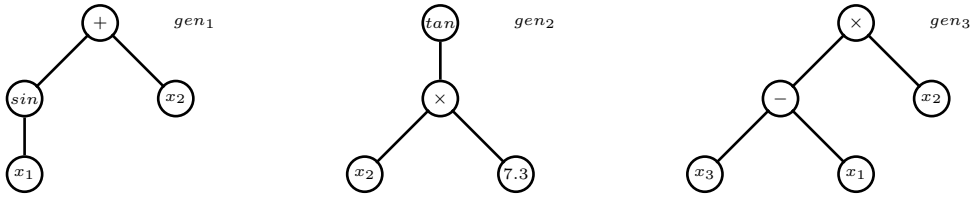


Figura 3.2: Ejemplo de representación multigen.

Otra posible concepción del algoritmo de regresión simbólica multigen, es el implementado en [Hii *et al.* (2011)] y [Searson *et al.* (2010)]. Estos autores trabajan con individuos compuestos por varias estructuras de árbol independientes. En su propuesta, cada modelo es una combinación lineal ponderada de las salidas provenientes de cierto número de estructuras de árbol convencionales (ver figura 3.3).



$$\hat{y} = \alpha_0 + \alpha_1 \times gen_1 + \alpha_2 \times gen_2 + \alpha_3 \times gen_3$$

$$\hat{y} = \alpha_0 + \alpha_1 \times (sin(x_1) + x_2) + \alpha_2 \times (tan(7.3 \times x_2)) + \alpha_3 \times (x_2 \times (x_3 - x_1))$$

Figura 3.3: Ejemplo de representación multigen en combinación lineal ponderada.

Al igual que en la representación multigen propuesta por [Ferreira (2001)], esta representación permite incluir operaciones genéticas adicionales al realizar combinaciones de genes

entre individuos (ver figura 3.4).

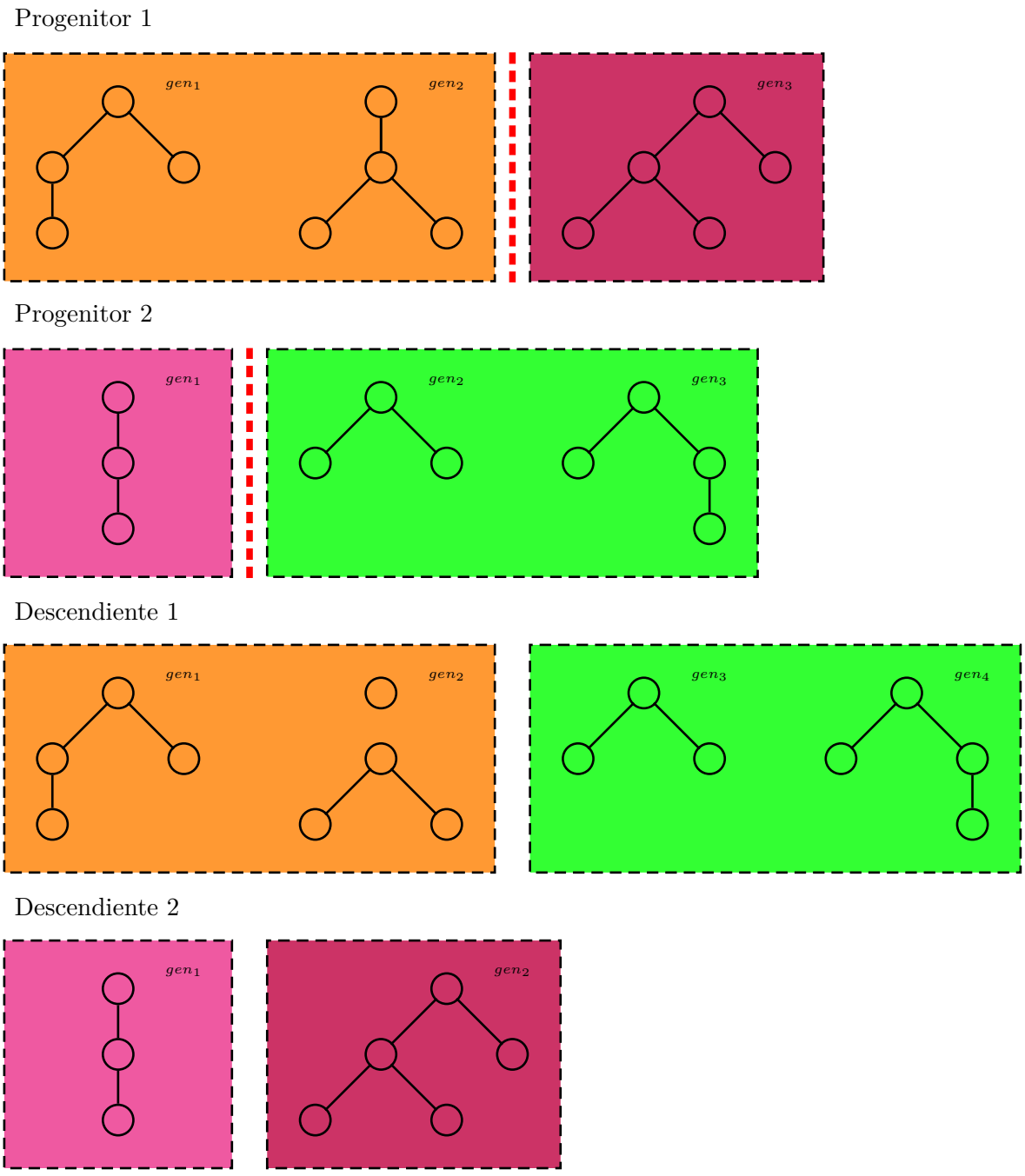


Figura 3.4: Ejemplo de operación de entrelazado de genes en representación multigen.

Los pesos o coeficientes lineales son estimados a partir de una base de datos de entrena-

miento mediante técnicas ordinarias de mínimos cuadrados. De acuerdo con los autores, esto permite combinar el poder de la regresión lineal clásica con la habilidad de capturar no linealidades sin la necesidad de suponer una estructura de modelo de la regresión simbólica.

Esta forma de representación permite la inclusión de mayor número de combinaciones de operadores y operandos en la expresión matemática sin necesidad de considerar estructuras con grandes longitudes. Aunque a primera vista, esta propuesta podría llevar a pensar en la obtención de estructuras mucho más complejas que aquellas entregadas por las representaciones convencionales, los autores afirman que si se restringe la longitud máxima de cada gen a 4 ó 5 niveles y se consideran individuos con un máximo de 5 genes, los modelos en evolución son relativamente compactos.

3.1.2.3. Construcción de una solución a partir de las salidas de varios modelos.

La propuesta de [Kordon *et al.* (2004)] para obtener una medida inferida confiable y con mayor precisión consiste en el cálculo de la media de los valores entregados por no uno sino un conjunto de modelos obtenidos al final del proceso de búsqueda. De acuerdo con el autor, hacer uso de una medida dada por un conjunto de predictores acarrea consigo numerosas ventajas, entre ellas la posibilidad de ser alertado cuando el sensor inferencial ha salido del rango de valores contemplado en el entrenamiento, pues las medidas de todos los modelos diferirán, en mayor medida, una de la otra. Otra ventaja resaltada del uso de un conjunto de modelos es la creación de redundancia. Si cada modelo del conjunto cuenta con entradas provenientes de sensores de variables diferentes y alguno de estos sensores llegara a fallar, aún quedarían otros modelos, que no cuenten con esa variable en su estructura, capaces de predecir correctamente.

3.1.3. Mejoras en generalización.

Otro punto importante a atacar, además de la precisión, es la capacidad de generalización de los modelos hallados. Es posible que al realizar un proceso de búsqueda, evaluando el desempeño de los individuos con un conjunto de datos determinado se llegue a una solución que se ajuste satisfactoriamente pero que al incluir datos de prueba no considerados en el

proceso de entrenamiento, el error de estimación se incremente de forma considerable. Este fenómeno es conocido como sobre-entrenamiento u «overfitting» y es atacado, generalmente, con la inclusión de métodos que suponen la necesidad de experimentar con la base de datos para obtener un modelo que se ajuste, en la mayor medida posible, a la totalidad de estos.

Investigadores interesados en el tema como [Gonçalves & Silva (2013)], [Martínez *et al.* (2011)] y [Gathercole & Ross (1994)], pregonan un mejor desempeño del algoritmo de regresión en casos en los que se hace uso de subconjuntos diferentes de la base de datos para evaluar el nivel de aptitud de los individuos de una población en cada generación y no de la base de datos completa. Este procedimiento es conocido como muestreo de la base de datos y quienes han indagado sobre el tema afirman mejoras no solo en temas de reducción de sobre-entrenamiento y aumento de la generalización sino también en disminución de gasto computacional y consecución de expresiones más compactas ([Gonçalves & Silva (2013)], [Martínez *et al.* (2011)]). Algunas técnicas propuestas consideran muestreo intercalado o «interleaved sampling» y muestreo intercalado aleatorio «random interleaved sampling».

3.1.3.1. Muestreo intercalado (Interleaved Sampling).

Descrito por [Gonçalves & Silva (2013)], este método propone hacer uso, en el cálculo del valor de aptitud, del conjunto completo de datos en ciertas generaciones y de tan solo ciertos de ellos en las restantes. [Gonçalves & Silva (2013)] propone tres variaciones. La primera de ellas llamada simplemente «intercalado» propone hacer uso del total de los ejemplos generación de por medio y de tan solo un ejemplar en las generaciones restantes. Una segunda variación llamada «intercalado único» pretende dar preferencia al uso de un solo elemento de la base de datos. Para ello debe considerarse la inclusión de un parámetro que, de forma porcentual, especifique por cada generación en la que se haga uso de la base de datos completa, cuántas generaciones serán evaluadas haciendo uso de un solo ejemplo. Finalmente, una tercera variante con fundamento opuesto llamada «intercalado total» pretende dar preferencia al uso del total de los datos de entrenamiento y para ello ha de definirse un parámetro que especifique, por cada generación en la que se haga uso de un solo ejemplar, en cuántas generaciones se hará el cálculo del valor de aptitud teniendo en cuenta la totalidad de los datos.

3.1.3.2. Muestreo intercalado aleatorio (Random Interleaved Sampling).

Descrito, de igual forma, por [Gonçalves & Silva (2013)], esta metodología propone decidir, probabilísticamente cuántas instancias de la base de datos serán utilizadas en la evaluación del desempeño en cada generación. Cada generación debe decidirse si se tomarán todas las instancias o solo una de ellas de acuerdo con un valor de probabilidad establecido para este último caso.

3.1.3.3. Algoritmo de búsqueda de novedad (Novelty Search Algorithm).

Este algoritmo propone dar prioridad en operaciones de selección a aquellos individuos que se desempeñen mejor en el ajuste de las instancias o ejemplos de la base de datos, es decir, a aquellos individuos que generen un menor error de aproximación en el máximo número de instancias posibles. De igual forma, se da prioridad en la evaluación de la función objetivo haciendo uso de las instancias más difíciles, es decir, aquellas que provocan mayores errores de estimación en el total de los individuos de la población. El algoritmo inicia con la asignación, a cada individuo, de un vector de longitud equivalente al número de instancias en la base de datos. Este vector binario es construido de forma tal que un 1 en la i-ésima posición significa que la solución se encuentra dentro del «top» de individuos con desempeño sobresaliente en el ajuste a los datos contenidos en esa instancia en particular. La construcción de estos vectores en cada generación permite identificar a los individuos con mejor desempeño general y los casos o instancias más complejas en las que se hará hincapié, en generaciones futuras, para evaluar el desempeño de las poblaciones [Martínez *et al.* (2011)].

3.1.3.4. Muestreo intercalado conservando al peor (Keep-Worst Interleaved Sampling).

Esta propuesta, hecha por [Martínez *et al.* (2011)] es basada en ideas de muestreo intercalado y el algoritmo de búsqueda de novedad. El autor propone hacer uso de la base de datos en su totalidad generaciones de por medio y en las restantes, hacer uso solo de cierto porcentaje de aquellas instancias que en la generación anterior hayan generado un mayor error de aproximación en el general de la población.

3.1.4. Mejoras en complejidad.

[Hii *et al.* (2011)], [Smits & Kordon (2008)] y [Kordon *et al.* (2004)], por su parte, han concentrado sus esfuerzos en el desarrollo de técnicas que aseguren la obtención de modelos o expresiones matemáticas poco complejas. Esto con el fin principal de facilitar su implementación. Lamentablemente, el algoritmo estándar u original tiende a producir modelos que contienen expresiones con muy poco o nulo efecto sobre la predicción final [Hii *et al.* (2011)]. Este fenómeno es conocido como «inflación» o «bloat» [Luke & Panait (2006)]. En este algoritmo, la complejidad de las estructuras es controlada, únicamente, mediante el parámetro de longitud máxima establecido por el diseñador. Limitar este número, sin embargo, supone también limitar el espacio de búsqueda. Es por esta razón que investigadores se han dado a la tarea de plantear y evaluar diferentes metodologías que aseguren una búsqueda correcta, control del fenómeno de inflación y resultados satisfactorios.

3.1.4.1. Penalidad en la función de desempeño.

En métodos heurísticos de optimización es frecuente introducir el concepto de penalidad en el que soluciones potenciales que violan algún tipo de restricción son castigadas disminuyendo su calificación de desempeño. La disminución de este valor puede mantener a estas soluciones al margen de procesos de selección que les permitan, ya sea, pasar de una generación a otra o contribuir en la construcción de una generación emergente. Esta noción de penalidad ha sido utilizada para intentar combatir la aparición de expresiones demasiado complejas en el proceso de búsqueda [Babu & Karthik (2007)]. En cada generación, la calificación de aquellos individuos que sobrepasan un límite de complejidad es manipulada para disminuir su probabilidad de participación en operaciones posteriores. Sin embargo, esta solución podría no ser lo suficientemente eficiente pues dado el caso en el que el total, o la mayor parte, de la población esté compuesta por estructuras complejas hace que el proceso de penalización pierda sentido.

3.1.4.2. Multiobjetivo con frentes de Pareto.

En documentos como [Smits & Kordon (2008)] y [Kordon *et al.* (2004)], se toma el proceso de inferencia o estimación de medidas como un problema de optimización multi-objetivo en el que se pretende obtener un modelo que genere el mínimo error de aproximación posible y que, a la vez, consista en una expresión matemática poco compleja. Estos autores miden la complejidad mediante el conteo del número de nodos o ramas que componen cada estructura de árbol y clasifican a los individuos de acuerdo con el concepto de dominancia de Pareto, naciente de las ciencias económicas y ampliamente utilizado en optimización multiobjetivo. En estos casos, en concreto, un modelo domina a otro cuando lo supera en desempeño o cuenta con menor nivel de complejidad. La clasificación de los modelos creados en una generación en frentes de Pareto que determinen qué individuos dominan a qué individuos, permite incluir técnicas de selección de expresiones, para participar en operaciones genéticas, que den prioridad a aquellas no dominadas por ninguna otra.

Por su parte, [Hii *et al.* (2011)] se aventura a incluir conceptos del algoritmo genético NSGA-II dentro del proceso de búsqueda. Estos autores proponen que al final de cada generación, tanto los individuos de la población actual como los de la generación anterior sean clasificados en frentes de Pareto y, de allí, ordenados de acuerdo a medidas de aglomeración. El ideal es incluir un proceso de selección de individuos que dé preferencia a aquellos ubicados en los primeros frentes y resaltar características de diversificación.

3.1.4.3. Variación en la función objetivo.

Otra de las estrategias que han sido estudiadas para enfrentarse a la aparición de modelos extremadamente complejos incurre en el reemplazo de la función objetivo original usada, el error medio de aproximación, por otro tipo de medida que penalice a aquellos modelos con estructuras engorrosas. [Martínez & Velásquez (2013)], por ejemplo, hacen uso del criterio de información de Akaike ([Akaike (1974)]), que tiene en cuenta el número de parámetros, además del nivel de ajuste del modelo a los datos experimentales, para asignar una calificación.

3.1.4.4. Otras técnicas de control de inflación (Bloat control techniques).

Otras técnicas que pretenden contrarrestar la creación de estructuras complejas contemplan, por ejemplo, la inclusión del algoritmo de mínimos cuadrados ortogonales (OLS). [Babu & Karthik (2007)] describen este procedimiento en el que en cada generación, el árbol que constituye a cada individuo es dividido en sub-árboles que componen los términos de un modelo lineal en parámetros (la totalidad de la estructura). El algoritmo de mínimos cuadrados ortogonales calcula las relaciones de reducción del error, medida de la disminución en la varianza de la salida, de cada uno de estos sub-árboles, permitiendo eliminar aquellas ramas con baja o nula contribución. De acuerdo con [Babu & Karthik (2007)], esta metodología produce soluciones simples y precisas.

Aunque con resultados satisfactorios en la mayoría de los casos presentados, la inclusión de estas técnicas de control de complejidad conlleva la inclusión de nuevas rutinas que pueden aumentar, considerablemente, el tiempo y costo computacional requerido.

3.2. Propuesta de modificación: contextualización.

La forma de representación de las soluciones potenciales considerada dentro de la *regresión simbólica multigen* trabajada por autores como [Hii *et al.* (2011)] y [Searson *et al.* (2010)] cuenta con características interesantes que dan pie a la generación de interrogantes a partir de los cuales es posible plantear una nueva propuesta de modificación que contempla la hibridación de este algoritmo mediante la inclusión de algunos otros conceptos de regresión. En las siguientes secciones se expande el concepto de *regresión simbólica multigen* y se presenta la teoría matemática necesaria para comprender el funcionamiento del algoritmo híbrido resultado de la modificación propuesta.

3.2.1. Algoritmo de regresión simbólica multigen.

Una de las modificaciones realizadas al algoritmo original de regresión simbólica, considerada, entre otros, en los trabajos propuestos por [Hii *et al.* (2011)] y [Searson *et al.* (2010)], consiste en cambiar la manera de representar cada solución potencial. Esta variación resulta en una *regresión simbólica multigen*. En ella, cada individuo se compone, no de una única estructura de árbol, sino de un número máximo de expresiones matemáticas codificadas de esta forma. En este caso, el valor de la inferencia dado por el individuo deriva de la combinación lineal ponderada del resultado de la evaluación de los datos de las variables independientes en las estructuras. Así, un ejemplo de individuo con tres genes puede ser el mostrado en la figura 3.5.

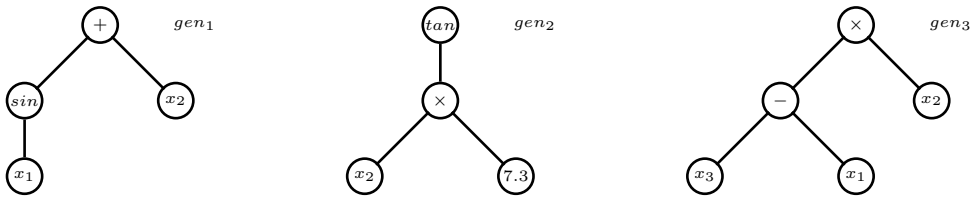


Figura 3.5: Ejemplo de representación multigen en combinación lineal ponderada.

El resultado de la inferencia dada por el individuo de la figura 3.5 ante una entrada com-

puesta por los valores de tres variables independientes, sería:

$\hat{y} = \alpha_0 + \alpha_1 \times gen_1 + \alpha_2 \times gen_2 + \alpha_3 \times gen_3$
$\hat{y} = \alpha_0 + \alpha_1 \times (sin(x_1) + x_2) + \alpha_2 \times (tan(7.3 \times x_2)) + \alpha_3 \times (x_2 \times (x_3 - x_1))$

Donde $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ constituyen los pesos de ponderación y x_i, con $i = 1, 2, 3$, las variables independientes consideradas en el problema de inferencia abordado.

Esta representación supone que al evaluar, en los genes de un individuo, cada una de las instancias que componen la base de datos experimental, será posible construir un sistema de ecuaciones lineales cuyo tamaño dependerá del número de instancias contempladas en la base. Este sistema de ecuaciones lineales puede expresarse en forma matricial de acuerdo con la ecuación 3.1.

$$\hat{\mathbf{y}} = \mathbf{X}\alpha \tag{3.1}$$

Las filas de la matriz $\mathbf{X}$ corresponden a los vectores $(\mathbf{x_1}', \mathbf{x_2}', \ldots, \mathbf{x_l}')$ cuyos elementos son el resutado de evaluar cada instancia de la base de datos en cada uno de los genes del individuo. Su dimensión es $l \times (g + 1)$ siendo l el número de instancias y g el número de genes del individuo considerado. El vector $\hat{\mathbf{y}}$ es el vector de inferencia mientras que el vector α agrupa los $g + 1$ pesos de ponderación a hallar de forma tal que la discrepancia entre el vector de inferencia $(\hat{\mathbf{y}})$ y el valor real de estimación $(\mathbf{y})$ consignado en la base de datos sea mínima. El cuadrado del error se utiliza como medida del nivel de discrepancia que, para el i-ésimo dato o instancia, se denota como se muestra en la ecuación 3.2.

$$L((x_i, y_i), \hat{y}_i) = (y_i - \hat{y}_i)^2 \tag{3.2}$$

Surge así, la necesidad de hallar, para cada individuo, un vector de pesos α para los cuales, la suma de la función de error sobre el total de instancias de la base de datos $(\mathbf{S})$ evaluadas en éste sea mínima, dando preferencia, dado el caso de hallar varios vectores de peso que satisfagan esta condición, al vector de pesos con mínima norma.

En el desarrollo de ambos trabajos ([Hii *et al.* (2011)] y [Searson *et al.* (2010)]) se propone hallar el valor de estos coeficientes mediante la aplicación del algoritmo de mínimos cuadra-

dos ordinarios cuya matemática se describe en secciones posteriores.

Contando con el vector de pesos que pondera los genes de un individuo, es posible realizar inferencia ante un nuevo conjunto de valores de las variables independientes, aplicando la relación mostrada en la ecuación 3.3) en la que $\mathbf{x}$ corresponde al vector de dimensión $g+1$ resultado de evaluar los valores de las variables independientes en las g expresiones codificadas en árbol que forman los genes de un individuo en particular.

$$\hat{y} = \langle \alpha, \mathbf{x} \rangle \tag{3.3}$$

Esta forma de entender la codificación de los individuos y el cálculo de inferencia en el algoritmo de regresión simbólica ha demostrado ser ventajoso respecto a las consideraciones del algoritmo original. Contar con más de una estructura en forma de árbol por individuo permite introducir mayor número de combinaciones de funciones, operaciones y terminales (variables y constantes). Esta situación adiciona a su vez cierto grado de diversidad en la búsqueda de la expresión que mejor se ajuste a los datos contemplados.

3.2.2. Regresión lineal mediante mínimos cuadrados ordinarios.

Mediante regresión lineal, los autores [Hii *et al.* (2011)] y [Searson *et al.* (2010)] pretenden hallar los elementos del vector α que generen mejor ajuste en la ecuación 3.1 para un individuo en particular. Para ello, se busca minimizar la función de pérdida contemplada en la ecuación 3.4, equivalente a la suma del error cuadrático de estimación por cada instancia de la base de datos ($\mathbf{S}$), es decir, la diferencia entre el valor real de la variable a inferir y el valor inferido al evaluar en la expresión los valores de las variables independientes.

$$L(\alpha, S) = \sum_{i=1}^{l} ((y_i - \hat{y}_i)^2) \tag{3.4}$$

En forma matricial, esta función de pérdida puede expresarse como:

$$L(\alpha, S) = (\mathbf{y} - \mathbf{X}\alpha)'(\mathbf{y} - \mathbf{X}\alpha) \tag{3.5}$$

expresión a partir de la cual, al aplicar ciertas propiedades de la operación de transposición de matrices, es posible obtener:

$$L(\alpha, S) = (\mathbf{y}' - \alpha'\mathbf{X}')(\mathbf{y} - \mathbf{X}\alpha) = \mathbf{y}'\mathbf{y} - \alpha'\mathbf{X}'\mathbf{y} - \mathbf{y}'\mathbf{X}\alpha + \alpha'\mathbf{X}'\mathbf{X}\alpha \tag{3.6}$$

En la ecuación 3.6, los términos $\alpha'\mathbf{X}'\mathbf{y}$ y $\mathbf{y}'\mathbf{X}\alpha$ resultan en escalares idénticos y por tanto es posible llegar a la expresión:

$$L(\alpha, S) = \mathbf{y}'\mathbf{y} - 2\alpha'\mathbf{X}'\mathbf{y} + \alpha'\mathbf{X}'\mathbf{X}\alpha \tag{3.7}$$

para la función de costo a minimizar.

Encontrar el mínimo de esta función de costo supone la necesidad de derivar la expresión respecto al vector de variables α e igualar a cero:

$$\frac{dL(\alpha, S)}{d\alpha} = -2\mathbf{X}'\mathbf{y} + \mathbf{X}'\mathbf{X}\alpha + \alpha'\mathbf{X}'\mathbf{X} = 0 \tag{3.8}$$

El hecho de que el producto de una matriz por su transpuesta sea una matriz simétrica, permite afirmar que, en la ecuación 3.8, $\mathbf{X}'\mathbf{X}\alpha = \alpha'\mathbf{X}'\mathbf{X}$ y por tanto resulta:

$$\frac{dL(\alpha, S)}{d\alpha} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\alpha = 0 \tag{3.9}$$

De la ecuación 3.9 es posible despejar el valor del vector de pesos α siempre y cuando $(\mathbf{X}'\mathbf{X})^{-1}$ exista:

$$\alpha = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{3.10}$$

[Hii *et al.* (2011)] y [Searson *et al.* (2010)], hacen uso en sus desarrollos no de la inversa de $(\mathbf{X}'\mathbf{X})$ sino de su pseudo-inversa para asegurar obtener una solución para el vector de pesos.

3.2.3. Algoritmo de regresión simbólica multigen con regresión no lineal.

El tipo de regresión simbólica multigen considerada hasta el momento conlleva a la búsqueda de un vector de pesos α que ponderan en forma lineal el conjunto de genes que componen a un individuo en particular. Si bien, aquellos autores que han indagado en esta metodología

abogan la obtención de mejores resultados en el proceso de inferencia, la suposición de existencia de una relación lineal entre los genes de un individuo despierta la inquietud acerca del comportamiento del algoritmo si se considerara una relación no lineal de los mismos.

A partir de la teoría multigen detrás del algoritmo trabajado por [Hii *et al.* (2011)] y [Searson *et al.* (2010)] y de la matemática desarrollada en regresión lineal mediante mínimos cuadrados ordinarios, es posible considerar la inclusión de otro tipo de regresión, lineal en esencia, pero cuya representación permite trabajar en búsqueda de relaciones no lineales.

De acuerdo con el proceso descrito en la sección anterior, hallar un vector de pesos de ponderación en el proceso de mínimos cuadrados ordinarios depende de la existencia de la inversa de la matriz $(X'X)$. Los autores [Hii *et al.* (2011)] y [Searson *et al.* (2010)] hacen uso de la pseudo-inversa para sobreponerse a problemas dado el caso de la singularidad de la matriz. Sin embargo, hacer uso de la pseudo-inversa no es la única solución. Expresar el problema de mínimos cuadrados mediante la representación dual e incluir un término de concesión entre la minimización del error de aproximación y la norma del vector de pesos de ponderación, proceso conocido como «regresión ridge», constituye una segunda opción. Esta concepción trae consigo, además, otra fuerte ventaja. Las expresiones matemáticas resultantes permiten la inclusión de la noción de funciones kernel en el proceso de regresión lineal contemplado por el algoritmo de mínimos cuadrados. Este último supone una estimación a partir de la combinación lineal ponderada de los datos. Considerar una combinación no lineal de estos últimos podría generar mejores resultados de ajuste. El objetivo de incluir funciones kernel en el proceso de regresión lineal es, justamente, realizar regresión no lineal a través del transporte de los datos a un nuevo espacio o espacio de características en el que realizar regresión lineal corresponde a realizar regresión no lineal en el espacio original.

3.3. Teoría matemática tras la modificación propuesta.

3.3.1. Representación dual del problema de regresión lineal.

Suponiendo la existencia de la inversa de $(\mathbf{X}'\mathbf{X})$, también es posible expresar el cálculo del vector de pesos de ponderación α como lo indica la ecuación 3.11 siendo $\beta = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-2}\mathbf{X}'\mathbf{y}$.

$$\alpha = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-2}\mathbf{X}'\mathbf{y} = \mathbf{X}'\beta \tag{3.11}$$

Esta expresión, conocida como representación dual del problema de regresión lineal, permite visualizar al vector de pesos α como una combinación lineal de los datos de la matriz $\mathbf{X}$ construida a partir de los datos experimentales de las variables independientes con los que se pretende realizar el entrenamiento del sistema de inferencia. La utilidad de este tipo de representación surgirá en secciones posteriores en las que se considere la inclusión de funciones kernel en el proceso de regresión.

Ahora bien, existen situaciones en las que el hallazgo de un vector de pesos de ponderación que permita el ajuste de los datos experimentales no es exitoso. Esto debido, ya sea a la falta de la cantidad suficiente de datos que aseguren la existencia de la inversa de la matriz $\mathbf{X}'\mathbf{X}$ o al tratamiento de datos considerablemente ruidosos que hagan del ajuste un proceso infructuoso. En estas situaciones es prudente introducir el concepto de *regularización*, en el que, mediante la inclusión de un nuevo término, se realizan concesiones entre el nivel de minimización del error de aproximación (precisión) y la minimización de la norma del vector de pesos (complejidad). Este concepto se introduce precisamente en el proceso conocido como «regresión rigde».

3.3.2. Regresión ridge con representación dual y regularización.

Este tipo de regresión pretende realizar optimización sobre la función de costo considerada en la ecuación 3.12, expresión en la que λ corresponde a un valor positivo de concesión entre la minimización de la norma del vector de pesos y la minimización del error de estimación.

$$min_{\alpha} L_{\lambda}(\alpha, S) = min_{\alpha} \lambda ||\alpha||^2 + \sum_{i=i}^{l} (y_i - \hat{y}_i)^2 \tag{3.12}$$

Derivando en la ecuación 3.12, con cálculos idénticos a los realizados en secciones anteriores, considerando que en forma matricial $\sum_{i=i}^{l}(y_i - \hat{y}_i)^2$ es igual a $(\mathbf{y} - \mathbf{X}\alpha)'(\mathbf{y} - \mathbf{X}\alpha)$, e igualando a cero en búsqueda del mínimo se obtiene:

$$2\lambda\alpha - 2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\alpha = 0 \tag{3.13}$$

equivalente a:

$$\alpha = \lambda^{-1}(\mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\alpha) = \lambda^{-1}\mathbf{X}'(\mathbf{y} - \mathbf{X}\alpha) = \mathbf{X}'\beta \tag{3.14}$$

donde $\beta = \lambda^{-1}(\mathbf{y} - \mathbf{X}\alpha)$.

Esta forma de expresar el cálculo del vector de pesos α permite, de nuevo, visualizarlo como una combinación lineal de los datos contenidos en la matriz $\mathbf{X}$. A partir de allí, el vector de coeficientes β puede reescribirse como:

$$\beta = \lambda^{-1}(\mathbf{y} - \mathbf{X}\mathbf{X}'\beta) \tag{3.15a}$$

$$\beta = (\mathbf{X}\mathbf{X}' + \lambda\mathbf{I})^{-1}\mathbf{y} \tag{3.15b}$$

En este caso, la matriz $(\mathbf{X}\mathbf{X}' + \lambda\mathbf{I})$ es siempre invertible siempre y cuando λ sea un valor positivo. La matriz $\mathbf{X}\mathbf{X}'$ es conocida como la matriz de «Gram» y es una matriz que define el producto escalar.

En la regresión «ridge», el cálculo de la inferencia ante un nuevo conjunto de datos ($\mathbf{x}$) se realiza mediante la expresión consignada en la ecuación 3.16, siendo $k_i = \langle \mathbf{x_i}, \mathbf{x} \rangle$, es decir, $\mathbf{k}$ equivale al vector compuesto por el producto punto entre cada uno de los l vectores de datos considerados en el proceso de entrenamiento ($\mathbf{x_i}$) y el nuevo vector de datos que ha llegado para efectuar la inferencia ($\mathbf{x}$).

$$\hat{y} = \langle \alpha, \mathbf{x} \rangle = \mathbf{y}'(\mathbf{X}\mathbf{X}' + \lambda\mathbf{I})^{-1}\mathbf{k} \tag{3.16}$$

Es de notar que para realizar el cálculo de la inferencia es necesario contar con la totalidad de los datos de la base de entrenamiento.

3.3.3. Regresión ridge no lineal mediante inclusión de funciones kernel.

Ambos procesos de regresión considerados en secciones anteriores se desenvuelven en el problema de hallar una relación *lineal* entre una variable seleccionada (variable a inferir) y otras «características». En el caso específico de la regresión simbólica multigen, se halla la relación lineal entre la variable a inferir y los genes de un individuo. Sin embargo, no es imperativo que considerar una relación lineal genere los mejores resultados. Es posible que una relación no lineal pueda mejorar el proceso de inferencia. Los métodos de regresión conocidos como métodos de kernel consideran, justamente, esta posibilidad. Estos métodos realizan el transporte de los datos a un nuevo espacio llamado «espacio de características» mediante la aplicación de una función de mapeo $\phi(\cdot)$. Obtener una relación lineal de los datos en este nuevo espacio corresponde a conseguir una relación no lineal en el espacio original [Shawe-Taylor & Cristianini (2004)].

Si en el espacio original se cuenta con un conjunto de datos de entrenamiento $((\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_l}, y_l))$ después de aplicar la función de mapeo se trabaja con una nueva base de datos $((\phi(\mathbf{x_1}), y_1), (\phi(\mathbf{x_2}), y_2), \ldots, (\phi(\mathbf{x_l}), y_l))$. La tarea consiste, ahora, en realizar regresión lineal sobre la nueva base de datos. El objetivo continúa siendo la minimización del error de aproximación $|\xi|$.

$$|\xi| = |y - \hat{y}| = |y - \langle \alpha, \phi(\mathbf{x}) \rangle| \tag{3.17}$$

El proceso de regresión «ridge» con representación dual propone el cálculo de la inferencia como se indica en la ecuación 3.18 siendo $\mathbf{G}$ la matriz de productos internos de los datos cuyos componentes se conforman como $G_{ij} = \langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle$ y $\mathbf{k}$ el producto interno entre el mapeo de un nuevo dato $\phi(\mathbf{x})$ y los datos de la base de entrenamiento cuya composición es $k_i = \langle \phi(\mathbf{x_i}), \phi(\mathbf{x}) \rangle$.

$$\hat{\mathbf{y}} = \mathbf{y}'(\mathbf{G} + \lambda \mathbf{I})^{-1}\mathbf{k} \tag{3.18}$$

El hecho de incluir a la matriz «Gram» en el proceso de inferencia, permite el cómputo eficiente del producto punto dato a dato mediante el uso de funciones kernel. Lo anterior sin necesidad de conocer la función de mapeo ($\phi(\cdot)$) ni la ubicación de los datos en el nuevo espacio tras su aplicación. Es ésta, justamente, una característica principal de los algoritmos cobijados bajo el nombre «métodos de kernel».

Función de kernel: función encargada de computar el producto interno en un espacio de características directamente desde las entradas en el espacio original. De acuerdo con la definición dada por [Shawe-Taylor & Cristianini (2004)], un kernel es una función $\kappa()$ que para todo $\mathbf{x}, \mathbf{z} \in \mathbf{X}$ satisface:

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \tag{3.19}$$

Donde $\phi(\cdot)$ es un mapeo desde el espacio original $\mathbf{X}$ a un espacio de características $\mathbf{F}$.

$$\phi(\cdot) : \mathbf{x} \longrightarrow \phi(\mathbf{x}) \in \mathbf{F} \tag{3.20}$$

Estas funciones de kernel, encargadas del cálculo del producto interno en otros espacios, cuentan con propiedades bien definidas y estudiadas (ver [Shawe-Taylor & Cristianini (2004)]).

Algunas de las funciones más conocidas y trabajadas son:

- *Kernel lineal:* $\kappa(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$
- *Kernel polinomial:* $\kappa(\mathbf{x}, \mathbf{z}) = p(\kappa(\mathbf{x}, \mathbf{z}))$ siendo $p(\cdot)$ un polinomio con coeficientes positivos y $\kappa(\mathbf{x}, \mathbf{z})$ la evaluación de cualquier otra función kernel. Frecuentemente se considera $\kappa_d(\mathbf{x}, \mathbf{z}) = p(\langle \mathbf{x}, \mathbf{z} \rangle + R)^d$.
- *Kernel gaussiano:* $\kappa(\mathbf{x}, \mathbf{z}) = exp(-||\mathbf{x} - \mathbf{z}||^2/2\sigma^2)$ siendo σ un parámetro ajustable.

Además, es posible llegar a la construcción de nuevas funciones kernel, a partir de cierta función kernel inicial ($\kappa_i(\mathbf{x}, \mathbf{z})$), de acuerdo con las siguientes propiedades contempladas en [Shawe-Taylor & Cristianini (2004)].

- $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z}) + \kappa_2(\mathbf{x}, \mathbf{z})$
- $\kappa(\mathbf{x}, \mathbf{z}) = \alpha\kappa_1(\mathbf{x}, \mathbf{z})$

- $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z})\kappa_2(\mathbf{x}, \mathbf{z})$
- $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_3(\phi(\mathbf{x}), \phi(\mathbf{z}))$
- $\kappa(\mathbf{x}, \mathbf{z}) = p(\kappa_1(\mathbf{x}, \mathbf{z}))$
- $\kappa(\mathbf{x}, \mathbf{z}) = exp(\kappa_1(\mathbf{x}, \mathbf{z}))$

3.4. Propuesta de modificación: algoritmo híbrido de regresión simbólica multigen con regresión ridge no lineal mediante la inclusión de funciones kernel.

Trabajos como los realizados por [Hii *et al.* (2011)] y [Searson *et al.* (2010)] proponen asignar a cada individuo de la población, en el algoritmo evolutivo, un número máximo de genes o estructuras de árbol que albergan expresiones matemáticas que relacionan funciones y operaciones de variables independientes y constantes. En la evaluación del desempeño de cada individuo al ajuste de la información contenida en la base de datos de entrenamiento, se lleva a cabo un proceso de regresión lineal mediante mínimos cuadrados ordinarios que permite hallar el vector de pesos que mejor estimación y menor discrepancia genera entre el valor inferido y el real de una variable de interés.

Ahora bien, ¿qué pasaría si en la evaluación del desempeño de un individuo no se considera una relación lineal entre sus genes y se hace uso de funciones kernel para obtener relaciones no lineales? Con el fin de evaluar esta posibilidad, se propone expresar el proceso de regresión de la evaluación de los genes de los individuos en términos de la regresión «ridge» con representación dual, representación que permite la inclusión de funciones kernel mediante la matriz «Gram» o de producto interno ($\mathbf{G}$) que expresada de forma general, en un espacio de características mapeado por la función $\phi(\cdot)$ correspondería a:

$$\mathbf{G}_{ij} = \langle \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) \rangle = \kappa(\mathbf{x_i}, \mathbf{x_j}) \tag{3.21}$$

$\mathbf{K}$	1	2	...	l
1	$\kappa(\mathbf{x_1}, \mathbf{x_1})$	$\kappa(\mathbf{x_1}, \mathbf{x_2})$	...	$\kappa(\mathbf{x_1}, \mathbf{x_l})$
2	$\kappa(\mathbf{x_2}, \mathbf{x_1})$	$\kappa(\mathbf{x_2}, \mathbf{x_2})$	...	$\kappa(\mathbf{x_2}, \mathbf{x_l})$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
l	$\kappa(\mathbf{x_l}, \mathbf{x_1})$	$\kappa(\mathbf{x_l}, \mathbf{x_2})$	...	$\kappa(\mathbf{x_l}, \mathbf{x_l})$

El funcionamiento del algoritmo de regresión simbólica, como proceso de búsqueda evolutivo, continúa sin alteración. Tan solo son modificadas las etapas de codificación de los individuos y de evaluación de desempeño de estos en cada una de las generaciones consideradas.

En síntesis, las variaciones en la implementación del algoritmo original de regresión simbólica necesarias para llevar a cabo la evaluación del efecto de la modificación propuesta incluyen:

- Modificar la función encargada de codificar los modelos potenciales, de forma tal que cada individuo de la población consista en un arreglo que albergue un número máximo de estructuras tipo árbol.
- Modificar la función encargada de la evaluación de desempeño de cada individuo incluyendo:
  - La construcción de la matriz $\mathbf{X}$, considerada en secciones anteriores, al evaluar los datos de entrenamiento en cada uno de los genes del individuo.
  - La construcción de la matriz de producto interno o matriz «Gram» ($\mathbf{G}$) mediante el hallazgo del kernel gaussiano entre cada una de las instancias consideradas dentro de la matriz $\mathbf{X}$.
  - El cálculo de la inferencia a partir de la aplicación de la ecuación 3.18.

Es de notar que el proceso de inferencia propuesto es dependiente de la cantidad de instancias consideradas en la base de datos, por lo tanto se hace imperativo proponer además cierto proceso de muestreo de la misma que permita reducir el tiempo computacional consumido en el procesamiento. Por esta razón, se implementa un muestreo aleatorio de la base de datos en el que cada dos generaciones son elegidas al azar un número de instancias para participar en el proceso de evaluación de la población, mientras que las restantes lo hacen

en la generación siguiente.

Un diagrama de flujo general, que describe el funcionamiento del algoritmo resultante, se muestra en la figura 3.6. El proceso de configuración de parámetros incluye tanto aquellos asociados al proceso evolutivo como los pocos contemplados dentro del proceso de construcción del kernel gaussiano y la regresión «ridge».

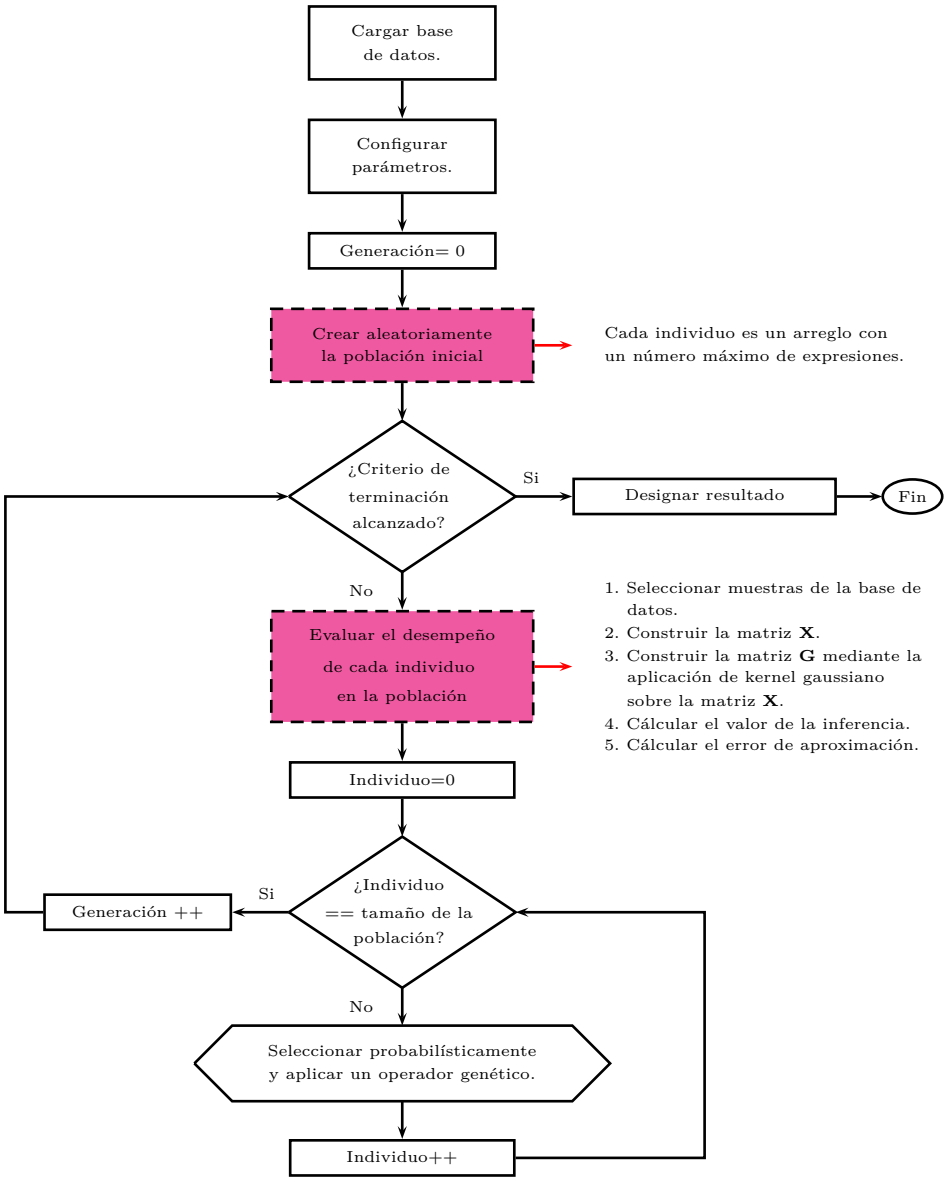


Figura 3.6: Diagrama de flujo general del algoritmo híbrido de programación genética propuesto.

¿Cuál es el aporte resultado de este proceso investigativo?

Mediante la aplicación de esta modificación se pretende estudiar el efecto de la combinación de dos diferentes teorías de regresión, ambas con buenos comentarios acerca de su utilidad en la solución de tareas de modelado ([Sharma & Tambe (2014)], [Bolshakov (2013)], [Douak *et al.* (2013)], [Shawe-Taylor & Cristianini (2004)]). Por un lado se cuenta con un algoritmo heurístico con características evolutivas y por otro se tiene un algoritmo de regresión perteneciente al conjunto de *métodos de kernel*. Es de esperar, y de hecho así se demuestra en secciones posteriores, que ambas teorías se complementen para entregar un mejor resultado final en la ejecución de su trabajo.

Hasta el momento no se conocen trabajos que efectúen esta combinación de teorías debido, posiblemente, al amplio espacio existente entre ambos campos de investigación. A pesar de consistir ambos en procesos de regresión, los métodos de kernel presentan cierta rigurosidad matemática que los aleja de la naturaleza que caracteriza a los métodos heurísticos. El máximo acercamiento, hallado en literatura científica, a conceptos de regresión «ridge» y programación evolutiva en conjunto son las propuestas hechas por [Ahn *et al.* (2012)] y [Castillo *et al.* (2011)], en las que se hace uso de teorías de programación genética para fortalecer el método lineal de regresión «ridge». Sin embargo, el algorítmo propuesto diverge en gran medida de estas propuestas y contempla, además, la inclusión de conceptos adicionales de regresión no lineal y funciones kernel.

Capítulo 4

Validación: resultados experimentales de la aplicación del algoritmo propuesto sobre bases de datos de variables industriales.

El algoritmo original de regresión simbólica, junto con el algoritmo multigen trabajado por [Hii *et al.* (2011)] y [Searson *et al.* (2010)] y el algoritmo híbrido propuesto, son aplicados sobre dos bases de datos de variables industriales seleccionadas de acuerdo con su disposición en repositorios virtuales; la primera de ellas resultado de la simulación de un proceso de neutralización de pH y la segunda consistente en valores relacionados con la calidad del concreto. Los algoritmos son evaluados bajo condiciones y valores de parámetros similares de acuerdo con recomendaciones y comentarios hechos por el autor del algoritmo original y concepciones personales después de procesos de prueba y ensayo. En ambos casos se considera una población inicial de 50 individuos que interactúan durante 100 generaciones con probabilidad de cruce de 90 % y 10 % de reproducción. Se considera, además, un comportamiento elitista del 10 % y metodología de selección proporcional al desempeño a partir de un «mating pool». Con el fin de entregar resultados estadísticamente confiables, los algoritmos son corridos y evaluados en 30 ocasiones. El error de estimación, haciendo uso de las bases de datos de validación, es determinante en la comparación de desempeños. El error mínimo, error máximo y promedio de error de estimación de los modelos hallados por los algoritmos son comparados mediante gráficas de cuartiles o diagramas de cajas y bigotes.

4.1. Resultados experimentales sobre la base de datos 1: Proceso de neutralización de pH.

La necesidad de monitoreo y control del valor de potencial de Hidrógeno (pH) es común en industrias de procesos químicos y biotecnológicos. Plantas de tratamiento de agua en las que es necesario mantener dentro de rangos establecidos el pH de las corrientes o procesos de producción de farmacéuticos son algunos ejemplos de estas industrias. Estos procesos suelen ser difíciles de modelar y controlar debido, principalmente, a su alta no linealidad y comportamiento variante en el tiempo [Henson & Seborg (1994)].

4.1.1. Descripción de la base de datos.

[Henson & Seborg (1994)] se dieron a la tarea de modelar, mediante el uso de ecuaciones y relaciones de conservación y equilibrio, el comportamiento de un sencillo sistema de neutralización de pH. Este proceso, mostrado en la figura 4.1, consiste en un flujo ácido (Q_1), un flujo neutro (Q_2) y un flujo base (Q_3) que son mezclados en el tanque 1. Previo al proceso de mezclado, el flujo ácido entra al tanque 2 en el que le son introducidas dinámicas adicionales. La velocidad de flujo del ácido y la base son reguladas mediante válvulas de control mientras que el flujo neutro es controlado de forma manual con un rotámetro. El nivel del tanque (h) y el pH del flujo de salida son variables medibles.

Si bien este modelo fue inicialmente concebido para estudiar el diseño de controladores, [Baffi *et al.* (1999)] hicieron uso del mismo para construir una base de datos de valores estáticos en los que valores aleatorios fueron asignados a los flujos Q_1, Q_2 y Q_3, simulando flujos de ácido nítrico (HNO3), bicarbonato de sodio (NaHCO3) e hidróxido de sodio (NaOH) respectivamente , mientras que el flujo de salida (Q_4) era variado para mantener un nivel (h) constante. El grado de pH en el flujo de salida era consignado al alcanzar el estado estable. La base de datos generada cuenta con un comportamiento altamente no lineal pues el flujo base (Q_3) fue variado de forma tal que los valores de pH recorren un rango establecido entre 3 y 11. Lo anterior genera un alto nivel de dependencia de la variable de salida del estado o valor del flujo Q_3.

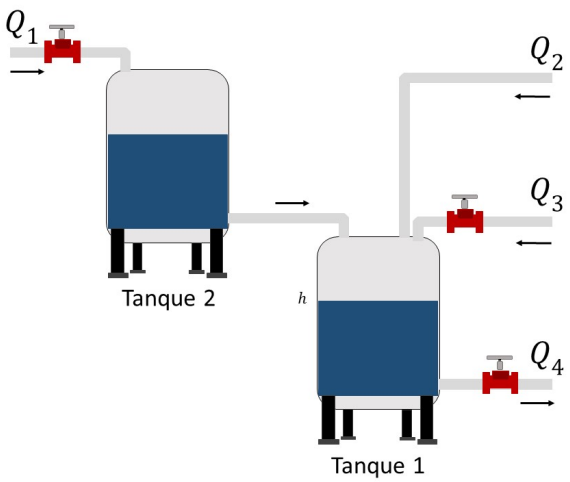


Figura 4.1: Sistema de neutralización de pH.

Asi pues, se cuenta con una base de datos con valores de cuatro variables independientes, Q_1 (flujo de ácido nítrico), Q_2 (flujo de bicarbonato de sodio), Q_3 (flujo de hidróxido de sodio), Q_4 (flujo de salida del proceso de neutralización) y una variable dependiente correspondiente al grado de pH en el flujo de salida del tanque 1. La base de datos trabajada consiste en 999 instancias divididas, con fines de modelado, en 700 puntos de entrenamiento y 299 de validación.

4.1.2. Resultado de la aplicación de los algoritmos de regresión simbólica sobre la base de datos 1.

Los tres algoritmos trabajos son ejecutados en 30 ocasiones haciendo uso de la base de datos descrita en el apartado anterior. En cada ejecución es seleccionado como solución, el modelo que genera el menor error de estimación sobre la base de datos en entrenamiento considerando la totalidad de las generaciones. Los resultados obtenidos, consistentes en el valor medio del error de estimación sobre las bases de entrenamiento y validación, se resumen en las siguientes secciones.

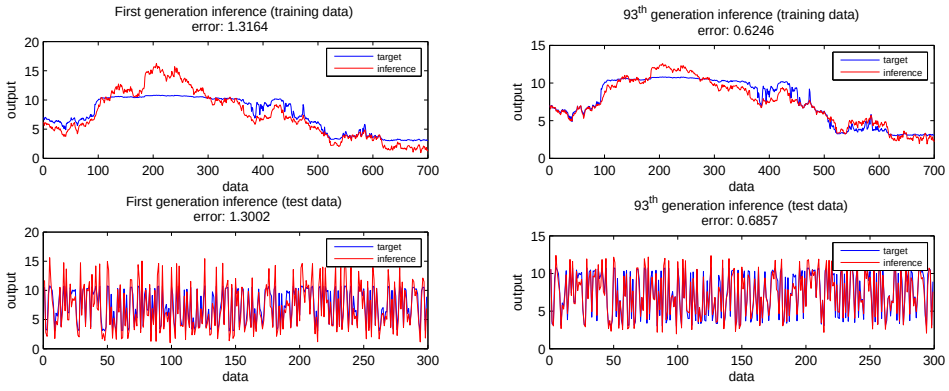
4.1.2.1. Algoritmo original

Los resultados obtenidos al ejecutar en 30 ocasiones el algoritmo original de regresión simbólica sobre la base de datos 1 se resumen en la tabla 4.1.

Entrenamiento			
	error mínimo	error máximo	error medio
	0.6246	2.1701	1.0053
Validación			
	error mínimo	error máximo	error medio
	0.6857	2.2694	1.0685

Cuadro 4.1: Errores de estimación generados en la aplicación del algoritmo original sobre la base de datos 1.

Por su parte, la gráfica 4.2, enfrenta el resultado de la estimación entregado por el mejor modelo en la generación inicial y el modelo solución obtenido por el algoritmo original. Esta gráfica corresponde al mejor resultado obtenido (mínimo error de validación) durante las 30 ejecuciones del algoritmo.



(a) (b)

Figura 4.2: Resultados sobre el mejor modelo obtenido mediante el algoritmo original de regresión simbólica. (a) error de estimación en la generación inicial. (b) error de estimación en la mejor generación.

La disminución en el error de estimación, con el correr de las generaciones, es notoria (52.6 % y 47.3 % para entrenamiento y validación respectivamente), dejando por sentado la validez del proceso evolutivo en la consecución de un modelo mucho más apto en la representación del sistema.

El modelo solución, en esta oportunidad, es hallado en la 93ª generación. Sin embargo, a lo largo de las 30 experimentaciones, el mejor individuo es hallado, en promedio, en la generación 42. Este bajo valor se debe a la simplicidad de la estructura tipo árbol manejado por el algoritmo, que limita la diversidad y por tanto el proceso de búsqueda después de cierto número de iteraciones.

4.1.2.2. Algoritmo multigen.

Al ejecutar en 30 ocasiones el algoritmo multigen de regresión simbólica sobre la base de datos 1 se obtienen los resultados mostrados en la tabla 4.2.

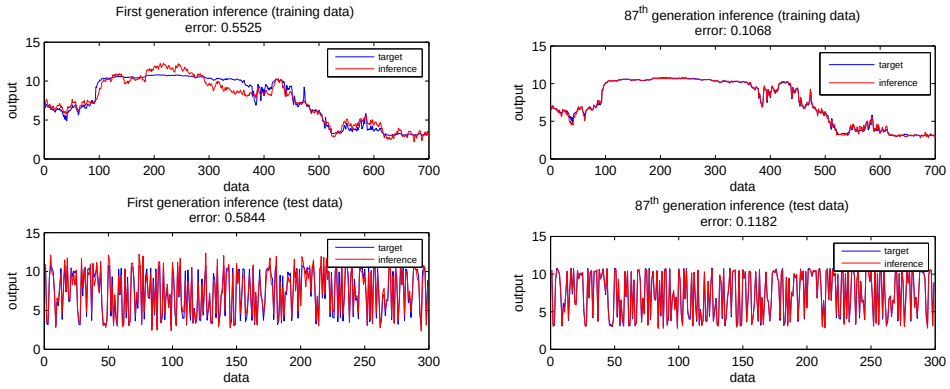
Entrenamiento			
	error mínimo	error máximo	error medio
	0.1068	0.3131	0.2018
Validación			
	error mínimo	error máximo	error medio
	0.1182	0.3241	0.2052

Cuadro 4.2: Errores de estimación generados en la aplicación del algoritmo multigen sobre la base de datos 1.

La dismunición del error de estimación, respecto a los valores generados por el algoritmo original, es considerable. Mediante la inclusión de la modificación multigen se obtienen disminuciones de alrededor de 80 % en ambos procesos de entrenamiento y validación.

Por su parte, la gráfica 4.3, enfrenta el resultado de la estimación entregado por el mejor modelo en la generación inicial y el modelo solución obtenido por el algoritmo multigen. Esta gráfica corresponde al mejor resultado obtenido (mínimo error de validación) durante

las 30 ejecuciones del algoritmo.



(a) (b)

Figura 4.3: Resultados sobre el mejor modelo obtenido mediante el algoritmo multigen de regresión simbólica. (a) error de estimación en la generación inicial. (b) error de estimación en la mejor generación.

4.1.2.3. Algoritmo híbrido.

Tras la ejecución en 30 ocasiones del algoritmo híbrido de regresión simbólica propuesto sobre la base de datos 1 se obtienen los resultados mostrados en la tabla 4.3.

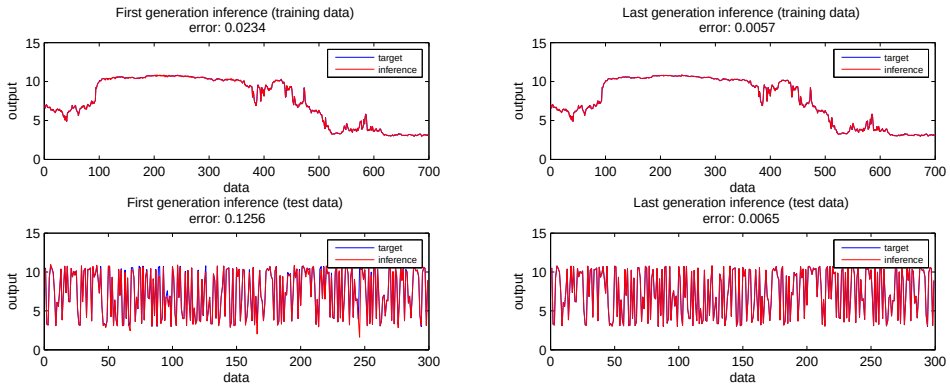
Entrenamiento			
	error mínimo	error máximo	error medio
	0.0057	0.0248	0.0108
Validación			
	error mínimo	error máximo	error medio
	0.0065	0.0265	0.0122

Cuadro 4.3: Errores de estimación generados en la aplicación del algoritmo híbrido sobre la base de datos 1.

Los resultados muestran disminución consistente del error de estimación, respecto a los valores generados por el algoritmo original y el algoritmo multigen. Gracias a la hibridación

mediante la inclusión de conceptos de regresión no lineal se obtienen disminuciones de alrededor de 98.9 % respecto al algoritmo original y de 94 % respecto a la modificación multigen.

Por su parte, la gráfica 4.4, enfrenta el resultado de la estimación entregado por el mejor modelo en la generación inicial y el modelo solución obtenido por el algoritmo híbrido propuesto. Esta gráfica corresponde al mejor resultado obtenido (mínimo error de validación) durante las 30 ejecuciones del algoritmo.



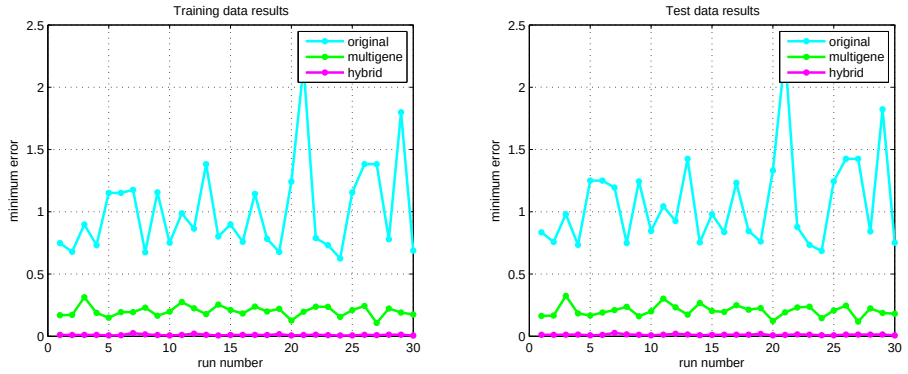
(a) (b)

Figura 4.4: Resultados sobre el mejor modelo obtenido mediante el algoritmo híbrido de regresión simbólica. (a) error de estimación en la generación inicial. (b) error de estimación en la mejor generación.

En este caso es importante resaltar la ausencia de características de sobre-entrenamiento altamente factibles en procesos que incluyen metodologías de funciones kernel. Dicha ausencia se ve reflejada en la similitud de los resultados de entrenamiento y validación. Un posible sobre-entrenamiento generaría altos valores de error en la validación respecto al proceso de entrenamiento. Esta ausencia lleva a concluir sobre la efectividad de la técnica de muestreo de la base de datos utilizada.

4.1.3. Comparación de desempeño de los algoritmos de regresión simbólica sobre la base de datos 1.

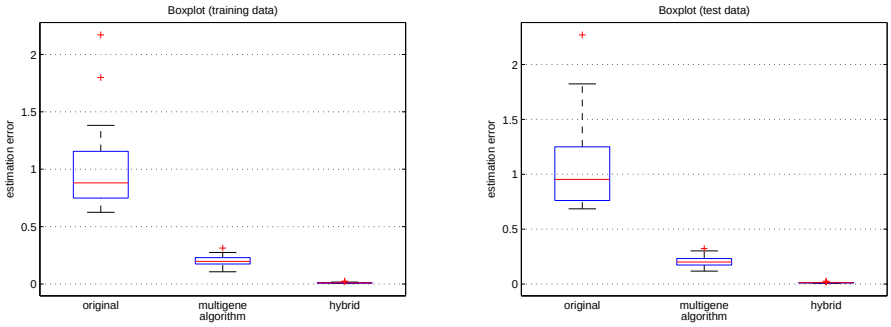
La figura 4.5 muestra el menor error de estimación, sobre los datos de entrenamiento y validación, generados por los mejores modelos hallados en las 30 ejecuciones de los algoritmos implementados. Esta imagen permite apreciar la disminución significativa del error de estimación al incluir las modificaciones, contempladas en secciones anteriores de este documento, al algoritmo original de regresión simbólica.



(a) (b)

Figura 4.5: Error de estimación generado por el mejor modelo obtenido por cada algoritmo en 30 ejecuciones. a) Con datos de entrenamiento. b) Con datos de validación.

Con el fin de analizar estadísticamente los resultados mostrados en la figura 4.5, se elabora el diagrama de cajas y bigotes de la gráfica 4.6. Este diagrama se encarga de presentar los valores mínimos y máximos de error obtenidos por cada algoritmo junto con información de la distribución de los resultados. En este diagrama son apreciables, de nuevo, los bajos valores de error generados por el algoritmo híbrido propuesto durante la totalidad de las ejecuciones. La mejora en precisión respecto, principalmente, al algoritmo original es contundente.



(a) (b)

Figura 4.6: Comportamiento del error de estimación generado por el mejor modelo obtenido por cada algoritmo en 30 ejecuciones.

4.1.4. Desempeño del algoritmo de máquinas de vector de soporte (SVM) en regresión sobre la base de datos 1.

Importante es, además de mostrar superioridad del algoritmo propuesto sobre algoritmos de computación evolutiva, hacerlo sobre otras técnicas consideradas en el diseño de dispositivos de medición inferencial. Por esta razón, bajo condiciones similares, la base de datos correspondiente al proceso de neutralización de pH es utilizada para entrenar una máquina de vector de soporte. Los resultados de estimación obtenidos se muestran en la figura 4.7.

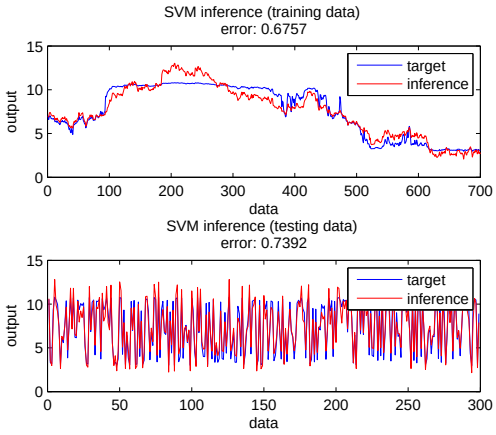


Figura 4.7: Error de estimación del modelo hallado mediante el algoritmo de máquinas de vector de soporte (SVM).

Con valores de error de aproximación de 0.7, el desempeño de la máquina de vector de soporte en la tarea de estimación sobre la base de datos considerada es relativamente similar al desempeño del algoritmo de regresión simbólica original. No supera, por lo tanto, los resultados obtenidos por el algoritmo híbrido propuesto.

4.2. Resultados experimentales sobre la base de datos 2: Resistencia a la compresión del concreto.

Si bien la medición de la resistencia a la compresión del concreto no es, precisamente, una variable que se mida en línea dentro de un proceso productivo como las variables objetivo de los sensores inferenciales, si es una variable de alto interés en la ingeniería cívil y la disposición de esta base de datos permite comprobar la eficiencia de los algoritmos en el modelado o identificación de relaciones.

4.2.1. Descripción de la base de datos.

La calidad del concreto es juzgada, en general, por su nivel de resistencia a la compresión. Esta medida es dependiente de las proporciones de materia prima utilizadas en la producción. La segunda base de datos utilizada en la evaluación del desempeño de los algoritmos de regresión simbólica trabajados consiste en 1030 instancias o ejemplos de 8 variables independientes que de una forma u otra han de relacionarse con el valor de esta propiedad.

Las 8 variables independientes de esta colección de datos consisten en valores de cantidades de cemento, escorias de alto horno, cenizas volátiles, agua, superplastificante, agregado grueso (grava), agregado fino (arena) y la edad de la mezcla. Las cantidades son dadas en kilogramos por metro cuadrado de mezcla y la edad se especifica en el rango [1,365] días.

Con fines de entrenamiento y validación, las 1030 instancias son repartidas en conjuntos de 721 y 309 ejemplos respectivamente.

4.2.2. Resultado de la aplicación de los algoritmos de regresión simbólica.

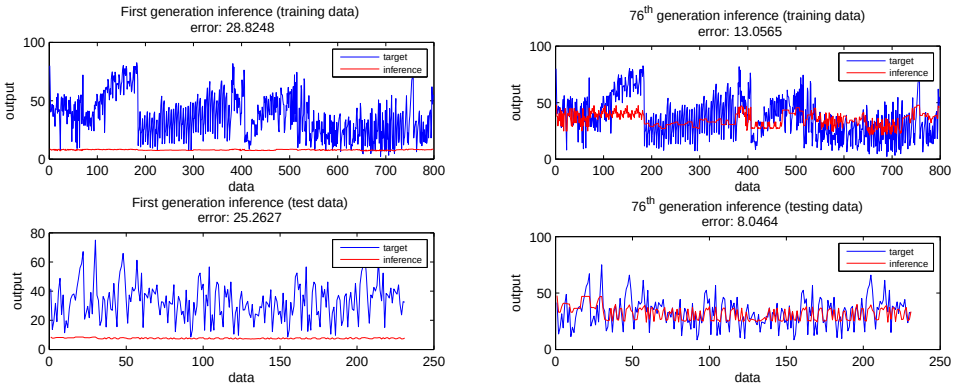
4.2.2.1. Algoritmo original.

Los resultados obtenidos al ejecutar en 30 ocasiones el algoritmo original de regresión simbólica sobre la base de datos 2 se resumen en la tabla 4.4.

Entrenamiento			
	error mínimo	error máximo	error medio
	9.8724	14.9965	13.9304
Validación			
	error mínimo	error máximo	error medio
	8.0464	14.1682	10.3542

Cuadro 4.4: Errores de estimación generados en la aplicación del algoritmo original sobre la base de datos 2.

Por su parte, la gráfica 4.8, enfrenta el resultado de la estimación entregado por el mejor modelo en la generación inicial y el modelo solución obtenido por el algoritmo original. Esta gráfica corresponde al mejor resultado obtenido (mínimo error de validación) durante las 30 ejecuciones del algoritmo.



(a) (b)

Figura 4.8: Resultados sobre el mejor modelo obtenido mediante el algoritmo original de regresión simbólica. (a) error de estimación en la generación inicial. (b) error de estimación en la mejor generación.

Tanto en la tabla 4.4 como en la figura 4.8 es observable el alto valor de error de estimación generado por los modelos solución hallados. Esta situación insatisfactoria ha de deberse a una compleja relación entre las variables involucradas imposible de describir por las estructuras simples consideradas en el algoritmo original de regresión simbólica.

4.2.2.2. Algoritmo multigen.

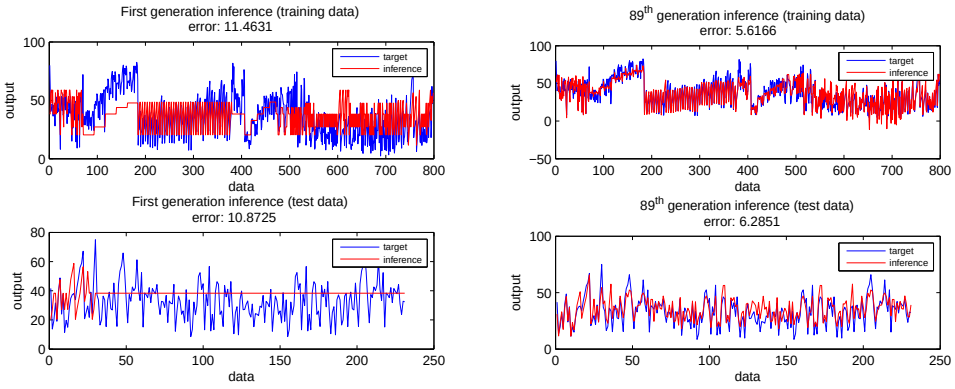
Al ejecutar en 30 ocasiones el algoritmo multigen de regresión simbólica sobre la base de datos 2 se obtienen los resultados mostrados en la tabla 4.5.

Entrenamiento			
	error mínimo	error máximo	error medio
	5.0979	7.1056	5.7898
Validación			
	error mínimo	error máximo	error medio
	6.2851	12.4146	9.2955

Cuadro 4.5: Errores de estimación generados en la aplicación del algoritmo multigen sobre la base de datos 2.

La disminución del error de estimación, respecto a los valores generados por el algoritmo original, es considerable, especialmente, sobre los datos de entrenamiento. Mediante la inclusión de la modificación multigen se obtienen disminuciones de alrededor de 58 % en este proceso. Los valores en el proceso de validación, por su parte, aunque disminuidos, no alcanzan el 11 %. Esta situación podría dar indicios de condiciones de sobre-entrenamiento dentro del procedimiento.

Por su parte, la gráfica 4.9, enfrenta el resultado de la estimación realizada por el mejor modelo en la generación inicial y el modelo solución obtenido por el algoritmo multigen. Esta gráfica corresponde al mejor resultado obtenido (mínimo error de validación) durante las 30 ejecuciones del algoritmo.



(a) (b)

Figura 4.9: Resultados sobre el mejor modelo obtenido mediante el algoritmo multigen de regresión simbólica. (a) error de estimación en la generación inicial. (b) error de estimación en la mejor generación.

Específicamente, en el caso mostrado en la figura 4.9, el modelo hallado mediante el algoritmo multigen da señales de un comportamiento altamente sobresaliente respecto al modelo obtenido mediante la ejecución del algoritmo original. Los valores de error, sin embargo, continúan siendo significativamente altos.

4.2.2.3. Algoritmo híbrido.

Tras la ejecución en 30 ocasiones del algoritmo híbrido de regresión simbólica propuesto sobre la base de datos 2 se obtienen los resultados mostrados en la tabla 4.6.

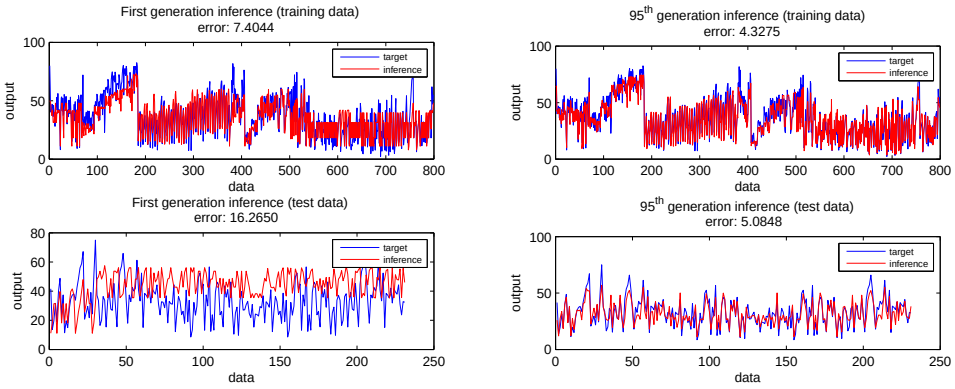
Entrenamiento			
	error mínimo	error máximo	error medio
	3.7410	8.5609	5.1559
Validación			
	error mínimo	error máximo	error medio
	5.0848	13.3288	9.0593

Cuadro 4.6: Errores de estimación generados en la aplicación del algoritmo híbrido sobre la base de datos 2.

Los resultados muestran disminución consistente del error de estimación, respecto a los valores generados por el algoritmo original. Gracias a la hibridación mediante la inclusión de conceptos de regresión no lineal se obtienen disminuciones de alrededor de 63 % respecto al algoritmo original en etapa de entrenamiento. En validación, sin embargo, la mejora en el error de estimación, aunque positiva, no es tan gratificante (13 %).

Los resultados promedio obtenidos por el algoritmo híbrido son relativamente similares a aquellos derivados del algoritmo de regresión multigen. Sin embargo, los valores para error mínimo tanto en entrenamiento como en validación, son considerablemente mejores. Lo anterior significa que, aunque el algoritmo híbrido genera modelos con desempeño similar a aquellos creados por el algoritmo multigen, los mejores modelos durante las 30 ejecuciones de los algoritmos fueron generados por el algoritmo híbrido propuesto.

Por su parte, la gráfica 4.10, enfrenta el resultado de la estimación entregado por el mejor modelo en la generación inicial y el modelo solución obtenido por el algoritmo híbrido propuesto. Esta gráfica corresponde al mejor resultado obtenido (mínimo error de validación) durante las 30 ejecuciones del algoritmo.



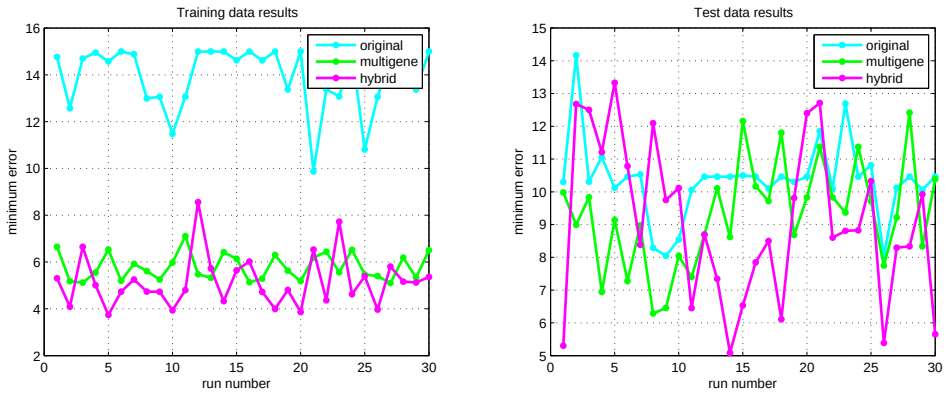
(a) (b)

Figura 4.10: Resultados sobre el mejor modelo obtenido mediante el algoritmo híbrido de regresión simbólica. (a) error de estimación en la generación inicial. (b) error de estimación en la mejor generación.

El comportamiento del modelo mostrado en la figura 4.10 es significativamente sobresaliente respecto a aquellos considerados en las figuras 4.8 y 4.9 para los mejores modelos generados mediante el algoritmo original y el algoritmo multigen respectivamente. El error de estimación, tanto de entrenamiento como de validación, continúa siendo considerable. Esta situación refleja la complejidad de la relación entre las variables involucradas en el proceso que se pretende modelar.

4.2.3. Comparación de desempeño de los algoritmos de regresión simbólica sobre la base de datos 2.

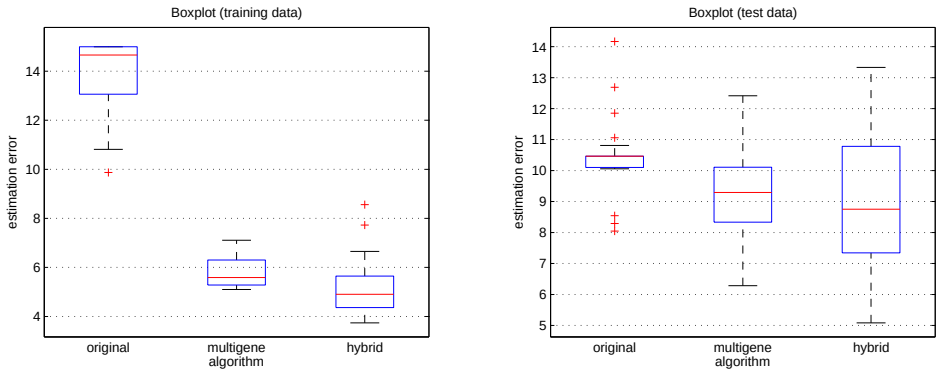
La figura 4.11 muestra el menor error de estimación, sobre los datos de entrenamiento y validación, generados por los mejores modelos hallados en las 30 ejecuciones de los algoritmos implementados. Esta imagen permite apreciar una disminución significativa del error de estimación en el proceso de entrenamiento al incluir modificaciones en el algoritmo original. Sin embargo, las variaciones en el error de estimación evaluado sobre datos de validación no son bien definidas.



(a) (b)

Figura 4.11: Error de estimación generado por el mejor modelo obtenido por cada algoritmo en 30 ejecuciones. a) Con datos de entrenamiento. b) Con datos de validación.

Si bien la situación descrita en el párrafo anterior podría poner en duda la utilidad de las modificaciones realizadas, la gráfica 4.12, que presenta los valores mínimos y máximos de error obtenidos por cada algoritmo, durante 30 ejecuciones, junto con información de la distribución de los resultados, deja entrever que los mejores modelos, es decir, aquellos que durante las 30 ejecuciones generaron el menor error de estimación tanto de entrenamiento como de validación, fueron generados por el algoritmo híbrido que considera la inclusión de conceptos de regresión no lineal mediante funciones kernel.



(a) (b)

Figura 4.12: Comportamiento del error de estimación generado por el mejor modelo obtenido por cada algoritmo en 30 ejecuciones.

A pesar de que el algoritmo de regresión híbrido propuesto no genera resultados con igual nivel de satisfacción respecto a aquellos obtenidos sobre la base de datos correpondiente al proceso de neutralización de pH, si lo hace respecto a los resultados generados por modelos provenientes de otros algoritmos. En los resultados mostrados, es notoria una posible tendencia a situaciones de sobre-entrenamiento que más que debido al funcionar del algoritmo ha de deberse a la complejidad de la relación a hallar.

4.2.4. Desempeño del algoritmo de máquinas de vector de soporte (SVM) sobre la base de datos 2.

El resultado de la aplicación del algoritmo de máquinas de vector de soporte sobre la base de datos considerada es apreciable en la figura 4.13.

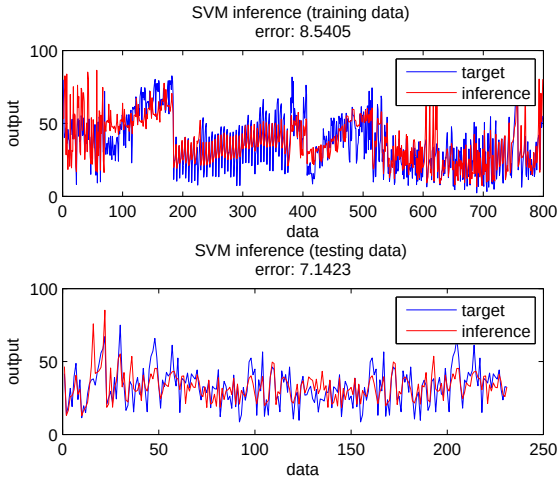


Figura 4.13: Error de estimación del modelo hallado mediante el algoritmo de máquinas de vector de soporte (SVM).

El alto valor de los errores de estimación continúa acentuando la hipótesis que plantea una relación compleja entre las variables consideradas dentro del proceso. Los valores obtenidos son, en general, mayores a aquellos generados por el algoritmo multigen e híbrido de las técnicas evolutivas, lo que enfatiza el desempeño satisfactorio del algoritmo de regresión simbólica modificado en la tarea de estimación.

Capítulo 5

Conclusión: discusión sobre los resultados obtenidos y planteamiento de trabajo futuro.

En esta sección se presenta una corta discusión sobre el proceso investigativo y los resultados obtenidos y presentados en secciones anteriores. Los comentarios han sido construidos a partir de la vasta interacción con los algoritmos y bases de datos trabajados. Si bien, el objetivo del proceso investigativo ha sido alcanzado mediante la inclusión, al algoritmo de regresión simbólica original, de una modificación que genera resultados de estimación mucho más precisos, de éste surgen algunas inquietudes e interrogantes que servirán como inspiración en el planteamiento de trabajo futuro.

5.1. Discusión.

El objetivo general de este proceso investigativo consistía en el planteamiento de alguna modificación al algoritmo de regresión simbólica original que trajera consigo mejoras, bien fuera en precisión, generalización o autonomía, en el proceso de estimación ligado al ejercicio de medición inferencial. El resultado final consiste en un algoritmo híbrido, a partir de la inclusión de conceptos de regresión no lineal y funciones kernel dentro del proceso evolutivo de búsqueda planteado por el algoritmo de regresión simbólica.

El punto de partida de esta propuesta yace en la modificación planteada dentro del algoritmo de regresión simbólica multigen trabajada por algunos autores, entre ellos [Hii *et al.* (2011)] y [Searson *et al.* (2010)]. Esta modificación plantea una forma más compleja de representar la «información genética» dentro del proceso de búsqueda evolutivo y una manera diferente de evaluar el desempeño de los modelos hallados. La forma de representación propuesta permite la construcción de expresiones matemáticas más complejas y sus combinaciones que, en teoría, han de permitir obtener modelos que se ajustan con mayor facilidad a los datos trabajados.

Buscando complementar esta modificación, que plantea la representación de cada posible solución como una combinación lineal de expresiones matemáticas, se propone suponer una relación no lineal de las mismas y se incluyen en el algoritmo las funciones necesarias para introducir conceptos de *regresión «ridge», representación dual y funciones kernel*. La inclusión de estas últimas, sin embargo, si bien trae consigo la posibilidad de realizar regresión mucho más ajustada a los datos de entrenamiento, también acarrea mayores posibilidades de alcanzar características de sobre-entrenamiento en el proceso. Por esta razón, fue importante considerar métodos para sobrepasar este tipo de situaciones y de allí la importancia de incluir, con resultados suficientemente satisfactorios, técnicas de muestro de la base de datos.

El desempeño en la tarea de modelado del algoritmo final, junto con sus algoritmos base, es evaluado sobre dos bases de datos de variables experimentales disponibles en repositorios virtuales. Los resultados obtenidos permiten afirmar la consecución del objetivo planteado pues valores de error de estimación dados por modelos generados por el algoritmo propuesto son significativamente menores a aquellos valores de error entregados por modelos generados

mediante las metodologías restantes, alcanzando así una mejora consistente en **precisión**.

Los resultados obtenidos en la aplicación del algoritmo propuesto sobre la base de datos 1 son contundentes. Porcentajes de reducción del error de estimación del 98 %, mediante el uso de modelos generados por este algoritmo respecto a modelos generados por el algoritmo original, así lo confirman. El valor promedio de error de validación cercano a 0.01 es significativamente bajo y evidencia la obtención de modelos de estimación suficientemente precisos. Estos resultados manifiestan, de igual forma, una estrecha relación entre las variables involucradas en el proceso de neutralización de pH trabajado que permite, a los algoritmos de regresión, hallar modelos o expresiones que la representen satisfactoriamente.

Por otra parte, si bien los resultados obtenidos al aplicar los algoritmos sobre la base de datos 2 parecieran no ser tan gratificantes, es posible sacar de allí ciertas conclusiones. En primera instancia, aunque en porcentajes más pequeños, continúan presentándose disminuciones en el error de estimación al ejecutar el algoritmo híbrido propuesto. Los valores promedio de error de estimación de modelos generados por este algoritmo son menores a los de aquellos generados tanto por el algoritmo original como por el algoritmo multigen. Además, la diferencia entre los errores de estimación mínimos generados por los algoritmos es significativa, lo que permite afirmar que los mejores modelos de estimación obtenidos fueron generados por el algoritmo modificado propuesto.

Los altos valores de error de estimación alcanzados en esta experimentación han de deberse a una relación altamente compleja de las variables involucradas dentro del proceso. Sin embargo, al no obtener valores más bajos mediante la aplicación de otros algoritmos, es posible calificar como satisfactorio el desempeño de la metodología propuesta.

Finalmente, los resultados de estimación obtenidos mediante la aplicación de las técnicas evolutivas trabajadas son comparados con resultados de estimación generados al considerar el algoritmo de máquinas de vector de soporte. El desempeño de este último es relativamente similar al del algoritmo de regresión simbólica original y por lo tanto inferior al de aquel de los algoritmos de regresión simbólica modificados, confirmando así la utilidad de las alteraciones planteadas.

En general, a partir del proceso investigativo desarrollado es posible llegar a las siguientes conclusiones:

- Las modificaciones realizadas sobre el algoritmo de regresión simbólica original permiten la obtención de modelos que generan estimaciones con mayor grado de precisión.
- La simplicidad de las estructuras que representan las soluciones en el algoritmo de regresión simbólica original constituye una fuerte limitación en el modelado de relaciones con un nivel considerable de complejidad, limitación superada por los algoritmos modificados.
- La totalidad de los algoritmos evolutivos trabajados muestran una disminución del error de estimación con el pasar de las generaciones, situación que confirma la utilidad del proceso evolutivo. Esta situación es sumamente importante pues ratifica la complementariedad del proceso evolutivo y las técnicas adicionales consideradas.
- En el proceso de búsqueda de modelos es fundamental evitar características de sobreentrenamiento. Los resultados obtenidos, para el algoritmo propuesto, muestran ausencia de esta característica corroborando la utilidad de las técnicas de muestreo de la base de datos en el entrenamiento.
- Aunque la inclusión de conceptos de regresión «ridge» y funciones kernel dentro del algoritmo de regresión simbólica trae consigo ventajas respecto a la precisión en la estimación, el algoritmo resultante pierde la posibilidad de acceder a una expresión matemática transparente y se convierte en un método de estimación tipo caja negra. Esta característica podría llegar a ser confusa para quienes pretenden obtener un modelo interpretable mas no afecta el desarrollo del proceso de inferencia.

5.2. Trabajo futuro.

Los resultados plasmados en este documento corresponden a ejecuciones de los algoritmos dentro de un contexto de experimentación construido de acuerdo con recomendaciones realizadas por el diseñador del algoritmo original de regresión simbólica. Los algoritmos tratados, sin embargo, cuentan con una gran cantidad de características cuya sintonización podría llevar a obtener aún mejores resultados. Como se menciona en diferentes apartados de este

escrito, la sintonización de los parámetros que comprenden, en general, los algoritmos de computación evolutiva es dependiente del tipo de problema enfrentado y características propias de las variables seleccionadas. El algoritmo híbrido propuesto cuenta, a su vez, con parámetros extra relacionados con el tipo de kernel a utilizar. A pesar del prestigio del kernel gaussiano en procesos de ajuste de datos con relaciones altamente no lineales, la experimentación con otro tipo de kernel dentro del algoritmo de regresión simbólica deja un camino abierto a exploración.

Igualmente, las modificaciones consideradas en este trabajo afectan, tan solo, la forma de representación de las soluciones potenciales en la búsqueda de modelos de estimación y la forma de evaluación del desempeño de las mismas. Sin embargo, como se evidencia en el capítulo 2, específicamente en la sección de evaluación de impacto de los parámetros involucrados en el algoritmo de regresión simbólica y en el capítulo 3 que presenta las propuestas de modificación hechas por otros autores, son muchas las variaciones que pueden ser estudiadas en la creación de un algoritmo de regresión completo y robusto.

Siendo así, trabajo futuro consistirá en la aplicación del algoritmo híbrido propuesto sobre datos de variables experimentales, realizando variaciones sobre la multitud de parámetros influyentes para lograr una mejor sintonización, además de la evaluación del efecto de modificaciones extra sobre el desempeño de la herramienta.

Bibliografía

[Acuña *et al.* (2014)] Acuña, G., Curilem, M., & Cubillos, F. (2014). Desarrollo de un Sensor Virtual basado en Modelo NARMAX y Máquina de Vectores de Soporte para Molienda Semiautógena. Revista Iberoamericana de Automática e Informática Industrial RIAI, 11(1), 109-116.

[Ahn *et al.* (2012)] Ahn, J. J., Byun, H. W., Oh, K. J., & Kim, T. Y. (2012). Using ridge regression with genetic algorithm to enhance real estate appraisal forecasting. Expert Systems with Applications, 39(9), 8369-8379.

[Akaike (1974)] Akaike, H. (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6), 716-723.

[ANDI (2014)] Informe Encuesta de Opinión Industrial Conjunta (EOIC) Agosto 2014. (2014). Recuperado el 9 de Noviembre de 2014 de http://www.andi.com.co/pages/proyectos_paginas/proyectos_detail.aspx?pro_id= 559&Id= 3&clae= 8&Tipo= 3

[Babu & Karthik (2007)] Babu, B. V., & Karthik, S. (2007). Genetic Programming for Symbolic Regression of Chemical Process Systems. Engineering Letters, 14(2), 42-55.

[Babuska (2004)] Babuska, R. (2004). Fuzzy and Neural Control Disc Course: Lecture Notes. Delft, The Netherlands: Delft University of Technology.

[Baffi *et al.* (1999)] Baffi, G., Martin, E. B., & Morris, A. J. (1999). Non-linear projection to latent structures revisited (the neural network PLS algorithm). Computers & Chemical Engineering, 23(9), 1293-1307.

[Bishop & Welch (2001)] Bishop, G., & Welch, G. (2001). An introduction to the kalman filter. Proc of SIGGRAPH, Course, 8(27599-23175), 41.

[Bolshakov (2013)] Bolshakov, V. (2013). Regression-based Daugava River Flood Forecasting and Monitoring. Information Technology and Management Science, 16(1), 137-142.

[Botero *et al.* (2009)] Botero C, H. A., Álvarez Z, H. D. (2009). Una Revisión de los métodos más frecuentes para la estimación del estado en procesos químicos. Dyna, 76(158), 135-146.

[Butler & Zhang. (2012)] Butler, D., & Zhang, H. (2012, September). Intelligent software sensors and process prediction for glass container forming processes based on multivariate statistical process control techniques. In IEEE 2012 UKACC International Conference on Control (CONTROL), 281-285.

[Byington *et al.* (2012)] Byington, C. S., Mackos, N. A., Argenna, G., Palladino, A., Reimann, J., & Schmitigal, J. (2012). Application of symbolic regression to electrochemical impedance spectroscopy data for lubricating oil health evaluation. In Proceedings of Annual conference of the prognostics and health management society 2012, Minneapolis, Minnesota.

[Blickle & Thiele (1995)] Blickle, T., & Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms.

[Cai *et al.* (2006)] Cai, W., Pacheco-Vega, A., Sen, M., & Yang, K. T. (2006). Heat transfer correlations by symbolic regression. International Journal of Heat and Mass Transfer, 49(23), 4352-4359.

[Castillo *et al.* (2011)] Castillo, F., Kordon, A., & Villa, C. (2011). Genetic programming transforms in linear regression situations. In Genetic Programming Theory and Practice VIII, 175-194.

[Coello & Zacatenco (2004)] Coello, C. A. C., & Zacatenco, C. S. P. (2004). Introducción a la computación evolutiva (notas de curso). Departamento de Ingeniería Eléctrica, Sección de Computación, Instituto Politécnico Nacional, México.

[Colmenares (2006)] Colmenares, W. (2006). PS2316. Estimadores de estado (notas de curso). Universidad Simón Bolivar. Departamento de Procesos y Sistemas.

[Donís Díaz *et al.* (2003)] Donís Díaz, C. A., Valencia Morales, E., & Morell Pérez, C. (2009). Support vector machine model for regression applied to the estimation of the creep ruptura stress in ferritic steels. Revista Facultad de Ingeniería Universidad de Antioquia, (47), 53-58.

[Douak *et al.* (2013)] Douak, F., Melgani, F., & Benoudjit, N. (2013). Kernel ridge regression with active learning for wind speed prediction. Applied Energy, 103, 328-340.

[Duda *et al.* (2000)] Duda, R. O., Hart, P. E., & Stork, D. G. (2000). Pattern Classification (2nd Edition), Wiley-Interscience.

[Espinosa (2004)] Espinosa, J. (2004). Control Avanzado de Procesos, Una Visión Integral. VI Congreso Colombiano de Automática, 3-19.

[Fan & Xu (2007)] Fan, L., & Xu, Y. (2007). A PCA-Combined Neural Network Software Sensor for SBR Processes. In Advances in Neural Networks-ISNN 2007, 1042-1047.

[Faris & Sheta (2013)] Faris, H., & Sheta, A. (2013). Identification of the tennessee eastman chemical process reactor using genetic programming. International Journal of Advanced Science and Technology, 50, 121-140.

[Ferreira (2001)] Ferreira C. (2001): Gene Expression Programming: A new adaptive algorithm for solving problems. Complex Systems, 13(2), 87-129.

[Fogel *et al.* (1966)] Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial intelligence through simulated evolution, New York: Wiley.

[Fu *et al.* (2007)] Fu, Y., Su, H., & Chu, J. (2007). MIMO soft-sensor model of nutrient content for compound fertilizer based on hybrid modeling technique. Chinese Journal of Chemical Engineering, 15(4), 554-559.

[Gathercole & Ross (1994)] Gathercole, C., & Ross, P. (1994). Dynamic training subset selection for supervised learning in genetic programming. In Parallel Problem Solving from Nature-PPSN III, 312-321.

[Goldberg *et al.* (1991)] Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. Foundations of genetic algorithms, 1, 69-93.

[Gómez & Sanchez (2011)] Gómez, Z., & Sanchez, A. (2011). Sensor virtual neuronal con variables instrumentales y su aplicación en un Convertidor Teniente. Revista Iberoamericana de Automática e Informática Industrial RIAI, 8(1), 54-63.

[Gonçalves & Silva (2013)] Gonçalves, I., & Silva, S. (2013). Balancing learning and overfitting in genetic programming with interleaved sampling of training data, 73-84.

[Gondro & Kinghorn (2008)] Gondro, C., & Kinghorn, B. (2008). Application of evolutionary algorithms to solve complex problems in quantitative genetics and bioinformatics (course notes). Guelph: University of Guelph.

[González (2010)] González, G. D. (2010). Soft Sensing. In Advanced Control and Supervision of Mineral Processing Plants, 143-212.

[Henson & Seborg (1994)] Henson, M. A., & Seborg, D. E. (1994). Adaptive nonlinear control of a pH neutralization process. IEEE Transactions on Control Systems Technology, 2(3), 169-182.

[Hii *et al.* (2011)] Hii, C., Searson, D. P., & Willis, M. (2011). Evolving toxicity models using multigene symbolic regression and multiple objectives. International Journal of Machine Learning and Computing, 1, 30-35.

[Holland (1975)] Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press.

[Ibargüengoytia *et al.* (2013)] Ibargüengoytia, P. H., Delgadillo, M. A., García, U. A., & Reyes, A. (2013). Viscosity virtual sensor to control combustion in fossil fuel power plants. Engineering Applications of Artificial Intelligence, 26(9), 2153-2163.

[Iriondo & Mota (2004)] Iriondo, A., & Mota, J. (2004). Desarrollo de una red neuronal para estimar el oxígeno disuelto en el agua a partir de instrumentación de EDAR. XXV Jornadas de Automática, 8-10.

[Jebari & Madiafi (2011)] Jebari, K., Madiafi, M. (2013). Selection Methods for Genetic Algorithms. International Journal of Emerging Science, 3(4), 333-344.

[Jia *et al.* (2011)] Jia, R. D., Mao, Z. Z., Chang, Y. Q., & Zhao, L. P. (2011). Soft-sensor for copper extraction process in cobalt hydrometallurgy based on adaptive hybrid model. Chemical Engineering Research and Design, 89(6), 722-728.

[Julstrom (1999)] Julstrom, B. A. (1999). It's all the same to me: Revisiting rank-based probabilities and tournaments. In Proceedings of the 1999 IEEE Congress on Evolutionary Computation, 2, 1501-1505.

[Kadlec *et al.* (2009)] Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven soft sensors in the process industry. Computers & Chemical Engineering, 33(4), 795-814.

[Kalman (1960)] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Journal of Fluids Engineering, 82(1), 35-45.

[Kaydani *et al.* (2014)] Kaydani, H., Najafzadeh, M., & Hajizadeh, A. (2014). A new correlation for calculating carbon dioxide minimum miscibility pressure based on multi-gene genetic programming. Journal of Natural Gas Science and Engineering, 21, 625-630.

[Kommenda *et al.* (2013)] Kommenda, M., Affenzeller, M., Kronberger, G., & Winkler, S. M. (2013). Nonlinear Least Squares Optimization of Constants in Symbolic Regression. In Computer Aided Systems Theory-EUROCAST 2013, 420-427.

[Komulainen *et al.* (2006)] Komulainen, T., Pekkala, P., Rantala, A., & Jämsä-Jounela, S. L. (2006). Dynamic modelling of an industrial copper solvent extraction process. Hydrometallurgy, 81(1), 52-61.

[Kordon *et al.* (2003)] Kordon, A., Smits, G., Kalos, A., & Jordaan, E. (2003). Robust soft sensor development using genetic programming. Nature inspired methods in chemometrics, 69-108.

[Kordon *et al.* (2004)] Kordon, A., Jordaan, E., Chew, L., Smits, G., Bruck, T., Haney, K., & Jenings, A. (2004). Biomass inferential sensor based on ensemble of models generated by genetic programming. In Genetic and Evolutionary Computation GECCO, 1078-1089.

[Kotanchek *et al.* (2008)] Kotanchek, M., Smits, G., & Vladislavleva, E. (2008). Trustable symbolic regression models: using ensembles, interval arithmetic and pareto fronts to develop robust and trust-aware models. In Genetic programming theory and practice V, 201-220.

[Koza (1989)] Koza, J. R. (1989). Hierarchical Genetic Algorithms Operating on Populations of Computer Programs. In International Joint Conferences on Artificial Intelligence, 768-774.

[Koza (1992)] Koza, J. R. (1992). Genetic programming: on the programming of computers by means of natural selection. Cambridge, MA: MIT Press.

[Kumar *et al.* (2014)] Kumar, B., Jha, A., Deshpande, V., & Sreenivasulu, G. (2014). Regression model for sediment transport problems using multi-gene symbolic genetic programming. Computers and Electronics in Agriculture, 103, 82-90.

[Langdon *et al.* (2008)] Langdon, W. B., Poli, R., McPhee, N. F., & Koza, J. R. (2008). Genetic programming: An introduction and tutorial, with a survey of techniques and applications. In Computational Intelligence: A Compendium, 927-1028.

[Liu (2007)] Liu, J. (2007). On-line soft sensor for polyethylene process with multiple production grades. Control Engineering Practice, 15(7), 769-778.

[Lopes & Weinert (2004)] Lopes, H. S., & Weinert, W. R. (2004). EGIPSYS: an enhanced gene expression programming approach for symbolic regression problems.International Journal of Applied Mathematics and Computer Science, 14(3), 375-384.

[Luenberger (1966)] Luenberger, D. G. (1966). Observers for multivariable systems. IEEE Transactions on Automatic Control, 11(2), 190-197.

[Luke & Panait (2006)] Luke, S., & Panait, L. (2006). A comparison of bloat control methods for genetic programming. Evolutionary Computation, 14(3), 309-344.

[Mamdani (1977)] Mamdani,E.H. (1977). Application of fuzzy logic to approximate reasoning using linguistic systems. Fuzzy Sets and Systems, 26, 1182-1191.

[Marsland (2009)] Marsland, S. (2011). Machine learning: an algorithmic perspective. New Jersey, CRC Press.

[Martínez & Velásquez (2013)] Martínez, C. A., & Velásquez-Henao, J. D. (2013). Una modificación de la metodología de regresión simbólica para la predicción de series de tiempo. Ingeniería y Universidad, 17(2), 325-338.

[Martínez *et al.* (2011)] Martínez, Y., Trujillo, L., Naredo, E., & Legrand, P. (2014). A comparison of fitness-case sampling methods for symbolic regression with genetic programming. In EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V, 201-212).

[Michalewicz (1996)] Michalewicz, Z. (1996). Heuristic methods for evolutionary computation techniques. Journal of Heuristics, 1(2), 177-206.

[Modirnia & Boulet (2013)] Modirnia, R., & Boulet, B. (2013). Model-Based Virtual Sensors and Core-Temperature Observers in Thermoforming Applications. IEEE Transactions on Industry Applications, 49(2), 721-730.

[Nelles (2001)] Nelles, O. (2001). Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Berlin Heidelberg, Springer-Verlag.

[Noraini & Geraghty (2011)] Noraini, M. R., & Geraghty, J. (2011). Genetic algorithm performance with different selection strategies in solving TSP. In Proceedings of the World Congress on Engineering 2011, 2, London, UK.

[Ogata (1998)] Ogata, K. (1998). Ingeniería de control moderna. Minnesota, PEARSON EDUCACION, 669-843.

[Parasuraman *et al.* (2007)] Parasuraman, K., Elshorbagy, A., & Carey, S. K. (2007). Modelling the dynamics of the evapotranspiration process using genetic programming. Hydrological Sciences Journal, 52(3), 563-578.

[Pereira *et al.* (2011)] Pereira, M. D., Postolache, O., & Girao, P. S. (2011, May). A virtual conductivity sensor for environmental measurements. In IEEE Instrumentation and Measurement Technology Conference (I2MTC), 1-5).

[Pearl (1984)] Pearl, J. 1984. Heuristics: intelligent search strategies for computer problem solving. London, Addison-Wesley Publ. Co.

[Pohlheim (2000)] Pohlheim, I. H. (2000). Genetic and Evolutionary Algorithm Toolbox for Matlab. In Evolutionäre Algorithmen, 157-170.

[Rechenberg (1971)] Rechenberg, I. (1971). Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Stuttgart, Frommann-Holzboog.

[Richter *et al.* (2010)] Richter, T., Oliveira, A. F., & da Silva, I. N. (2010). Virtual oxygen sensor implementation using artificial neural networks. In Technological Developments in Education and Automation, 219-224.

[Schaefer *et al.* (2005)] Schaefer, R., & Hauptmann, P. (2005, September). Acoustic impedance measurement using PLSR based analysis of ultrasonic signals. In 2005 IEEE Ultrasonics Symposium, 1, 178-181.

[Searson *et al.* (2010)] Searson, D. P., Leahy, D. E., & Willis, M. J. (2010, March). GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In Proceedings of the International multiconference of engineers and computer scientists, 1, 77-80.

[Shenghui *et al.* (2011)] Shenghui, P., Chuan, L., Menghe, L., & Lezhu, C. (2011, January). Virtual Sensor for Vehicle Sideslip Angle Based on Extended Kalman Filter. In 2011 Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 3, 1131-1134.

[Sharma & Tambe (2014)] Sharma, S., & Tambe, S. S. (2014). Soft-sensor development for biochemical systems using genetic programming. Biochemical Engineering Journal, 85, 89-100.

[Shawe-Taylor & Cristianini (2004)] Shawe-Taylor, J., & Cristianini, N. (2004). Kernel methods for pattern analysis. Cambridge university press.

[Sliskovic *et al.* (2012)] Sliskovic, D., Grbic, R., & Hocenski, Z. (2012). Methods for plant data-based process modeling in soft-sensor development. AUTOMATIKA: casopis za automatiku, mjerenje, elektroniku, racunarstvo i komunikacije, 52(4), 306-318.

[Smits & Kordon (2008)] Smits, G.F., Kordon, A.K. (2008). Inferential sensors developed using three-dimensional pareto-front genetic programming. U.S., 20100049340, 25 Sep 2008, PCT/US2008/054599, 21 Feb 2008.

[Smola & Schölkopf (2004)] Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and computing, 14(3), 199-222.

[Sucar (2006)] Sucar, L.E. (2006). Redes Bayesianas. BS Araujo, Aprendizaje Automático: conceptos básicos y avanzados, 77-100, Pearson Educación.

[Takagi & Sugeno (1985)] Takagi, T. & M. Sugeno. (1985). Fuzzy identification of systems ans its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, 15(1), 116-132.

[Tianjun & Changfu (2009)] Tianjun, Z., & Changfu, Z. (2009, May). The Road Friction Coefficient Estimation Based on Extended Kalman Filter. In ISA 2009 International Workshop on Intelligent Systems and Applications, 1-4.

[Valencia & Pamies-Teixeira (2013)] Valencia, M. C., & Pamies-Teixeira, J. (2013). Predicción de la rugosidad superficial en texturizados por electroerosión usando redes bayesianas. Sistemas y Telemática, 11(27), 77-92.

[Vapnik (1995)] Vapnik V. (1995). The Nature of Statistical Learning Theory. New York, NY: Springer.

[Webster & Eren (2014)] Webster, J. G., & Eren, H. (Eds.). (2014). Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement. Boca Raton, FL, USA, CRC press.

[Wu & Luo. (2009)] Wu, Y., & Luo, X. (2009). A design of soft sensor based on data fusion. In 2009 International Conference on Information Engineering and Computer Science, 1-4.

[Yong-Hong *et al.* (2012)] Yong-Hong, H., Li-Na, S., & Xin-Lei, S. (2012, December). Soft sensor modeling based on GD-FNN for microbial fermentation process. In IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012), 1-5.

[Yuqiao *et al.* (2006)] Yuqiao, W., Guangxu, C., Haijun, H., & Jieguo, T. (2012, July). A soft sensor for carbon content of spent catalyst in a continuous eforming plant using LSSVM-GA. 31st Chinese Conference In Control (CCC), 7056-7060.

[Zhang & Vagapov (2006)] Zhang, H., & Vagapov, Y. (2006, May). LS-SVM based software sensor for fed-batch yeast fermentation and comparative studies. In 2006 IEEE International Conference on Electro/information Technology, 564-568.

[Zhang *et al.* (2004)] Zhang, Y., Shao, C., & Wu, Q. (2004). RBF neural networks-based software sensor for Aluminum powder granularity distribution measurement. In Advances in Neural Networks-ISNN 2004, 860-865.

[Zadeh (1965)] Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.