



**UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE HIDALGO**

**INSTITUTO DE CIENCIAS BÁSICAS
E INGENIERÍA**

**Centro de Investigación en Tecnologías
de Información y Sistemas (CITIS)**



Tesis Doctoral

**“Una metodología basada en Algoritmos Genéticos Autoadaptables
para la construcción de modelos lineales para Series de Tiempo y
Funciones de Transferencia Discretas”**

Presentada por:

MC. Pedro Flores Pérez

Director de tesis:

Dr. Luis B. Morales

Asesor:

Dr. Julio Waissman Vilanova

Pachuca de Soto, Hidalgo, México, Septiembre 2009

*“Prediction is very difficult,
especially about the future.”*
Niels Henrick David Bohr. (1885-1962)

Dedicatoria:

A mi madre, de quien aprendí a sacar lo mejor de
La Vida en las buenas y en las malas.

A Lupita, la mujer a quien amo y que me ama.

A mis hijos: Piván y Pavel.

Agradecimientos:

Al Dr. Luís Morales quien me enseñó las bondades y beneficios
del trabajo arduo y constante.

A mi gran amigo el MC. Carlos Anaya que me introdujo en el área de
Series de Tiempo y quien aportó importantes ideas para el desarrollo
de este trabajo.

A los Doctores: Joel Suárez, Julio Waissman y Luís Enrique Ramos
por su interés y atinada asesoría durante mis estudios de doctorado.

Contenido

Resumen	iii
----------------------	-----

Capítulo 1.- Introducción

1.1 Descripción de la problemática.....	1
1.2 Antecedentes.....	3
1.3 Objetivos.....	5
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos.....	5
1.4 Principales aportaciones.....	6
1.5 Contenido de la tesis.....	6

Capítulo 2.- Implementación de los Algoritmos Genéticos Autoadaptables para Series de Tiempo.

2.1 Metodología.....	8
2.2 Aspectos básicos de Algoritmos Genéticos.....	11
2.3 Algoritmos Genéticos Autoadaptables.....	13
2.3.1 Cruzamiento.....	14
2.3.2 Mutación.....	15
2.4 Uso de Algoritmos Genéticos Autoadaptables.....	16

Capítulo 3.- Modelos para Series de Tiempo

3.1 Modelos Lineales	19
3.2 Principales resultados de la metodología de Box Jenkins.....	22
3.3 Algoritmos propuestos.....	23
3.3.1 Primer algoritmo.....	24
3.3.2 Segundo algoritmo.....	25
3.4 Comparación con Box Jenkins.....	26
3.5 Resultados de NN3.....	28
3.5.1 Gráficas de NN3-Reduced	29
3.6 Elección de un modelo.....	34
3.6.1 Prueba de bondad de ajuste.....	34

Capítulo 4.- Modelos ARMA para Series de Tiempo

4.1 Construcción de modelos ARMA.....	37
4.2 El fenómeno del <i>retardo en el pronóstico</i> (RP).....	38
4.2.1 El procedimiento de <i>Adelantar el Pronóstico</i>	39
4.3 El origen del <i>retardo en el pronóstico</i>	41
4.4 Comparación con otras metodologías.....	43

Capítulo 5.- Modelos Autorregresivos de Funciones de Transferencia Discretas

5.1 Modelos autorregresivos.....	49
5.1.1 Elección del modelo autorregresivo.....	50
5.2 Ejemplos de modelos lineales para funciones de transferencia discretas.....	51
5.2.1 Primer ejemplo.....	51
5.2.2 Segundo ejemplo.....	53
5.2.3 Tercer ejemplo.....	54

Capítulo 6.- Conclusiones

6.1 Conclusiones.....	58
6.2 Trabajo a futuro.....	59

Bibliografía.....	60
-------------------	----

Anexo 1.- Algoritmo 2G

A. Introducción.....	64
B. Algoritmo 2G.....	65
B.1 Entropía y Ganancia de Información.....	65
B.2 Discretización de valores.....	66
B.3 Selección de atributos.....	67
B.4 Resumen del Algoritmo 2G.....	68
C. Conclusiones.....	69
Referencias bibliográficas.....	70

Anexo 2.- Descripción de algoritmos

A. Algoritmos referenciados en el capítulo 3.....	73
B. Algoritmos referenciados en el capítulo 4.....	77

Resumen

En este trabajo se desarrolla una propuesta metodológica para construir modelos lineales de Series de Tiempo y Funciones de Transferencia Discretas, a partir de plantear el problema de obtener un buen modelo lineal, como resolver un problema de optimización no lineal con variables acotadas. Cabe mencionar que para construir estos problemas se toman algunas ideas del enfoque estadístico tradicional.

Dado que los problemas aquí planteados pueden presentar múltiples mínimos locales, se requiere utilizar una técnica especial para resolverlos. Con este fin se desarrolló una versión de Algoritmos Genéticos Autoadaptables, con codificación sobre números reales que permite, sin intervención del usuario, encontrar soluciones satisfactorias con el mismo código.

Como producto de la metodología aquí presentada, en este trabajo se describen dos algoritmos heurísticos desarrollados para el tratamiento de Series de Tiempo, que permiten construir varios modelos para un mismo problema, donde la precisión de los mismos puede aumentarse incrementando el número de términos del modelo, situación que no ocurre con el enfoque estadístico tradicional. Así, con estos algoritmos, se pueden tener varias propuestas de solución para un mismo problema, de las cuales se puede elegir la que presente mejores resultados en el pronóstico. Además, los algoritmos que se proponen en este documento, permiten construir versiones lineales distintas, pero equivalentes a los modelos Autorregresivos y de Promedios Móviles clásicos, con la ventaja adicional de que se pueden encontrar modelos para Series de Tiempo no estacionarias, y con varianza no estacionaria, en los casos donde la metodología tradicional no funciona. Por otra parte, tomando las ideas básicas de uno de los algoritmos desarrollados para Series de Tiempo, éstas se extendieron para construir un algoritmo con el fin de modelar linealmente Funciones de Transferencia Discretas.

Por otra parte, cabe destacar que entre los principales resultados del trabajo que se describe en este documento, están los hechos de que, en muchos casos, estas versiones lineales presentan un fenómeno que he denominado '*retardo en el pronóstico*', lo que permite modificar el modelo lineal obtenido para obtener un pronóstico más certero. Además, dado que las hipótesis clásicas de los modelos estadísticos no se cumplen, se desarrolló una propuesta para medir la bondad de ajuste de los modelos construidos que permite evaluar cuándo éstos tienen un buen comportamiento lineal.

Capítulo 1

Introducción

En este capítulo se describe el problema central que se aborda en el trabajo de tesis que se ha venido desarrollando, que es el de construir buenos modelos para series de tiempo y funciones de transferencia discretas, con la idea de que a futuro tengan aplicación en el área de control industrial. También se establecen los objetivos que perseguimos y se resumen los resultados obtenidos.

1.1 Descripción de la problemática.

En la mayoría de las plantas industriales de los países desarrollados, los responsables de administrar la producción, tienen dentro de sus tareas principales, la obligación de implementar políticas y procedimientos que permitan contar con información histórica precisa sobre los diversos procesos que ahí se realizan, con el fin de que sirvan para detectar y resolver problemas, mejorar la producción, y/o diseñar sistemas de control para los mismos.

Dicha información histórica normalmente se puede presentar en documentos o dentro de bases de datos agrupadas en tablas, en donde se pueden distinguir tres bloques de información. En el primer bloque de datos, aparece la información de las características más importantes (o que se pueden medir) del material que en cierto momento se está procesando. En el segundo bloque, se presenta la información de la forma en que se está operando en dicho momento el proceso, y en el último bloque, aparecen las características del material producido.

Cabe agregar que, en un proceso industrial, puede haber varias etapas donde las entradas en una etapa son salidas en otra y así sucesivamente. Por esta razón, la etapa más simple de un proceso industrial la representaremos por medio de un diagrama de caja negra, en donde la información señalada en el párrafo anterior se presenta en la entrada de dos tipos de variables con un sólo tipo de salida (ver Figura 1). Los datos en el primer tipo de entradas corresponderían a los valores de las características de los materiales que se están alimentando al proceso, y los del segundo tipo a las variables que representan las condiciones de operación del proceso. Los datos de las entradas son los candidatos a ser modelados por Series de Tiempo (ST).

Una ST es una secuencia ordenada de valores de observaciones de una variable física hecha en intervalos de tiempo igualmente espaciados. A los datos los representaremos por $\{Z_t\}$ con la suposición implícita de que t toma los valores $1, 2, \dots, N$ donde el parámetro N indica hasta qué momento se tiene la información.

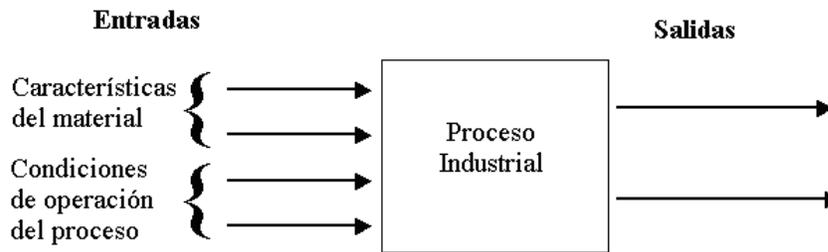


Fig. 1.- Representación de un proceso industrial.

Para realizar el control industrial de un proceso, que consiste en encontrar condiciones de operación que permitan, bajo distintas características de los materiales de entrada, que las características de los productos sean lo mejor posible, es indispensable contar con un modelo que prediga adecuadamente los valores de las salidas en términos de los valores de entrada. Los modelos que describen las respuestas dinámicas de las salidas ante los cambios de las entradas se llaman modelos de funciones de transferencia. Cuando las observaciones de las entradas y las salidas se realizan a intervalos de tiempo igualmente espaciados, se dice que se trata de modelos discretos de funciones de transferencia [4], y para efectos de este trabajo, a las funciones que modelan las salidas en términos de las entradas, las llamaremos Funciones de Transferencia Discretas (FTD).

El término más amplio del análisis de ST abarca actividades como

- a. Descripción: Describir los datos usado un sumario de las estadísticas y/o métodos gráficos. Una grafica de los datos respecto al tiempo es especialmente valiosa.
- b. Modelación: Encontrar un modelo adecuado para describir el proceso de generación de la ST. Un modelo univariado para una variable dada está basado únicamente en los valores pasados de dicha variable.
- c. Pronóstico y Predicción: Pronosticar o predecir los futuros valores de una ST a partir los valores actuales y pasados.
- d. Control: El diseño de esquemas simples que, mediante las desviaciones potenciales de la salida del sistema de un objetivo deseado, puedan, en la medida de lo posible, ser compensada por el ajuste de los valores de entrada de la ST.

En esta tesis se desarrolla una metodología basada en Algoritmos Genéticos Autoadaptables (AGA) para construir modelos lineales de ST y FTD. Los AGA son versiones especiales de los Algoritmos Genéticos (AG) y para nuestro problema son la herramienta ideal para abordarlos, pues permiten resolver distintas instancias de un mismo problema en forma robusta, sin necesidad de realizar ajustes de parámetros, lo que permitirá, a futuro, el desarrollo de un sistema automatizado para resolver estos problemas. Además, se prefirió utilizar estos algoritmos, en lugar de los basados en trayectorias (Búsqueda Tabú, Recocido Simulado, etc.), por su capacidad de explorar más rápida y ampliamente el espacio de soluciones, que es una idea básica de nuestra propuesta metodológica.

1.2 Antecedentes.

Existen diversas metodologías para construir modelos de ST y un panorama sobre las más importantes se puede encontrar en [44]. En [37] se presenta una reseña de las metodologías más utilizadas del área de inteligencia computacional. Una de las metodologías más usadas está basada en considerar a las ST como una realización de un proceso estocástico y este enfoque es la base del tratamiento estadístico de ST que desarrollan Box y Jenkins (BJ) y que se puede consultar en [4] y [17].

Cuando se construyen modelos de BJ se requiere que los residuales sean un ruido blanco, es decir que sean variables independientes con distribución normal. El estadístico de Ljung-Box se utiliza para investigar la independencia de los residuales y nos puede dar información sobre si el modelo requiere más datos o está sobredeterminado. Para estimar si un modelo lineal es adecuado se requiere minimizar algún criterio de información como son el Bayesiano o el de Akaike. Actualmente se utilizan metodologías de optimización que permiten evitar quedar atrapado en mínimos locales lo cual ocurre si se usan las metodologías tradicionales.

Una de las metodologías más utilizadas en el área de ST son los AG. Estos se utilizan para decidir cuál es el modelo más apropiado, de entre una familia establecida de antemano, y que valores tienen los parámetros del modelo. Los AG pertenecen a una familia de algoritmos de optimización llamados Algoritmos Evolutivos (AE), que implementan búsquedas globales en el espacio de soluciones y rápidamente encuentran regiones con muy buenas soluciones. El uso de AE en el área de ST se ha incrementado por el hecho de que permiten construcción de modelos sin quedar atrapados en mínimos locales. En el pasado se utilizaron versiones binarias de estos algoritmos para construir modelos de Holt-Winters [1], y modelos ARMA [6] [20] para ST, sin embargo a últimas fechas se ha despertado un interés en el uso de codificación real [33] por el hecho de que los códigos son más simples, escalables y permiten mayores posibilidades en el uso de operadores genéticos. La versión de AGA que se desarrolla en este trabajo utiliza representación real. No obstante la existencia de artículos donde se utilizan AG en ST [18] [43] [45], es importante señalar que no se encontró ninguna referencia al uso de AGA para este fin.

En [8] [9] [10] los autores construyen modelos ARMA para ST usando una metaheurística basada en AGs. Para esto utilizan un AG en el que cada individuo es un vector binario que define cuáles términos de la parte AR y MA son los más importantes y una vez seleccionados estos, utiliza otro AG para encontrar los valores de los parámetros más adecuados, todos dentro del intervalo (0,1). Como valor de desempeño del individuo utiliza un valor aplicando el Criterio Bayesiano de Información. Cabe señalar que antes de definir el vector binario del individuo se elige primero de qué tipo es tomando en cuenta información visual de la ST.

En [20] se utiliza Programación Evolutiva (PE) para construir modelos ARMA de ST. El procedimiento define primero una población donde cada individuo representa las componentes del modelo ARMA y posteriormente calcula los valores óptimos de los parámetros utilizando también PE. El valor que aquí se utiliza para evaluar el desempeño de un individuo es la Predicción Final del Error de Akaike. No obstante que en este artículo se asegura que esta metodología es de aplicación general, solo presentan resultados para ST que representan el consumo de energía eléctrica, del cual se sabe que es una ST estacionaria y periódica.

En [39] se utiliza programación genética para construir modelos ARMA para ST. Aquí los individuos se representan por dos vectores, uno binario y otro con valores reales. Además, se utiliza el estadístico de Ljung-Box y otras distribuciones estadísticas para medir la

independencia de los residuales y esto se incluye en el establecimiento del valor de desempeño del individuo.

Para la modelación de ST también se han utilizado métodos basados en Wavelets. En [24] se abordan problemas de ST por este medio y además se desarrollan modelos utilizando Programación Genética. Según los resultados aquí presentados Wavelets modela mejor ST estacionarias y periódicas comparado con Programación Genética que al parecer da mejores resultados en ST generales.

Otra de las metodologías utilizadas para modelación y pronóstico de ST son Redes Neuronales (RN). En [42] se aplican optimización por medio de Enjambres de Partículas y Backpropagation para obtener una RN con buenas capacidades de pronóstico. Aquí se utilizan dos funciones, la primera mide la precisión de la RN en el conjunto de entrenamiento y la segunda en el conjunto de pronóstico.

Por otra parte, en el área de pronósticos de ST, las competencias internacionales han jugado un papel muy importante pues gracias a ellas se comparan las metodologías más actuales que se están desarrollando para el tratamiento de ST. Una historia de alguna de las principales competencias (M, M2 y M3) se presenta en [28] donde se resumen los resultados de la competencia M3.

En el 2004 dentro del evento internacional *International Joint Conference on Neural Networks* se realizó la competencia [46]:

“TIME SERIES PREDICTION COMPETITION. The CATS Benchmark”

con el propósito de evaluar modelos de ST que permitieran recuperar información perdida.

En el 2006-2007 se realizó la competencia internacional:

“NN3 Artificial Networks & Computational Intelligence Forecasting Competition”

Que de ahora en adelante llamaremos NN3 [49]. Dicha competencia se realiza anualmente con el fin de evaluar la precisión de métodos del área de Inteligencia Computacional en diversos problemas de ST. En NN3 el problema a tratar era pronosticar con una misma metodología, 18 valores futuros de un conjunto de series donde la mayoría son mediciones de fenómenos reales. La competencia tiene dos categorías: la NN3-Complete que contiene 111 problemas y la NN3-Reduced que consta de 11 problemas. En esta competencia se participó con la metodología desarrollada en esta tesis para construir modelos AR y se obtuvo el tercer lugar en la categoría NN3-Complete de entre 29 concursantes, y el sexto en la categoría NN3-Reduced de entre 53 concursantes. En este trabajo se hará referencia en varias secciones a los ejemplos de la competencia NN3.

Durante el 2008 se realizó la versión de la competencia anterior llamada NN5 [47] con el propósito de evaluar metodologías para modelar ST que simultáneamente tuvieran valores erróneos y valores faltantes. Actualmente está en desarrollo la NGG1 (2009-210) [48] en donde se pretende evaluar metodologías para modelar ST con distintas escalas de tiempo y donde el periodo de pronóstico es variable.

Actualmente la construcción de modelos para ST es un área de gran desarrollo como lo demuestran los artículos del Journal of Time Series Análisis [36].

1.3 Objetivos.

Para desarrollar una metodología que permita construir modelos lineales sean útiles en el área de control industrial, se requiere primero que dicha metodología utilice únicamente la información numérica que proporcionan los datos. Es decir que no se establezcan tratamientos a los problemas de acuerdo a información visual que dependa de los datos.

Otro aspecto importante es que la metodología debe de ser de carácter general y con posibilidades de ser automatizada, es decir debe de poder construirse modelos para la mayor parte de las ST que se presenten en el área de control sin la intervención del usuario.

Por último, los modelos construidos deben de dar buenos valores de pronósticos en el corto plazo pues a partir de esta información es como se definen las estrategias de control. Por todo lo anterior se establecieron los siguientes objetivos para este trabajo de tesis.

1.3.1 Objetivo General.

Desarrollar una propuesta de solución basada en AGA para la construcción de modelos lineales de ST y FTD.

La elección de versiones autoadaptables de AG permitirá que la construcción de los modelos sea susceptible de ser automatizada, pues estas versiones de algoritmos buscan los mejores parámetros y por lo tanto no requieren de la etapa de pruebas que se realiza con este fin.

1.3.2 Objetivos Específicos.

Para cumplir con el objetivo general establecido anteriormente se requiere cumplir con los siguientes objetivos específicos:

- I. Desarrollar una versión de AGA para el tratamiento de problemas generales de optimización no lineales y con variables acotadas.

La versión que aquí desarrolla utiliza variables reales y se construye en base a la experiencia que se ha tenido en una versión anterior desarrollada con codificación binaria y que ya ha sido probada en el tratamiento de problemas complejos.

Dicha versión, además de utilizarse en la construcción de modelos lineales para ST y FTD, también podría utilizarse para resolver problemas de optimización no lineal y con variables acotadas.

- II. Desarrollar una metodología basada en AGA para construir modelos lineales de ST.

Según la revisión bibliográfica realizada, no aparece en la literatura el uso de versiones autoadaptables de AG para la construcción de modelos lineales de ST. El uso de AGA permitirá que la metodología desarrollada sea automatizable, además se utiliza todo el potencial que tienen los AG para encontrar buenas soluciones en espacios que cuentan con múltiples mínimos locales.

- III. Desarrollar una metodología basada en AGA para construir modelos lineales de FTD.

No obstante la importancia de las FTD en los procesos industriales, no se han encontrado referencias sobre propuestas para construcciones robustas de modelos para estas que sean de carácter general. Al parecer se sigue utilizando las versiones tradicionales desarrolladas por Box y Jenkins en sus trabajos iniciales.

1.4 Principales aportaciones.

Para cumplir con los objetivos propuestos se requiere realizar en la tesis lo siguiente:

- El desarrollo de una versión original de AGA que permite encontrar buenas soluciones para problemas de optimización de variables reales, no lineales, con variables acotadas y sin restricciones.
- El desarrollo de algoritmos heurísticos para encontrar modelos Autorregresivos (AR por sus siglas en inglés *Autorregresive*) de ST, que son la base para construir modelos AR y de Promedios Móviles (ARMA por sus siglas en inglés *Autorregresive and Movable Average*) para las mismas.
- El desarrollo de un algoritmo para tratar el caso de FTD con el fin de encontrar modelos lineales para éstas.

Adicionalmente, relacionados con el tema de tesis, se obtuvieron los siguientes resultados adicionales:

- Dado que las técnicas clásicas para medir bondad de ajuste de los modelos lineales, no se aplican a los modelos aquí desarrollados, se construyó una prueba alternativa basada en prueba de hipótesis para comparación de medias y de varianzas.
- El descubrimiento y caracterización de un fenómeno que no aparece reportado en la literatura, al que se ha denominado *retardo en el pronóstico* (RP). Dicho fenómeno permite construir modelos lineales con mayor capacidad de pronóstico comparados con los modelos lineales tradicionales.

1.5 Contenido de la tesis.

Esta tesis consta de seis capítulos y dos anexos. En el capítulo segundo se desarrolla un AGA que será la base para la construcción de modelos lineales para ST y FTD. Cabe señalar que la versión de AGA que se presenta tiene posibilidades de aplicación más generales, pues permite

resolver problemas de optimización de variables reales, sin restricciones, no lineales y con variables acotadas.

En el capítulo tercero se presenta la construcción de modelos AR para las ST. Además, aquí aparece una primera comparación de los modelos desarrollados con esta nueva metodología con los reportados por Box-Jenkins (BJ). También se comentan los resultados obtenidos en la competencia NN3, y en la última sección de este capítulo se propone una prueba de bondad de ajuste alternativa a las tradicionales, para evaluar el desempeño de los modelos lineales obtenidos para ST y FTD con esta metodología.

En el capítulo cuarto, se agregará a estos modelos una componente autorregresiva de los residuales, lo que permitirá construir modelos equivalentes a los ARMA tradicionales. Además, aquí se estudia el fenómeno del RP que permitirá la definición de nuevos modelos para ST, que mejoran substancialmente la capacidad de pronóstico. Además, en este capítulo se compara la capacidad de predicción de los modelos obtenidos para ST con esta metodología, con otros modelos que son clásicos de la literatura de pronósticos.

En el capítulo quinto, se desarrolla la parte correspondiente a la construcción de modelos lineales de FTD. Para esto se extiende naturalmente la aplicación a FTD de uno de los algoritmos desarrollados para ST y, para evaluar la calidad de los resultados obtenidos, se compara la precisión de los modelos aquí desarrollados para FTD con un método que aparece en una publicación reciente. En el capítulo sexto aparecen las conclusiones del presente trabajo, y por último, en el anexo 1, se presenta una descripción del algoritmo 2G que fue desarrollado en colaboración con Maria de Guadalupe Cota Ortiz, el cual está basado en reconocimiento de patrones y árboles de decisión para resolver problemas de clasificación, y se utiliza para caracterizar el fenómeno del RP, y en el anexo 2 se presenta una descripción del funcionamiento de los códigos de los algoritmos más importantes de esta tesis.

Capítulo 2

Implementación de los Algoritmos Genéticos Autoadaptables para Series de Tiempo.

La herramienta principal que se utiliza para tratar los problemas de optimización que se describe en este trabajo son los AGA. En este capítulo presentamos la propuesta de metodología que utilizaremos para construir los modelos lineales de este trabajo.

Posteriormente se mencionan a grandes rasgos las características de los AG para pasar a presentar las versiones autoadaptables de dichos algoritmos, que permiten la resolución de diversos problemas de optimización, minimizando la etapa de ajuste de parámetros. Por último, se presentan las distintas etapas en que se aplican estos algoritmos para encontrar los modelos lineales de ST y FTD.

2.1. Metodología.

El proceso de pronosticar consiste en calcular o predecir el valor de algún evento que va a ocurrir en el futuro. Para realizar adecuadamente este proceso se requiere analizar los datos del evento de que se trate y construir un modelo que permita incorporar los patrones del comportamiento que han ocurrido en el pasado bajo el supuesto de que pueden volver a ocurrir en el futuro. Es importante señalar que no se está interesado en explicar cómo funciona el mecanismo que produce los eventos, sino de predecir su comportamiento.

Dado que el futuro es incierto, el pronóstico no puede ser perfecto, por lo cual el objetivo principal de los métodos de pronóstico es reducir el error del mismo. Con este fin se han desarrollado múltiples métodos basados en modelos matemáticos y en particular se está interesado en los modelos deterministas en donde la relación entre la variable de interés F y las variables de predicción Z_1, Z_2, \dots, Z_p está dada por una expresión de la forma

$$F = f(Z_1, Z_2, \dots, Z_p, \beta_1, \beta_2, \dots, \beta_m)$$

Donde la estructura de la función f y los parámetros $\beta_1, \beta_2, \dots, \beta_m$ son conocidos con certeza.

En este trabajo, la variable F representará el valor en un cierto instante t de una ST y las variables Z_1, Z_2, \dots, Z_p serán los valores que ha tomado la serie junto con los errores obtenidos con el pronóstico del modelo hasta antes del momento t y se utilizarán funciones f lineales.

Como se está interesado en desarrollar modelos que sean de utilidad en el área de control industrial es muy importante que el proceso de construcción del modelo pueda ser totalmente automatizable, es decir que no utilice información obtenida por medios visuales sobre el comportamiento de la serie obtenida. Además, los modelos desarrollados deben tener buen desempeño durante el horizonte de pronóstico el cual generalmente es corto.

Cuando se tiene un modelo para el conjunto de datos entonces se pueden estimar valores para la ST, que representaremos por $\{F_t\}$, y lo que uno requiere de un buen modelo es que los valores de $\{F_t\}$ sean “parecidos” a los de $\{Z_t\}$. El propósito principal de este trabajo es construir modelos lineales para el conjunto de datos de tal manera que se tengan buenas estimaciones de los K valores desconocidos del fenómeno que se está estudiando en los momentos $N+1, N+2, \dots, N+K$.

En el área de pronósticos, cuando se tiene una ST con estos $N+K$ datos, al conjunto de los primeros N se le llama conjunto de entrenamiento, y se usa para construir el modelo de la serie y realizar la estimación de los parámetros del mismo. Al conjunto de los últimos K términos de la serie, se le denomina conjunto de prueba, y se utiliza para comparación de distintos modelos con el fin de elegir el más adecuado.

En particular, se está interesado en la construcción automatizada de modelos autorregresivos de orden p (AR(p)). Para las ST son expresiones de la forma:

$$Z_t = \delta + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t \quad (1)$$

Donde Z_t es la variable observable de que se trata, δ y ϕ_j son los parámetros a determinar y la variable a_t representa una variable aleatoria (llamada *residual*) de ruido. La expresión (1) significa que para predecir lo que va a ocurrir en el momento t se requieren los p valores anteriores a t , a estos valores se les denomina *retrasos* o *retardos*.

En la teoría clásica de modelos lineales, se establece la restricción de que a_t representa un ruido blanco, pero en este trabajo no incluimos esta limitante, lo que permitirá encontrar expresiones AR para los residuales con las que se podrá aumentar la precisión de los modelos.

El interés en este tipo de modelos, se origina en el hecho de que representan la información más importante sobre el comportamiento de la serie eliminando el “ruido” que pudiera aparecer. Además, cabe agregar que, para estos modelos, es importante que en la expresión (1) sólo aparezca cierto número de términos establecido de antemano. Esto último permitirá encontrar modelos para una ST, controlando la precisión de la aproximación de la misma de acuerdo al número de términos que se utilicen.

Para construir estos modelos se plantea inicialmente el siguiente problema de optimización:

Problema 1: Si $\{Z_t\}$ es la ST original y $\{F_t\}$ es el pronóstico obtenido de la forma

$$F_t = \delta + \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} \quad \text{para } t > p$$

y (2)

$$F_t = Z_t \quad \text{para } t \leq p$$

Requerimos encontrar valores para δ y ϕ_i que minimizan la función:

$$RSS = \sqrt{\sum_{t=0}^N (Z_t - F_t)^2} \quad (3)$$

A esta función la denominaremos RSS (por sus siglas en inglés *Root of the Sum of Squares*).

En este planteamiento inicial, el construir el modelo se presenta como resolver un problema de interpolación lineal, pero dado que los valores para δ y ϕ_i no serán arbitrarios, sino que se buscarán en ciertos intervalos, es necesario desarrollar métodos de solución que resuelvan el **Problema 1**, trabajando además con variables acotadas.

La elección de (3) está motivada por el hecho de que, en el conjunto de entrenamiento, el modelo debe de tener un comportamiento muy parecido al del fenómeno que se está estudiando. Cabe agregar que por rapidez en cómputo es preferible usar el cuadrado de esta función obteniéndose los mismos resultados. Por otra parte, uno de los aspectos más importantes de este trabajo, es que se pueden utilizar otras funciones siguiendo la metodología aquí presentada.

De la función RSS esta reportado [23] que puede tener múltiples óptimos locales (ver gráfica 12), y para resolver este problema, se desarrolla una versión original de Algoritmos AGA, que permite resolver problemas de optimización reales no-lineales y con variables acotadas. La selección de una versión autoadaptable, se realizó por el hecho de que se desea automatizar lo más posible el proceso de construcción de estos modelos, con posibilidades futuras de aplicarse en control industrial.

En este trabajo también se construyen modelos lineales para FTD. Las FTD son modelos matemáticos discretos de la respuesta de un sistema ante los cambios de los valores de las variables de entrada del mismo. Para efectos de esta tesis, se está interesado en construir modelos de la variable de salida como una suma de expresiones autorregresivas de las series de las variables de entrada y también de la misma serie de salida.

Para lo anterior se utiliza una metodología semejante a la del tratamiento de ST, eligiendo de antemano cuántos términos se van a considerar para todas las series involucradas, y aplicando un algoritmo basado en uno de los algoritmos heurísticos construidos para ST, encontramos modelos lineales para las FTD.

El algoritmo que desarrollamos para FTD construye varias soluciones distintas, pero con valores bastante aproximados del error. Para elegir la solución más adecuada, se toma en cuenta la importancia de las variables de entrada, con el fin de predecir los valores de la salida a partir de los coeficientes de correlación entre estas variables. Para esto se elige la solución en donde aparezca un mayor número de términos en las expresiones de las variables más correlacionadas.

2.2. Aspectos básicos de Algoritmos Genéticos.

Los AG fueron desarrollados por John Holland [19], y están inspirados en los principios básicos que rigen la evolución de las especies. Han sido utilizados exitosamente en la resolución de muchos problemas, y según se señala en [33], para aplicarlos, se requiere:

- 1) Una representación genética del individuo.
- 2) Alguna manera de crear una población inicial.
- 3) Una función de desempeño que revise qué tan adaptado al medio se encuentra un individuo y lo separe, por un procedimiento llamado selección, según su desempeño.
- 4) Los operadores genéticos que alteran la estructura de los hijos. Normalmente se consideran el cruzamiento y la mutación.
- 5) Los valores de varios parámetros que son requeridos por los Algoritmos Genéticos como son: tamaño de población, probabilidad de cruzamiento y probabilidad de mutación.

En [16] [33] se presentan los aspectos básicos relacionados con los AG y las técnicas para los ajustes de los parámetros que se señalan en 5). En la Figura 2, se presenta un diagrama para la implementación de un AG tradicional.

A continuación veremos los detalles de la implementación de nuestra versión de AGA, siguiendo los puntos que aparecen en la sección anterior.

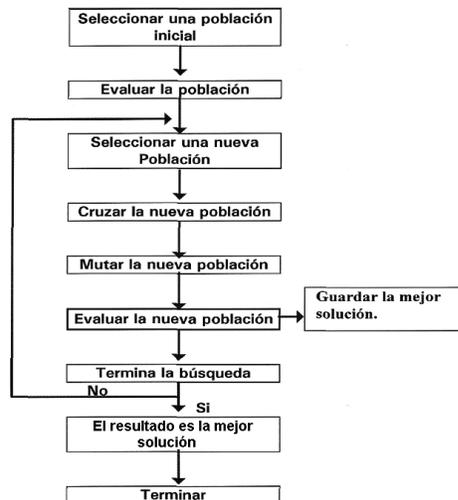


Fig. 2.- Operación de los algoritmos genéticos.

En este trabajo los individuos se representan por vectores de coordenadas reales que tienen dos partes. En la primera parte, aparecen los coeficientes de la expresión lineal (2) del modelo AR para la ST, y en la segunda, los cuatro parámetros autoadaptables que aquí se utilizan y que de momento denotaremos por *. En (7) se presenta la descripción de estos parámetros. Para la representación de los individuos, se utiliza la metodología para especificación de números reales que aparece en el capítulo 5, intitulado “Binary or Float” de [33], en donde además se establece la precisión numérica de la misma.

$$(\delta, \phi_1, \phi_2, \dots, \phi_p, *, *, *, *) \quad (4)$$

Los individuos de la población inicial se crean aleatoriamente generando uniformemente sus coordenadas entre los valores de los límites establecidos de antemano para cada coordenada. El establecimiento de los valores de los límites para las variables de (4) se hará en el capítulo tercero.

Para evaluar el desempeño de un individuo, se construye la expresión lineal que resulta al tomar los coeficientes que aparecen en la primera parte del vector que representa al individuo (4), y con dicha expresión, se genera un vector de pronósticos para $\{F_t\}$, de la misma longitud que la serie original, donde las coordenadas de dicho vector son los valores obtenidos por la expresión lineal construida (2). Con estos valores se estiman los errores en cada coordenada y la función que se presenta en (3), de tal forma que con este último valor se cuantifica el desempeño de un individuo. A continuación presentamos un ejemplo de este procedimiento.

Suponiendo que para una cierta serie $\{Z_t\}$ se generó al individuo:

$$(5.32, 0.47, 0.0, 0.0, 0.89, 0.0, 0.0, 0.56, 2.3, 0.04, 1.7) \quad (5)$$

Como se indica en (4) este individuo tiene dos partes las cuales son:

$$(5.32, 0.47, 0.0, 0.0, 0.89, 0.0, 0.0) \quad (0.56, 2.3, 0.04, 1.7)$$

En las primeras siete componentes del vector (5) aparecen los coeficientes del modelo AR

$$F_t = 5.32 + 0.47Z_{t-1} + 0.89Z_{t-4} \quad (6)$$

y las últimas cuatro componentes del vector (5) aparecen los valores de los parámetros de los algoritmos genéticos autoadaptables que serán definidos en la Sección 2.3.

$$(0.56, 2.3, 0.04, 1.7)$$

Con (6) se generan los valores de la serie de pronósticos $\{F_t\}$ y se calcula el valor de RSS dado por la ecuación (3), asignando este valor de desempeño al individuo (5).

El procedimiento de selección que se utilizará es el de k -torneos [34] con valor de k igual a 2. Para esto, en cada momento de selección se eligen dos individuos al azar, y de entre ellos, se selecciona al que tenga mejor valor de función de desempeño. Además, se implementa el elitismo, que consiste en asegurar que siempre existe, en cada iteración, un individuo igual al Mejor Individuo (MI) obtenido durante todo el proceso.

Uno de los problemas que se pueden presentar en la aplicación de los AG, consiste en la aparición excesiva de copias del MI en alguna iteración, pues esto puede producir el fenómeno llamado Convergencia Prematura, que consiste en que el algoritmo converge muy pronto a soluciones que no son muy buenas. Con el fin de evitar el problema de Convergencia Prematura, en esta propuesta se modifica el procedimiento de selección, para que en cada población de 100 individuos sólo pueda haber diez copias de MI, para esto en cada etapa de selección se lleva un conteo de cuantos individuos son iguales a MI, y cuando se llega a diez y vuelve a aparecer un MI en una selección, se realiza otro torneo con otros individuos hasta que se obtenga un individuo distinto de MI.

De todos los parámetros utilizados en los AGA, para nuestra versión sólo se requiere establecer el tamaño de población (que se fija en 100 individuos), pues los parámetros relacionados con cruzamiento y mutación serán autoadaptables.

2.3. Algoritmos Genéticos Autoadaptables.

Los Algoritmos AGA fueron desarrollados por Thomas Back [2] [3] y tienen la característica de que ellos solos buscan los mejores parámetros para su funcionamiento. En ellos, los parámetros que serán autoadaptables se codifican en la representación del individuo, por lo cual son alterados por las acciones de los operadores genéticos. Con esto, los mejores valores de estos parámetros producirán mejores individuos, los cuales tienen mayor probabilidad de sobrevivir, y en consecuencia, propagarán hacia toda la población los mejores valores de los parámetros. Existen diversas versiones de AG autoadaptables que difieren sobre todo en los parámetros que van a ser adaptados automáticamente [12].

La principal desventaja que tiene la utilización de AGA, es el mayor costo computacional comparado con las versiones tradicionales, pero la ventaja que obtenemos, es que con el mismo código se puede resolver automáticamente una gran cantidad de problemas, sin necesidad de ajustar parámetros para cada problema en particular. En nuestro caso utilizamos cuatro parámetros autoadaptables: *probabilidad individual de cruzamiento* p_c , *repetición de cruzamiento* r_c , *probabilidad individual de mutación* p_m y *repetición de mutación* r_m como se presenta en (4). La selección de estos parámetros y sus valores, está basada en la idea de que las operaciones genéticas de cruzamiento y mutación pueden ser múltiples, pero no pueden tener valores muy grandes. Una versión binaria de éste algoritmo ya había sido utilizada anteriormente por el autor de esta tesis en otros problemas [14] [15] [40], y ésta, así como la presentada aquí (con representación sobre números reales), según la revisión bibliográfica realizada, son originales de él mismo.

En este trabajo todos los problemas serán problemas de optimización de funciones no lineales de varias variables reales, donde las variables tomarán valores en un intervalo conocido de antemano. Uno de los objetivos principales aquí establecidos, es que estos problemas puedan ser resueltos con el mismo código, donde lo único que se cambiará son las funciones a optimizar.

Los individuos para estos problemas serán propuestas de solución para los mismos, y adicionalmente tendrán cuatro componentes más, donde se representarán los valores de:

probabilidad individual de cruzamiento p_c , repetición de cruzamiento r_c , probabilidad individual de mutación p_m y repetición de mutación r_m . A esta sección del individuo se le denominara *sección de los parámetros autoadaptables*, y con esto, nuestro individuo completo se representa por

$$(\delta, \phi_1, \phi_2, \dots, \phi_p, p_c, r_c, p_m, r_m) \quad (7)$$

Lo anterior es necesario, pues en este modelo, la probabilidad de cruzamiento y de mutación serán características de cada individuo (no de la población como es tradicional en los AG), y además se considera que el cruzamiento y la mutación pueden ser múltiples, es decir, actuar varias veces en el mismo momento. El cruzamiento y la mutación múltiple, son repeticiones del cruzamiento y la mutación que se utiliza en los AG, cuando se utilizan individuos representados por vectores de componentes reales. La forma de operar con estos parámetros es semejante a la presentada en [2] [3].

Los límites que se usaron en nuestro código para los parámetros autoadaptables son: probabilidad individual de cruzamiento p_c que varía en el intervalo (0.5, 0.95), repetición de cruzamiento r_c en (1.0, 4.0) lo que significa que solo se puede cruzar de una a tres veces, probabilidad individual de mutación p_m que varía en (0.5, 0.85) y repetición de mutación r_m en (1.0, 5.0) lo que significa que se puede mutar de una a cuatro veces. Los límites de estos parámetros autoadaptables se eligieron en base a la experiencia de otros trabajos [14] [15] [40], en donde probaron que daban buenos resultados.

A continuación se detallan los procedimientos de cruzamiento y mutación.

2.3.1. Cruzamiento.

Para el cruzamiento, dados dos individuos, se realiza el cruzamiento tomando como probabilidad de cruzamiento el promedio de los valores de los cruzamientos individuales. Una vez que se ha decidido si se cruzan los individuos, se toma la parte entera del promedio de los cruzamientos individuales, y ese es el número de veces que se cruzan. El cruzamiento de dos individuos consiste en intercambiar las coordenadas de los dos vectores a partir de cierta coordenada elegida al azar. El cruzamiento múltiple resulta de aplicar este procedimiento varias veces a los mismos vectores. A continuación veremos un ejemplo de este procedimiento:

Suponiendo que se tienen a los individuos:

$$(\delta_1, \phi_{11}, \phi_{21}, \dots, \phi_{p1}, p_{c1}, r_{c1}, p_{m1}, r_{m1})$$

y

$$(\delta_2, \phi_{12}, \phi_{22}, \dots, \phi_{p2}, p_{c2}, r_{c2}, p_{m2}, r_{m2})$$

Se genera un número aleatorio en (0,1) y si es menor que $(p_{c1} + p_{c2})/2$ se cruzaran los dos individuos. Supongamos que este es el caso. Ahora se calcula el promedio de los valores de la repetición de cruzamiento.

$$(r_{c1} + r_{c2})/2$$

Tomándose la parte entera de este. Supongamos que el valor obtenido sea el numero 2. Ahora se eligen dos números entre uno y el número de coordenadas menos uno y en esas posiciones se cruzan los vectores. Supongamos que estos dos números fueron uno y tres. Después de cruzar en estos lugares los vectores resultantes serian:

$$(\delta_1 |, \phi_{12}, \phi_{21} |, \dots, \phi_{p1}, p_{c1}, r_{c1}, p_{m1}, r_{m1})$$

y

$$(\delta_2 |, \phi_{11}, \phi_{22} |, \dots, \phi_{p2}, p_{c2}, r_{c2}, p_{m2}, r_{m2})$$

El símbolo '|' señala los lugares uno y tres. A continuación veremos cómo opera la mutación.

2.3.2 Mutación.

Para la mutación se toma la probabilidad individual de mutación del individuo, y de acuerdo a esto se decide si se muta o no. Una vez que se ha decidido que se muta un individuo, se muta el mismo tantas veces como el valor de la parte entera del valor que tiene en la repetición de mutación del mismo. Para aplicar la mutación a un individuo, se elige una coordenada del vector al azar, y se cambia su valor por otro (elegido también al azar) entre los límites establecidos para dicha coordenada. La mutación múltiple es la aplicación de este procedimiento al mismo individuo varias veces. A continuación veremos un ejemplo del procedimiento de mutación.

Para aplicar el procedimiento al individuo:

$$(\delta, \phi_1, \phi_2, \dots, \phi_p, p_c, r_c, p_m, r_m)$$

Se genera un número aleatorio en (0,1), y si es menor que p_m , entonces se muta el individuo, suponiendo que éste es el caso. Ahora se toma el valor de repetición de mutación del individuo que es r_m y de este valor tomamos la parte entera, supongamos que este valor es 2. Para mutar el individuo se eligen dos coordenadas al azar y se cambian los valores entre los límites establecidos para estas coordenadas. Por ejemplo, si las coordenadas elegidas son la tercera y la penúltima, el individuo ya mutado seria

$$(\delta, \phi_1, \underline{\phi_2}, \dots, \phi_p, p_c, r_c, \underline{p_m}, r_m)$$

donde símbolo ϕ_m y p_m son los nuevos valores mutados. A continuación veremos la forma en que utilizamos los SAGA.

2.4. Uso de los Algoritmos Genéticos Autoadaptables.

Dado que los AG son aleatorios, es común realizar varias ejecuciones sobre el mismo problema y elegir el mejor resultado de éstas. Estos algoritmos se aplicarán en tres etapas para resolver nuestros problemas. En las primeras dos etapas, se definen cuales son las variables importantes para resolver nuestro problema, y en la tercera etapa, es donde propiamente se calcula la solución. Cabe recordar que nuestros individuos tienen dos partes, pero en esta sección solo se toma en cuenta la primera parte del individuo, que es la que corresponde a las componentes autorregresivas. A continuación presentamos el procedimiento basado en AGA, que se realiza para obtener una solución al problema.

En la primera etapa, se usan AGA para explorar el espacio de soluciones y posteriormente definir qué variables, de entre $\delta, \phi_1, \phi_2, \dots, \phi_p$, son las más importantes para el problema de que se trate. Para esto se realizan 10 repeticiones de 1000 iteraciones cada una, y con las soluciones de cada repetición, se construye un vector con la suma de los valores absolutos de $\delta, \phi_1, \phi_2, \dots, \phi_p$. (ver Figura 3).



Fig. 3.- Solución usando todas las variables.

En esta primera etapa, lo que se pretende es realizar una exploración del espacio de soluciones, y para esto se realizan las 10 iteraciones con todas las variables a considerar. Después, con las 10 soluciones obtenidas, se construye un vector al sumar las 10 soluciones con todas sus componentes positivas, y suponemos que los valores más grandes de estas componentes son los más importantes. A continuación presentamos un ejemplo de lo antes expuesto:

Suponiendo que la primera parte del vector que resulta de la primera etapa es el:

$$(54.23, 5.978, 7.025, 1.213, 3.56) \quad (8)$$

En la segunda etapa se aplican los AGA para encontrar soluciones tomando en cuenta sólo las variables importantes del problema. Para esto se define de antemano cuántas variables se requieren (esto se verá a detalle en el siguiente capítulo), y se eligen las que correspondan a valores más grandes de la primera etapa. En esta etapa se realizan 5 repeticiones, donde se termina cada una, hasta que no se modifica el óptimo en las últimas 200 iteraciones. De estas 5 repeticiones se elige el mejor resultado obtenido (ver Figura 4).



Fig. 4.- Solución con las variables más importantes.

En esta segunda etapa, solo se consideran las variables que tuvieron valores más grandes en la parte de las componentes autorregresivas del individuo, y para ellas se mantienen los intervalos originales de sus valores: Para todas las demás variables en la parte autorregresiva del individuo, se establece que los límites superior e inferior son cero. En esta etapa se realizan 5 repeticiones y de ellas se elige la que menor valor de RSS tenga.

Si se desea encontrar soluciones que tengan tres componentes a partir de (8), entonces para la segunda etapa se construyen individuos con las componentes cuatro y cinco iguales a cero.

Suponiendo que después de la primera etapa se obtuvo un individuo cuya primera parte es:

$$(5.23, 0.978, 0.005, 0.0, 0.0) \quad (9)$$

En la tercera etapa se encontrará la solución en donde solo se toman en cuenta las variables importantes obtenidas en la etapa anterior. Para esto se amplían los límites de las variables de la solución obtenida en la etapa anterior, cuyo valor absoluto se mayor que 0.01. Los límites superiores de las variables consideradas, serán los valores distintos de cero obtenidos en la mejor solución anterior de más de 1.0 y los inferiores con menos de 1.0. Los límites superiores e inferiores de las demás variables de las componentes autorregresivas del individuo serán cero. Con estos límites se resuelve una sola vez hasta que en 250 iteraciones no se mejora el óptimo.

Una posible solución obtenida a partir de (9) en esta etapa podría ser el vector (10), en donde la tercera componente aparece como 0.0 por ser menor que 0.01. Lo anterior significa que aunque se pida que el modelo tenga tres componentes, pudiera darse el caso de que tuviera menos debido a la eliminación de los valores muy pequeños.

$$(6.21, 1.55, 0.0, 0.0, 0.0) \quad (10)$$

Dado que los AG son aleatorios, para cada problema se realizaron 5 iteraciones, y de ellas se eligió la mejor.

Las principales características de la versión de AGA que utilizamos y que la hacen original son:

- Se utiliza una codificación real para las variables del problema. Esto permite que el código sea más simple y que fácilmente se pueda pasar de una etapa a otra de las aquí presentadas.
- Las probabilidades de cruzamiento y mutación son características de cada individuo y los procedimientos de cruzamiento y mutación se establecen en base a estas características individuales.
- Las repeticiones de cruzamiento y mutación son múltiples aunque los valores que toman no son muy grandes.
- Se introduce un mecanismo de control que evita la proliferación dentro de la población de copias del mejor individuo eliminando con esto el riesgo de convergencia prematura.

Sobre todo, las tres últimas características están inspiradas en el hecho de que la naturaleza tiene un comportamiento más flexible que rígido, y por lo tanto, debe de permitirse más la variabilidad dentro de los AGA.

La principal desventaja que tiene la utilización de las versiones autoadaptables es el mayor costo computacional comparado con las versiones tradicionales, pero la ganancia que se obtiene, es que con el mismo código se pueden resolver automáticamente todos los problemas de modelación lineal de ST. Además, es posible aplicarlos en un mismo problema para determinar las variables más importantes del modelo de un sistema, para posteriormente, incluir esta información en la solución de los modelos con el fin de disminuir el costo computacional.

En el siguiente capítulo, se presentan dos versiones de algoritmos heurísticos que se desarrollaron en esta tesis para resolver nuestro problema, lo que significa que para cada versión se realiza el procedimiento presentado en esta sección, y se elige la mejor solución de entre las obtenidas con estas dos versiones.

Capítulo 3

Modelos para Series de Tiempo

En este capítulo se presentan las ideas principales en las que se basa el desarrollo de los algoritmos heurísticos que permiten la construcción de los modelos lineales de las ST. Cabe señalar que aquí sólo se considera la construcción y ajuste de parámetros de los modelos, basados en la información del conjunto de entrenamiento. En el capítulo siguiente se presentará la parte de la selección de un modelo tomando en cuenta la precisión en el pronóstico sobre el conjunto de prueba.

En la primera parte de este apartado, se presentan las características de los modelos lineales, en particular del modelo autorregresivo, AR, pues a partir de ellos construimos todos nuestros modelos lineales. Posteriormente se resumen los principales resultados de la metodología de Box y Jenkins (BJ), pues de ahí se toman las ideas principales de este trabajo. En la tercera sección, se presentan los dos algoritmos heurísticos desarrollados que básicamente difieren en la forma en que realizan la estimación de los límites de los parámetros que aparecen en (1). Posteriormente, se realiza una comparación con los resultados obtenidos por BJ con los que son producto de esta metodología. En la quinta sección se presentan los resultados de NN3, y en la sexta, la propuesta de prueba de bondad de ajuste propuesta para los modelos.

3.1 Modelos Lineales.

Los modelos lineales de ST son importantes porque existen numerosas aplicaciones donde las estimaciones lineales son suficientes, además de que tienen un amplio uso en situaciones industriales. Por otra parte, también son importantes porque existen otras metodologías para pronósticos que los utilizan [31] [32]. En esta sección daremos a algunas clases de modelos lineales univariados, como son los Autorregresivos AR (Autorregressive), los de media móvil MA (moving average), los de Autorregresivos de Media Móvil ARMA (Autorregressive Moving Average) y los Autorregresivos Integrados de Media Móvil ARIMA (AutoRegressive Integrated Moving Average processes). Las ST univariadas fueron analizadas por BJ a partir de la construcción de ecuaciones en diferencias con un componente aditivo aleatorio denominado ruido blanco. Para dichos modelos se estudiaron las condiciones bajo las cuales se presenta la propiedad de estacionariedad de la serie, y se determinó el esquema que debe seguirse para estimar los parámetros del modelo particular. La referencia más clásica para el tratamiento de modelos lineales es el libro de Box y Jenkins

En el caso específico de los modelos AR que nos interesan para las ST, el valor en un cierto instante de tiempo debe calcularse como una expresión lineal de los valores de un cierto número de mediciones anteriores, según se señala en (1). Los modelos AR que aquí se desarrollan cumplen con que el proceso estocástico de los residuales $\{a_t\}$ asociado a ellos, no es un ruido blanco. Esto último permitirá que, una vez que se construye un buen modelo AR de una ST, se

les pueda construir otro modelo AR para los residuales $\{a_t\}$, que aunado al original, permita obtener el equivalente de un modelo ARMA, pero con mayores posibilidades de pronóstico.

Por otra parte, para resolver el **problema 1** planteado en el capítulo 2, es necesario primero resolver las siguientes preguntas, tomando en cuenta que queremos encontrar un modelo AR con K términos donde K se establece de antemano:

- 1) ¿Cuántos términos p es necesario considerar?
- 2) ¿En qué intervalos están los coeficientes de la expresión lineal?
- 3) ¿Cuáles términos K son los más apropiados para resolver este problema?
- 4) ¿Cuáles son los valores de estos K términos que minimizan la función (3)?

Para el tratamiento de los Modelos $AR(p)$ una común herramienta de la estadística para las ST es la auto correlación (AR), denotada por ρ_k , la cual da la medida de la correlación para los datos de la serie separados k unidades en el tiempo. La idea detrás de los procesos de AR es alimentar con los datos del pasado al valor actual del proceso. Esto induce una correlación entre el pasado y el presente. El efecto es tener al menos alguna correlación en todos los retardos. Si la auto correlación es calculada para cada todos los valores de k , entonces se obtiene la función de auto correlación, dada por

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \dots + \phi_p \rho_{k-p}$$

La función de auto correlación es un instrumento matemático útil para encontrar patrones repetidos, y puede ser de utilidad para la descomposición de las componentes principales de la ST. De las condiciones iniciales que satisfaga esta función se presentan los siguientes posibles comportamientos: decaimiento exponencial o decaimiento senoidal. Lo anterior permite determinar si una serie es estacionaria o no.

En la práctica, de una ST finita Z_1, Z_2, \dots, Z_N de N observaciones, solamente se puede obtener una estimación de la k -ésima correlación ρ_k como

$$\rho_k = \frac{\sum_{i=1}^{N-k} (Z_i - z)(Z_{i+k} - z)}{\sum_{i=1}^{N-k} (Z_i - z)^2}$$

donde z denota el promedio de los valores de la serie.

Una herramienta útil para analizar series $\{Z_t\}$, es el uso del operador de desplazamiento hacia atrás B (backwards shift operator), el cual es definido por $BZ_t = Z_{t-1}$; en consecuencia $B^k Z_t = Z_{t-k}$. Otro importante operador es el operador diferencia hacia atrás ∇ , definido por $\nabla^d Z_t = Z_t - Z_{t-d}$. Entonces estos dos operadores satisfacen

$$\nabla^d Z_t = Z_t - Z_{t-d} = (1-B)^d Z_t$$

Ahora se define un operador ϕ autorregresivo de orden p en términos del operador de desplazamiento hacia atrás B de la forma siguiente

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (11)$$

Note que la función ϕ es un polinomio en B de orden p .

Ahora analizaremos los procesos $AR(p)$ por medio del operador ϕ . Usando este operador el modelo $AR(p)$ puede ser escrito de forma concisa como

$$\phi(B)Z_t = a_t \quad (12)$$

El modelo autorregresivo $AR(p)$ puede ser estacionario o no. La condición necesaria para el modelo sea estacionario es que el valor absoluto de todas las raíces ($\phi(x)=0$) del polinomio $\phi(B)$ sea mayor que 1 en valor absoluto, es decir, todas sus raíces deben estar fuera del círculo unitario de los complejos. Otras propiedades del proceso $AR(p)$ definido por (12) pueden ser examinadas al analizar las propiedades de este polinomio, ver por ejemplo [4] [7].

El Modelo $MA(q)$. En algunas ST los datos muestran una correlación sólo a corto plazo, por ejemplo, sólo en el retardo 1 o sólo en los retardos 1 y 2. Los procesos AR no se comportan de esta manera y no se ajustan adecuadamente a los datos observados de estas series. En esta situación, una alternativa útil a un modelo AR es el modelo de medias móviles $MA(q)$ (moving average). Un proceso Z_t es un proceso de medias móviles si se puede expresar como un promedio ponderado (media móvil) de los últimos q valores del proceso del ruido blanco y, en lugar de los valores del pasado como sucede en un proceso AR . Así

$$Z_t = a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + \dots + \theta_q a_{t-q} \quad (13)$$

Ajustar modelos MA es complicado. Aún cuando la ecuación (13) parece similar al modelo de regresión con $a_t, a_{t-1}, a_{t-2}, \dots, a_{t-q}$ como las variables explicativas, no lo son. El problema es que $a_t, a_{t-1}, a_{t-2}, \dots, a_{t-q}$ no son observables y por lo tanto deben ser estimados al mismo tiempo que los coeficientes θ_i del modelo MA son estimados. Igual que en un modelo AR la estimación de a_t es llamado el residual.

Usando el operador θ autorregresivo de orden q podemos escribir el modelo $MA(q)$ de la forma

$$Z_t = \theta(B)a_t \quad (14)$$

El Modelo $ARIMA(p,q,d)$. El modelo más general para ST es $ARIMA(p,q,d)$ (AutoRegressive Integrated Moving Average processes). En la práctica muchas ST son no estacionarias y así no se pueden aplicar los procesos estacionarios AR , $ARMA$ directamente. Una posible forma de manipulación para ST no estacionarias es aplicar diferenciación y así hacerlas estacionarias. La primera diferenciación es $\nabla z_t = z_t - z_{t-1} = (1-B)z_t$. La d -ésima diferenciación puede escribirse como $(1-B)^d z_t$. Si los datos originales de la serie son diferenciables d veces antes de ajustar un

proceso ARMA(p, q), entonces el modelo para la serie original sin diferenciación es llamado un proceso ARIMA(p, d, q), donde d denota el número de diferenciaciones hechas.

El modelo ARIMA(p, d, q) puede escribirse de la forma concisa

$$\phi(B)(1-B)^d Z_t = \theta(B) a_t \quad (15)$$

Notar que la ecuación (14) es una generalización de la expresión (12). También es importante ver que la función polinomial $\phi(x)(1-x)^d$ tiene d raíces sobre el círculo unitario sobre los números complejos (cuando $x = 1, (1-x)=0$) lo que indica que el proceso no es estacionario, lo que obliga a la diferenciación.

En los modelos ARIMA, aparte de los problemas similares a los de los procesos AR, se tiene una adicional dificultad al escoger el orden requerido de diferenciación, es decir ¿cual d es más conveniente?. Existen algunos procedimientos para resolver esta dificultad, ver por ejemplo [7].

3.2 Principales resultados de la metodología de Box Jenkins.

La metodología BJ usada en el análisis, modelación y pronóstico de ST es ampliamente considerada como la técnica de pronóstico más eficiente, y se utiliza ampliamente, sobre todo para las series temporales univariantes. La estrategia de esta metodología consiste de tres fases: identificación, estimación y control de diagnóstico.

A diferencia de otras técnicas, BJ es un procedimiento que utiliza el comportamiento pasado de una variable para seleccionar el mejor modelo de previsión de una clase general de los modelos. Se supone que cualquier modelo de ST puede ser representado por una de tres categorías de modelos. Estas categorías incluyen:

- Los modelos autorregresivos: los pronósticos de una variable basados en la función lineal de sus valores pasados.
- Modelos de promedios móviles: los pronósticos basados en la combinación lineal de los errores del pasado.
- Modelos de promedios móviles: la combinación de las dos categorías anteriores.

Una de las preguntas claves que deben ser resueltas en esta metodología es cuantos valores del pasado (de la variable de y / o de sus errores) debe incluirse en el modelo.

Hay esencialmente tres etapas en un BJ procedimiento:

1. Identificación de un modelo tentativo. De las tres categorías mencionadas anteriormente, cuál de ellas es la apropiada para hacer en primer lugar los datos estacionarios (generalmente se hace la diferenciación de los datos) y entonces se autocorrelaciones parciales de los datos estacionarios. Este análisis indica el posible número de parámetros ϕ_j , θ_j que el modelo debería tener. Por lo tanto, determinar el tipo adecuado del modelo para

una situación específica es principalmente una cuestión de coincidencia de las correlaciones observadas a las correlaciones teóricas.

Además de la presencia de estacionariedad en una serie temporal, existe otra propiedad que se requiere en los modelos de ARIMA denominada invertibilidad, la cual permite representar la serie como un modelo autorregresivo de extensión infinita que satisface la condición.

$$\lim_{j \rightarrow \infty} \phi_j = 0$$

Lo anterior permite que, tomando un número finito de términos, se obtenga una expresión satisfactoria de la forma (1) para la serie. Lo anterior significa que solo los modelos ARIMA que tengan la propiedad de invertibilidad pueden ser aproximados por un modelo AR de la forma (1).

2. Estimación de los parámetros involucrados en el modelo, mediante técnicas de estimación no lineal.
3. Verificación de que el modelo proporciona un ajuste adecuado y de que los supuestos básicos implícitos en el modelo son satisfechos mediante el análisis del comportamiento de los residuales. Cabe señalar que en esta propuesta no se lleva a cabo dicho análisis debido a que los residuales no corresponden, en lo general, a ruido blanco.
4. Aplicación del modelo.

Las propuestas de BJ para los pronósticos proporcionar algunos de los más precisos pronósticos a corto plazo. Sin embargo, estas propuestas requieren de una gran cantidad de datos.

A continuación presentamos las características de los algoritmos heurísticos propuestos.

3.3 Algoritmos propuestos.

Los algoritmos heurísticos que construimos están basados en las siguientes premisas:

- I. Sin importar el tipo de serie original (estacionaria o no estacionaria), el modelo que se busca siempre será de la forma AR presentada en (1).
- II. Para determinar cuántos retrasos p se requieren, primero se elige con que serie de diferencias se van a estimar éstos. A la serie elegidas la denominaremos serie de trabajo. Después se define cuantos retrasos se requieren conforme al comportamiento de la función de autocorrelación muestral de la serie de trabajo. Esto implica una diferencia con la metodología BJ, que aplica el número de retrasos en términos de la información que brindan, tanto la función de autocorrelación muestral, como la función de autocorrelación parcial y la hipótesis del componente aleatorio como ruido blanco.

Esta elección trae como consecuencia que a_i en los modelos aquí desarrollados no sea ruido blanco.

- III. En nuestros modelos pediremos para la serie de trabajo la condición de que ϕ_i este en $(-1.0, 1.0)$.

Cabe agregar que los algoritmos heurísticos que aquí se presentan solo exploran un conjunto pequeño de todos los modelos AR para una ST. Sin embargo, permiten el tratamiento de series que tengan tendencia y varianza que dependan del tiempo, pues no requieren que se cumplan las condiciones que tradicionalmente se les pide a las ST, como es el hecho de que sean estacionarias o de varianza estacionaria o que resulten de aplicar alguna transformación de tipo logarítmico o de promedios móviles.

El primer algoritmo que proponemos, construye una aproximación lineal para la serie de diferencias (de primero, segundo o tercer orden) que pudiera ser estacionaria. Luego, a partir de esta aproximación lineal y utilizando el resultado 1 de este capítulo, se construye otro modelo lineal de la serie original. A continuación detallamos las etapas de este primer algoritmo.

3.3.1 Primer Algoritmo.

En la primera etapa lo primero que se hace es decidir con que serie se va a trabajar entre la serie original, la de primeras diferencias, la de segundas diferencias, y en este trabajo se incluye la posibilidad de trabajar con la de terceras diferencias. Para decidir esto se elige la que tenga mínima varianza, lo que consideramos un indicio de tener una serie estacionaria [17]. La serie elegida será la serie de trabajo.

Una vez definida la serie de trabajo, se estima cuantos términos son necesarios para la aproximación lineal de esta serie basándonos en la función de autocorrelación. En este trabajo calculamos 30 valores para la función de autocorrelación, y para elegir cuantos términos se requieren tenemos dos casos. Si la función es decreciente, se toma un valor de 4 y en caso contrario se considera un valor igual al valor en el que aparece el primer máximo de esta función (ver Figura 5). Con este procedimiento, si la serie presenta estacionalidad y el periodo es menor que 30, los modelos que aquí se construyen pueden representar apropiadamente dicha estacionalidad.

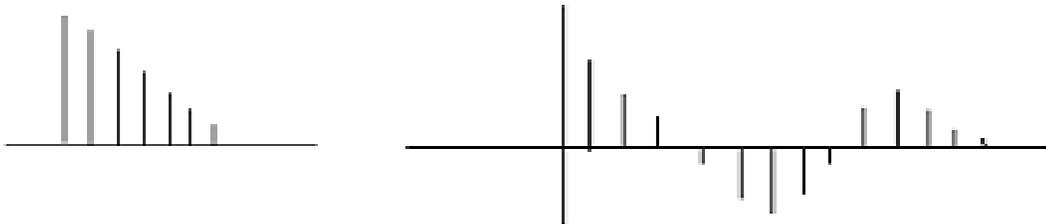


Fig. 5.- Posibles gráficas de funciones de autocorrelación.

Con la información anterior se construyen los límites para los intervalos de los coeficientes de la serie de trabajo para lo cual se toman todos los ϕ_j en $(-1.0, 1.0)$ excepto el termino

independiente δ cuyos límites se estiman entre cero y el valor promedio de la serie. Con todo lo anterior se tienen completa la propuesta sobre el número p de términos que se requieren y cuáles son los límites de sus coeficientes. A partir de esta información se resuelve el **problema 1** aplicando los AGA en las dos primeras etapas que se detallan en la sección 2.4.

Basándonos en el siguiente:

Resultado 1. Si $\{y_t\}$ es una serie de diferencias para $\{x_t\}$ con modelo

$$y_t = h_0 + h_1 y_{t-1} + h_2 y_{t-2} + \dots + h_k y_{t-k}$$

Entonces, para la serie de diferencias con términos $y_t = x_t - x_{t-1}$ se tiene que

$$x_t = h_0 + (1 + h_1)x_{t-1} + (h_2 - h_1)x_{t-2} + \dots + (h_k - h_{k-1})y_{t-k} - h_k y_{t-k-1} \quad (16)$$

Es un modelo para la serie $\{x_t\}$.

De este resultado se tienen dos consecuencias importantes:

- El modelo para la serie $\{x_t\}$ tiene un término más que la de la serie $\{y_t\}$.
- Si algún valor de y_t tiene coeficiente entre -1.0 y 1.0, el coeficiente de x_t puede no estar en este rango.

Aplicando el resultado 1 tantas veces como sea necesario, podemos obtener un modelo para la serie original, y a este modelo le aplicamos la etapa tres de la sección 2.4 para obtener un modelo lineal para la ST. Cabe observar que si se tiene un modelo AR para alguna serie de diferencias, el modelo que se construye para la serie original a partir de (16), tiene más términos que el de la serie de diferencias, así que si se requieren K términos para la serie original, se tienen que buscar modelos para las series de diferencias con menos términos que K . En el anexo 2 se presenta una descripción de este algoritmo.

3.3.2 Segundo Algoritmo.

El segundo algoritmo sólo utiliza de la metodología de BJ, la estimación de cuantos términos son necesarios en la aproximación lineal de la serie de diferencias que pudiera ser estacionaria, y a partir de esto, se estima cuantos términos se utilizarán para la serie original.

De ahí en adelante se aplican las etapas que se presentan en la sección 2.4, tomando los límites de todos los coeficientes en (-1.0, 1.0), pero trabajando siempre con la serie original. No existe algún resultado que justifique el uso de estos límites, y sólo se han encontrado unos autores que los usan [8] [9] [10]. Por otra parte, en NN3 es un hecho que en un alto porcentaje de casos, se obtienen mejores resultados con este algoritmo que con los del primero. Como un ejemplo de esto se tiene que el segundo algoritmo tuvo mejor

desempeño que el primero en cuarenta y seis de los ciento once ejemplos de NN3-Complete. En el anexo 2 se presenta una descripción del segundo algoritmo.

3.4 Comparación con Box Jenkins.

En esta sección se comparan los modelos desarrollados con esta metodología únicamente en la capacidad de aproximación sobre el conjunto de entrenamiento. Aquí la idea es demostrar que no obstante es más pequeño el conjunto de modelos AR que se construyen con esta metodología, la aproximación en el conjunto de prueba es muy buena. Un estudio más detallado sobre la capacidad de pronóstico sobre el conjunto de prueba se presentará en el capítulo siguiente.

Para probar nuestros algoritmos, utilizamos las ST que aparecen en [4] con el nombre de Series: A, B, C, D, E, F y G. Se eligieron éstas, por ser ampliamente conocidas y presentar una amplia variedad de patrones de ST. Las series A, C, D y F representan valores de concentración, temperatura, viscosidad y rendimiento respectivamente de diferentes procesos químicos. La serie B representa los valores de los precios diarios de las acciones de IBM en un cierto periodo de tiempo. La serie E mide el promedio anual de manchas solares desde 1770 hasta 1869. La serie G que aparece en la Figura 6, representa el número de pasajeros en vuelos internacionales entre enero de 1949 y diciembre de 1960

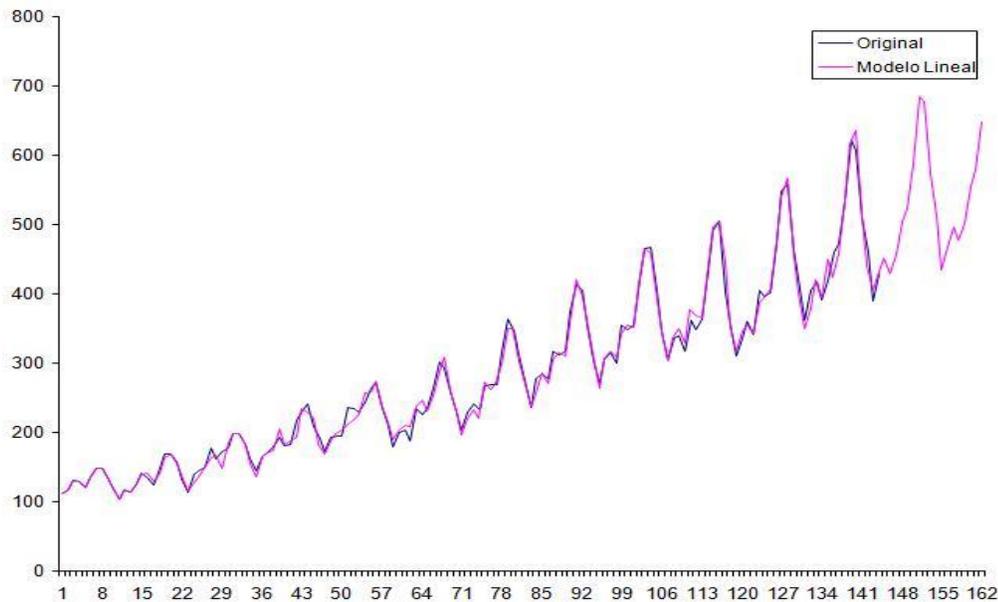


Fig. 6.- Número de pasajeros en vuelos internacionales.

Dado que los modelos que se construyen en este trabajo tienen preestablecido el número K de términos, y esto no es lo que se hace en la literatura, para poder comparar adecuadamente estos modelos con los obtenidos por BJ, es necesario primero obtener los modelos de BJ para estimar cuantos términos tienen, y después construir los modelos aquí propuestos, con el mismo número de términos que los de BJ.

Para esto se tomaron los resultados de los modelos de tabla 7.13 de la página. 239 de [4], y aplicando la propiedad de invertibilidad, se procedió a obtener expresiones de la forma (1), con el número de términos que aparecen en la tabla I. Posteriormente se aplicaron los algoritmos aquí desarrollados para construir modelos AR con el mismo número de términos que tienen los modelos de BJ.

Tabla I
Primera comparación con Box Jenkins.

Serie	Modelos Lineales	RSS
A-BJ	$F_t = 0.3z_{t-1} + 0.21z_{t-2} + 0.147z_{t-3} + 0.1029z_{t-4} + 0.072z_{t-5} + 0.0504z_{t-6} + 0.0348z_{t-7}$	20.003
A-OA	$F_t = 0.6446 + 0.3867z_{t-1} + 0.1588z_{t-2} + 0.2978z_{t-7} + 0.1194z_{t-16}$	4.131
B-BJ	$F_t = 1.09z_{t-1} - 0.0981z_{t-2}$	155.365
B-OA	$F_t = 0.1045 + 0.9992z_{t-1}$	139.01
C-BJ	$F_t = 1.82z_{t-1} - 0.820z_{t-2}$	1.989
C-OA	$F_t = 1.8136z_{t-1} - 0.8140z_{t-2}$	1.984
D-BJ	$F_t = 1.17 + 0.87z_{t-1}$	5.220
D-OA	$F_t = 0.9379z_{t-1} + 0.87z_{t-5}$	5.209
E-BJ	$F_t = 14.35 + 1.42z_{t-1} - 0.73z_{t-2}$	149.137
E-OA	$F_t = 1.7752z_{t-1} - 1.2745z_{t-2} + 0.4395z_{t-3}$	149.137
F-BJ	$F_t = 58.87 - 0.340z_{t-1} + 0.190z_{t-2}$	85.015
F-OA	$F_t = 52.1139 - 0.257z_{t-1} + 0.2413z_{t-2}$	85.085
G-OA	$F_t = 1.9991 + 0.2199z_{t-1} + 0.1472z_{t-11} + 0.7183z_{t-12}$	215.354

En esta tabla I no existe un modelo de BJ para la serie G, pues con la metodología de BJ no es posible encontrar una expresión lineal de la forma (1). Además, la serie G presenta varianza creciente, por lo que es necesario aplicar una transformación logarítmica para estabilizar su varianza. Cabe señalar que con esta propuesta se pueden resolver el encontrar modelos para todos los problemas presentados por BJ. En la Figura 6, se muestran nuestros resultados de la gráfica de la serie G, los últimos 18 valores corresponden al pronóstico obtenido por nuestro modelo.

En la tabla I también aparecen las expresiones de los modelos calculados por *Our Algorithms* (OA) al obtener un modelo lineal con el mismo número de términos que los del modelo de BJ correspondiente. Es de señalar que los modelos aquí construidos difieren substancialmente en sus componentes, de los construidos por BJ. Además, al comparar los valores del RSS concluimos que estos dan valores similares con la diferencia de que los algoritmos aquí desarrollados son totalmente automatizados, y además no requieren de aplicar una etapa previa de preprocesamiento de los datos. A continuación presentamos los resultados que se presentaron en la competencia NN3 [49].

3.5 Resultados de NN3

La competencia internacional *NN3 Artificial Neural Network & Computational Intelligence Forecasting Competition 2007* tiene como finalidad evaluar las últimas metodologías para el pronóstico de ST. Esta competencia está abierta a utilizar métodos basados en Redes Neuronales, Lógica Difusa, Algoritmos Genéticos y otros del área de inteligencia artificial. Los problemas a tratar se presentan en dos conjuntos llamados NN3-Complete (con 111 ejemplos de ST) y NN3-Reduced (con 11 ejemplos), y el propósito de la competencia es obtener los mejores modelos para cada ejemplo de los dos conjuntos utilizando una misma metodología. La notación de esta sección es semejante la que se usa en la pagina original [48].

Para evaluar el desempeño de un modelo en algún ejemplo s , se estima el pronóstico F y se mide el desempeño con el promedio del indicador *Symmetric Mean Absolute Percent Error* (SMAPE) sobre todos los valores de la serie. El SMAPE mide el error absoluto simétrico en porcentaje, entre los valores reales de la serie original Z y el pronóstico F sobre todas las observaciones t del conjunto de prueba de tamaño n para cada serie s con $SMAPE_s$ igual a:

$$\frac{1}{n} \sum_{t=1}^n \frac{|Z_t - F_t|}{(Z_t + F_t)/2} * 100 \quad (17)$$

y al final se promedia sobre todos los ejemplos en el mismo conjunto de datos. Otras medidas de la precisión del pronóstico de un modelo se pueden encontrar en [22].

Con este indicador se puede evaluar el desempeño de aplicar distintas metodologías sobre el mismo conjunto de datos y la metodología que produzca menor valor se considera la mejor. En el conjunto NN3-Complete el mejor resultado fue de 14.84% y aplicando los algoritmos aquí desarrollados se obtuvo el 16.31%. En el NN3-Reduced los resultados respectivos fueron 13.07% y 15.00%.

Sin embargo es posible construir otros modelos lineales con la metodología presentada en este trabajo que mejorar dichos resultados pues:

- Aunque el propósito de la competencia era determinar el mejor modelo para cada ejemplo, en este trabajo a todos los ejemplos se les encontró un modelo AR con 4 términos. Es de esperarse que si se parte la serie en un conjunto de entrenamiento y otro de prueba se encuentren modelos con mayor capacidad de pronóstico que mejoren los resultados obtenidos.
- Se compitió utilizando modelos lineales contra modelos obtenidos con técnicas no lineales.
- No se utilizaron modelos ARMA que incluyen el comportamiento de los residuales ni el *adelanto del pronóstico* (estos aspectos se presentaran en el capítulo 4) que mejoran substancialmente los resultados.

A continuación se describe el trabajo realizado para construir los modelos de la competencia NN3. Primero se trabajo en los problemas de NN3-Reduced en donde, con los dos algoritmos desarrollados, se realizaron 50 ejecuciones de cada algoritmo en cada ejemplo, buscando

modelos lineales con 4 términos. En la tabla II se presentan los resultados de las expresiones lineales así como el cálculo de RSS.

Después de revisar el comportamiento de las 50 soluciones de estos ejemplos se concluyó que bastaban cinco ejecuciones para obtener resultados satisfactorios. Por esta razón para los ejemplos de la serie NN3-Complete sólo se realizaron 5 ejecuciones con cada algoritmos y se eligió la mejor de estas. Los resultados de los ejemplos de NN3-Complete no se presentan en este trabajo.

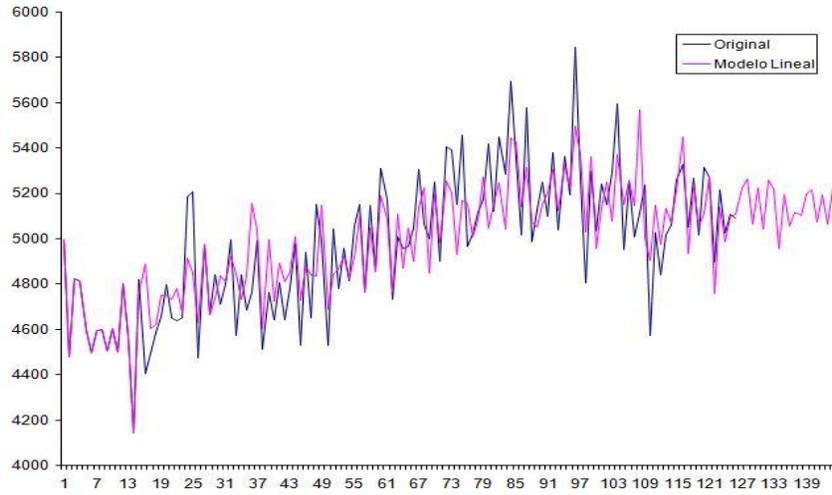
En la sección 3.5.1 se muestran las gráficas de las series obtenidas con el mejor resultado de los algoritmos heurísticos aquí presentados. Los valores correspondientes a los últimos 18 puntos que aparecen en las gráficas son el resultado del pronóstico obtenido al evaluar las expresiones de los modelos lineales que aparecen en la tabla II.

Tabla II
Modelos lineales para NN3-REDUCED.

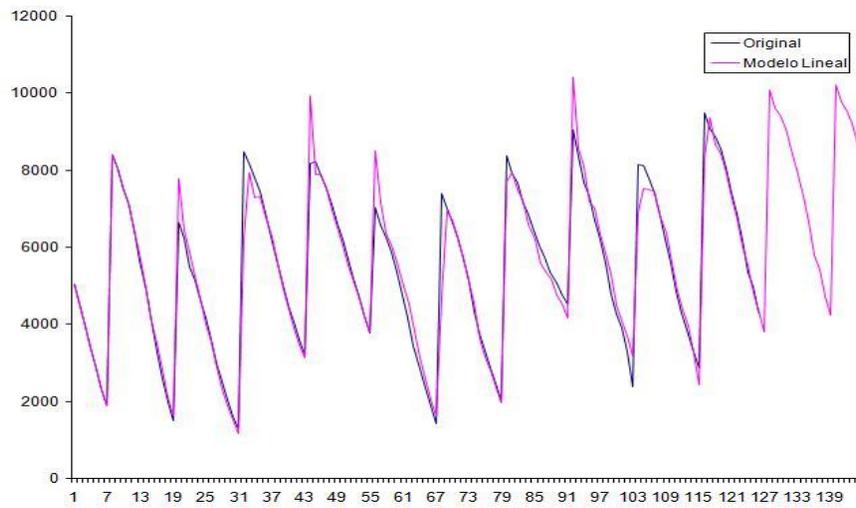
Problema	Modelo Lineal	RSS
101	$F_t = 1269.3358 + 0.3467z_{t-1} + 0.6978z_{t-12} - 0.2921z_{t-13}$	1713.55
102	$F_t = 1.9987 + 0.9218z_{t-1} + 0.9574z_{t-12} - 0.8792z_{t-13}$	5440.262
103	$F_t = 1.9989 + 0.5152z_{t-1} + 0.8124z_{t-12} - 0.3734z_{t-13}$	80019.738
104	$F_t = 9.113 + .7252z_{t-1} + 0.8316z_{t-12} - 0.5592z_{t-13}$	7321.538
105	$F_t = 1.998 + 0.9099z_{t-1} + 0.3104z_{t-11} - 0.2225z_{t-13}$	1513.984
106	$F_t = 2821.9541 + 0.2673z_{t-2} - 0.1699z_{t-7} + 0.3422z_{t-12}$	4464.87
107	$F_t = 0.9978 + 0.7937z_{t-1} + 0.3152z_{t-12} - 0.1125z_{t-13}$	1387.011
108	$F_t = 2000.5819 + 0.2885z_{t-2} - 0.1456z_{t-4} + 0.2379z_{t-5}$	10417.433
109	$F_t = 1.9988 + 0.9951z_{t-1}$	2297.306
110	$F_t = 1863.0699 + 0.252z_{t-1} - 0.10582z_{t-5} + 0.2379z_{t-11}$	18593.279
111	$F_t = 474.1106 + 0.242z_{t-11} - 0.3319z_{t-12} + 0.2688z_{t-13}$	7248.281

3.5.1 Gráficas de NN3-Reduced.

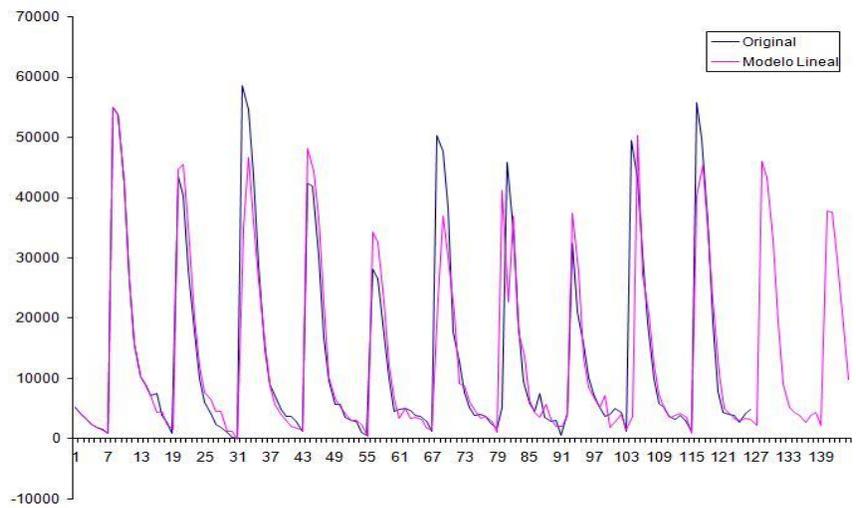
A continuación se presentan las gráficas de los resultados obtenidos para cada ejemplo de NN3-Reduced.



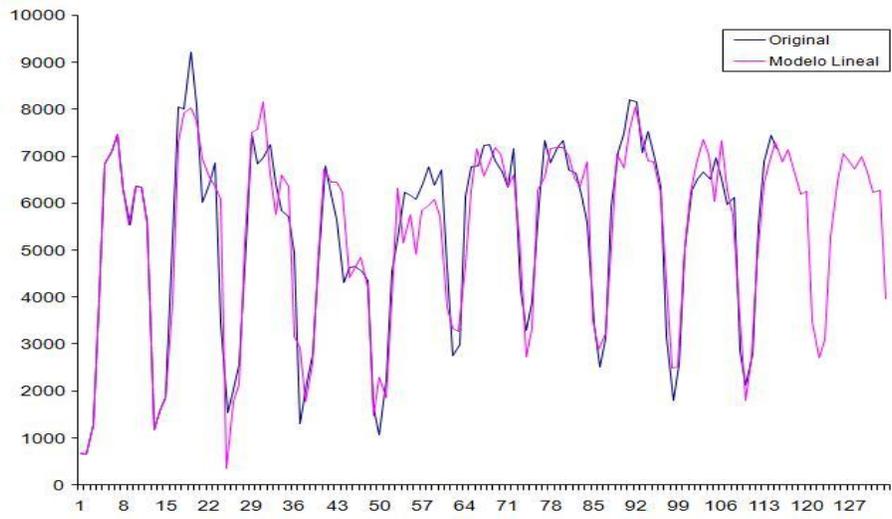
Gráfica 1. Ejemplo 101.



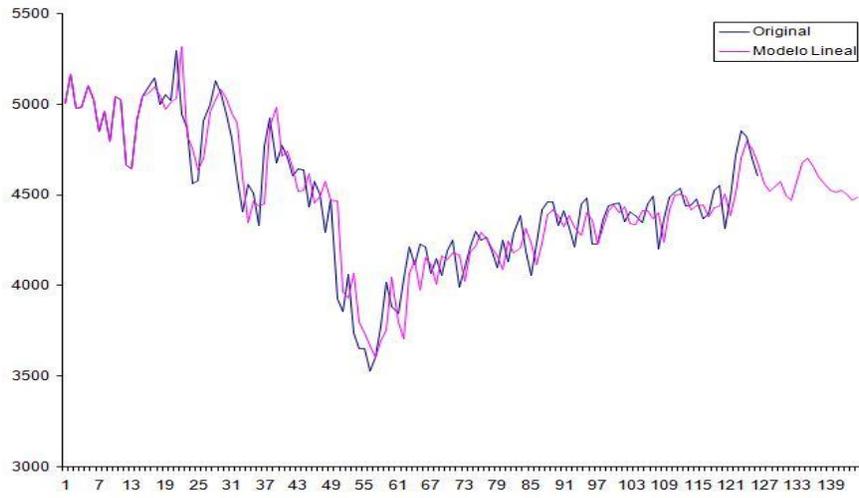
Gráfica 2. Ejemplo 102.



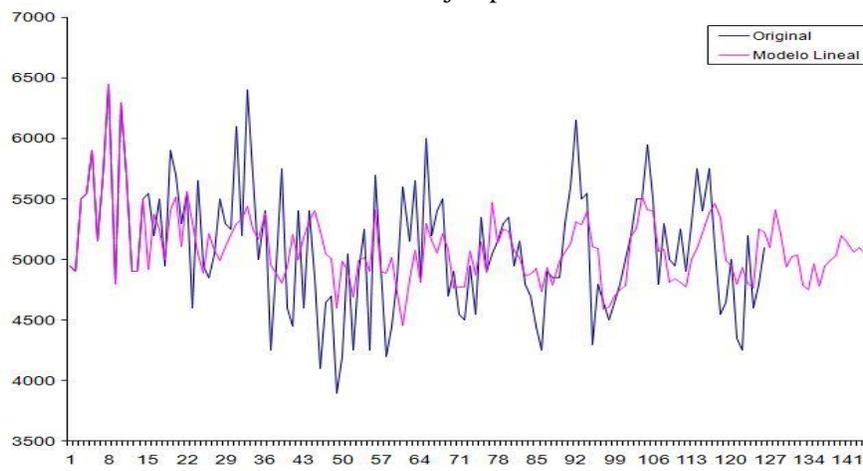
Gráfica 3. Ejemplo 103.



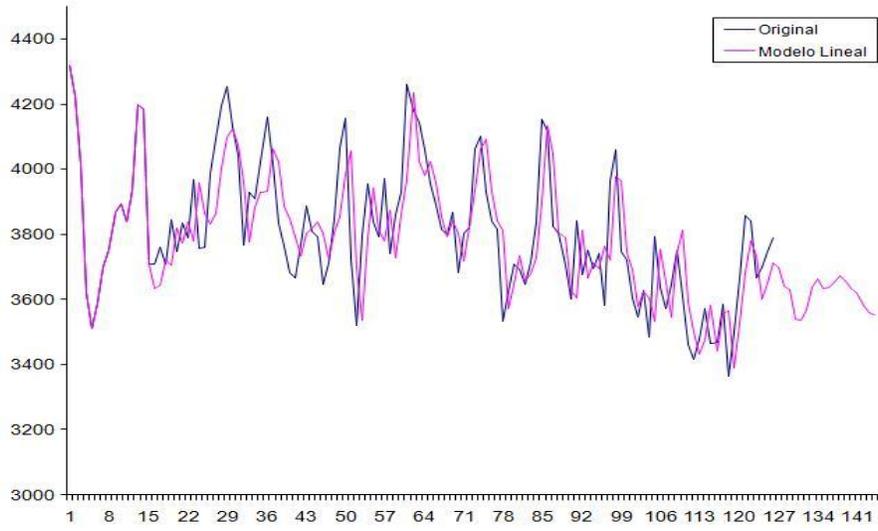
Gráfica 4. Ejemplo 104.



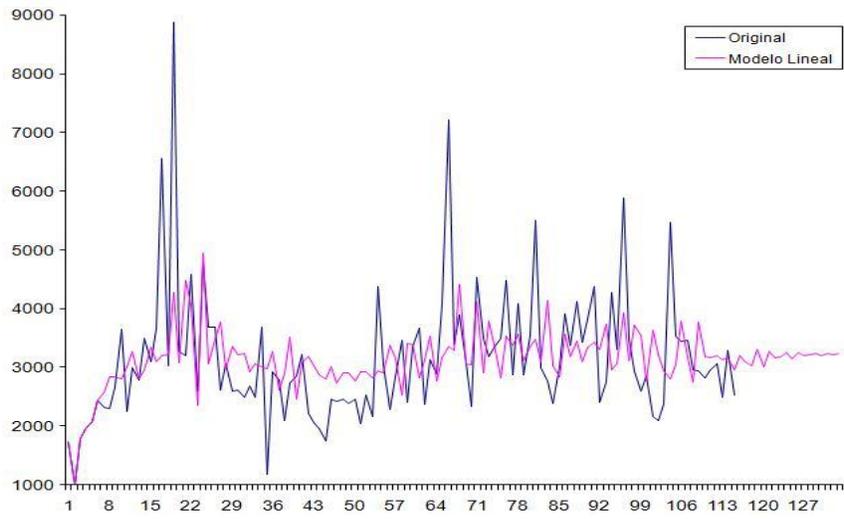
Gráfica 5. Ejemplo 105.



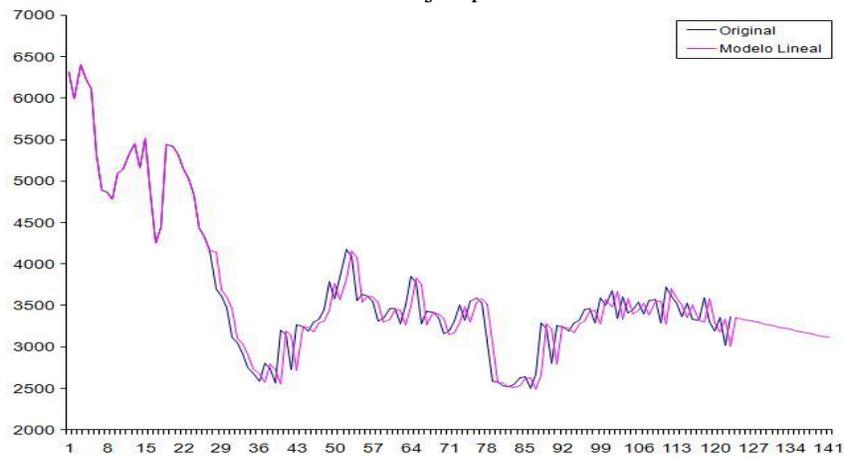
Gráfica 6. Ejemplo 106.



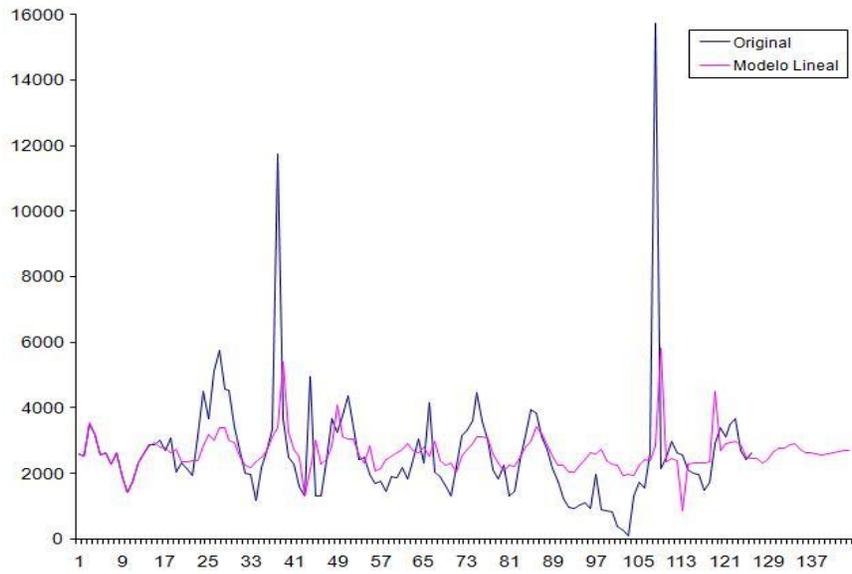
Gráfica 7. Ejemplo 107.



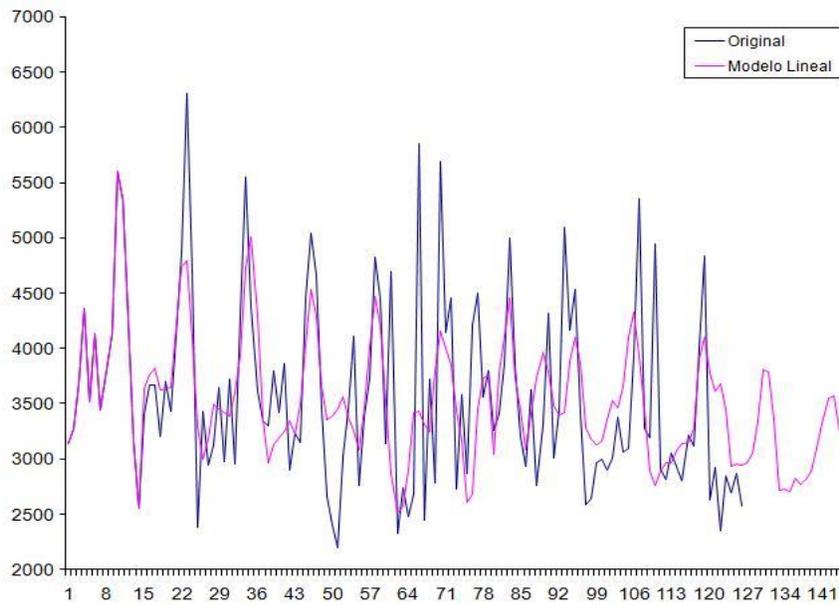
Gráfica 8. Ejemplo 108.



Gráfica 9. Ejemplo 109.

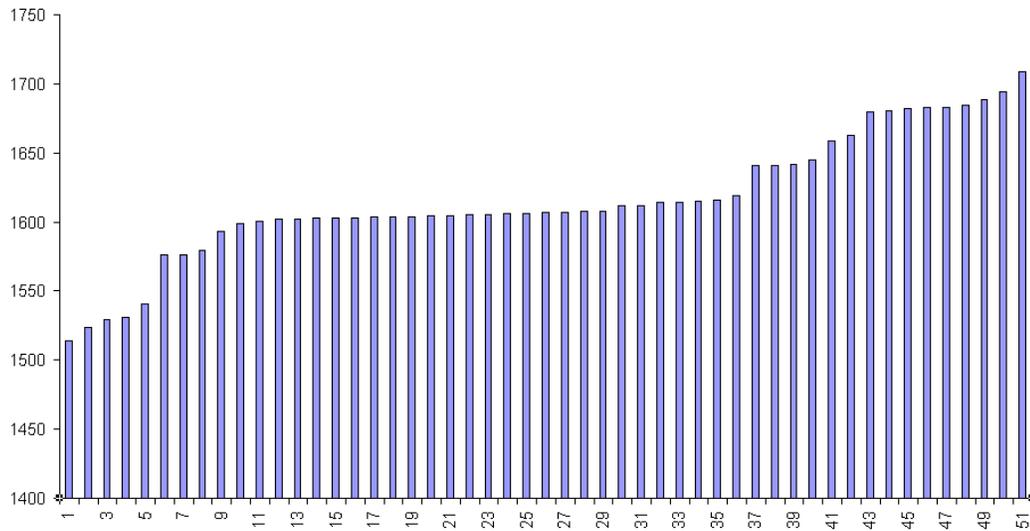


Gráfica 10. Ejemplo 110.



Gráfica 11. Ejemplo 111.

Como una muestra de que nuestro problema tiene varios mínimos locales, en la gráfica 12 se muestran los resultados de 50 iteraciones del primer algoritmo para el problema 105 de NN3, de los escalones que se observan en la gráfica concluimos que existen varios mínimos locales.



Gráfica 12. Soluciones del ejemplo 105.

3.6 Elección de un modelo.

Cuando se construye un modelo, una vez que se tiene éste, hay que responder a la pregunta: ¿Qué tan seguro estoy de que es un “buen” modelo? o en el caso de que se tengan varios candidatos ¿cómo decidir cual modelo es el más adecuado?. La buena elección de un modelo nos permitirá suponer que los pronósticos del modelo serán confiables al menos en el corto plazo.

Para abordar este tema en la metodología de BJ se incluye una etapa de verificación del modelo en donde se analiza este problema centrándose en el comportamiento de los residuales y estableciendo criterios para verificar cuando son un ruido blanco [17]. Cuando se logra construir un modelo y se prueba que los residuales son un ruido blanco, se tiene la certeza de que si se construye una banda centrada sobre los valores predichos por el modelo de ancho 2σ (para arriba y para abajo) con σ la desviación estándar del ruido blanco, se puede asegurar que los valores del proceso real estarán dentro de dicha banda con una probabilidad del 95%.

Por lo anterior, cuando se tienen dos candidatos, que ya aprobaron la prueba de residuales, sobre los que se tiene que decidir cuál elegir, se prefiere al que tenga menor desviación estándar pues esto significa que la banda sobre el modelo es más cerrada. En [37] se presentan otros criterios para selección de modelos basados en la desviación estándar.

3.6.1 Prueba de Bondad de Ajuste.

Dado que los residuales de los modelos aquí construidos no son ruido blanco, y la prueba tradicional de medición de bondad de ajuste del modelo está basada en este supuesto, es necesario desarrollar una prueba alternativa para medir el desempeño de los modelos aquí desarrollados. Para esto se parte de la suposición de que se tiene un conjunto de datos con comportamiento cercano a los datos reales, y si adicionalmente se incluye que, respecto a los datos reales, los valores predichos por el modelo cumplen con que tienen la misma

media y la misma varianza, las tres condiciones anteriores representarían que los valores del modelo tienen un comportamiento muy parecido a los datos reales.

Por todo lo anterior, se proponen como medidas de bondad del modelo encontrado para una ST con nuestra metodología, las siguientes:

La comparación estadística tanto de la media y de la desviación estándar de la serie observada y de la serie pronosticada mediante pruebas de hipótesis de la forma:

$$\begin{array}{ll} \text{Ho: } \mu = \mu_0 & \text{Ho: } \sigma^2 = \sigma_0^2 \\ \text{H1: } \mu \neq \mu_0 & \text{H1: } \sigma^2 \neq \sigma_0^2 \end{array}$$

Donde tanto μ_0 y σ_0^2 son tomados como los parámetros de la ST observada.

Para nuestro caso una prueba estadística es una afirmación sobre los parámetros del modelo y reflejan una conjetura sobre el comportamiento del mismo. Las comparaciones estadísticas anteriormente señaladas son clásicas de la literatura estadística y un tratamiento completo de ellas se puede encontrar en [35]. A continuación se ejemplifica el procedimiento.

Por ejemplo, en la comparación

$$\begin{array}{l} \text{Ho: } \mu = \mu_0 \\ \text{H1: } \mu \neq \mu_0 \end{array}$$

Al primer enunciado se le llama hipótesis nula y al segundo hipótesis alternativa y para probar una hipótesis se calcula un estadístico de prueba apropiado para rechazar o no estar en posición de rechazar la hipótesis nula.

Para probar una hipótesis se elige un estadístico de prueba de acuerdo al problema de que se trate y se especifica el conjunto de valores del estadístico que definen si se rechaza H_0 . Al conjunto de estos valores se le llama región crítica o de rechazo de la prueba.

Cuando se prueba una hipótesis se pueden cometer dos tipos de errores: El primero o error de tipo I ocurre cuando la hipótesis nula se rechaza siendo esta verdadera y el error de tipo II ocurre cuando la hipótesis nula no se rechaza y es falsa. Las probabilidades de estos dos errores se denotan por:

$$\begin{array}{l} \alpha = P(\text{error tipo I}) \\ \beta = P(\text{error tipo II}) \end{array}$$

En la prueba de hipótesis se especifica un valor para α y después, de acuerdo al estadístico usado, se diseña un procedimiento de prueba de tal forma que el valor de β tenga un valor convenientemente pequeño.

Para la prueba de hipótesis sobre la media de la serie se aplica como estadístico de prueba z y para la varianza, el estadístico χ^2 donde

$$z = \frac{\sqrt{n}(\bar{x} - \mu)}{\sigma} ; \quad \chi^2 = \frac{(n-1)s^2}{\sigma^2}$$

Siendo \bar{x} y s^2 estimadores de la media y la varianza de serie pronosticada.

Para evaluar el desempeño de nuestros modelos con respecto al criterio del valor medio y la varianza, se elige una valor de $\alpha = 0.025$ y en la Tabla III se indican los resultados obtenidos para las series modeladas. Los valores de las columnas dos a la seis son valores estadísticos obtenidos de la serie original y su modelo.

Tabla III
Comparación de valores medios y varianzas.

Prob.	μ	σ	\bar{x}	s	n	z	$z_{0.025}$	χ^2	$\chi^2_{0.025}, \chi^2_{0.975}$
101	4974	302	4974	250	126	0	∓ 1.96	85.66	95.95,157.84
102	5406	2058	5389	2042	126	0.0927	∓ 1.96	123.06	95.95,157.84
103	13884	16096	13459	14819	126	0.2964	∓ 1.96	105.95	95.95,157.84
104	5339	2054	5311	2016	115	0.1462	∓ 1.96	109.82	86.34,145.44
105	4477	376	4474	374	126	0.0896	∓ 1.96	123.67	95.95,157.84
106	5098	512	5103	307	126	-0.1096	∓ 1.96	44.94	95.95,157.84
107	3811	203	3802	183	126	0.4977	∓ 1.96	101.58	95.95,157.84
108	3143	1110	3143	530	115	0	∓ 1.96	25.99	86.34,145.44
109	3708	903	3704	908	123	0.0491	∓ 1.96	123.35	93.32,154.46
110	2638	1771	2638	609	126	0	∓ 1.96	14.78	95.95,157.84
111	3585	851	3584	553	126	0.0132	∓ 1.96	52.78	95.95,157.84

Al comparar los valores de las columnas siete y ocho se observa que los valores de la columna siete siempre andan dentro del intervalo definido en la columna ocho. De lo anterior se concluye que todos los modelos asociados a las series producen un valor medio estadísticamente similar.

Por otra parte al comparar los valores de las columnas nueve y diez se observa que los modelos de las series 101, 106, 108, 110 y 111 en la columna nueve no están en los intervalos definidos en la columna diez, y en consecuencia no pasaron la prueba para el valor de la varianza. Cabe señalar que al revisar las gráficas de estas series se encuentra que son las que más disparidad presentan entre los valores del modelo y la serie real.

Si se tiene un modelo para una ST cuyos residuales son un ruido blanco, esto no significa que deban de aprobar la prueba de bondad de ajuste aquí propuesta pues pudiera darse el caso de que la desviación estándar del ruido blanco fuera muy grande y no pasara las pruebas aquí planteadas. De manera semejante, si se tiene un modelo que aprueba los criterios aquí establecidos, lo más seguro es que los residuales no sean un ruido blanco y en consecuencia no sea un buen modelo según BJ.

Por último, cabe señalar que la propuesta de prueba de bondad de ajuste aquí presentada se aplica también a los modelos lineales del capítulo siguiente y a los obtenidos para FTD.

Capítulo 4

Modelos ARMA para Series de Tiempo

En este capítulo se aplican los resultados ya obtenidos para obtener componentes AR de la serie del error obtenida al restar de la serie original los valores que le asigna el modelo AR. Al considerar un nuevo modelo sumando estas dos componentes, se obtiene el equivalente en nuestra metodología de los modelos ARMA tradicionales.

En la primera sección de este capítulo, se presenta a manera de ejemplo, la gráfica 13 del error obtenido con nuestro método para una cierta serie, que por el comportamiento que presenta, se puede concluir que no es un ruido blanco. Cabe señalar que este es un resultado general de nuestra metodología. Por lo anterior podemos construir modelos AR para estas series de error, los cuales tendrán la capacidad de modelar adecuadamente el error, lo que permitirá, al considerar estos dos modelos, obtener una mayor capacidad de pronóstico.

Posteriormente presentaremos el fenómeno del *retardo en el pronóstico* (RP), y definiremos la operación de *adelantar el pronóstico* (AP), lo que permitirá construir modelos más precisos para ST. Dado que uno de los propósitos de la tesis es desarrollar una metodología que sea totalmente automatizable, lo cual se lograría totalmente si se supiera bajo qué condiciones se presenta el fenómeno del RP, se procederá a analizar este fenómeno aplicando un algoritmo de clasificación basado en reglas y las condiciones bajo las cuales se presenta dicho fenómeno.

Por último se presenta una comparación de nuestros modelos con otros obtenidos con distintas metodologías utilizadas para obtener modelos Lineales y no Lineales para ST.

4.1 Construcción de modelos ARMA.

Los modelos más generales que trabajaremos en esta tesis son $ARMA(p,q)$, que como se vio en el capítulo anterior contienen la presencia de componentes autorregresivas en las variable observable Z_t y en el error a_t . Así tenemos

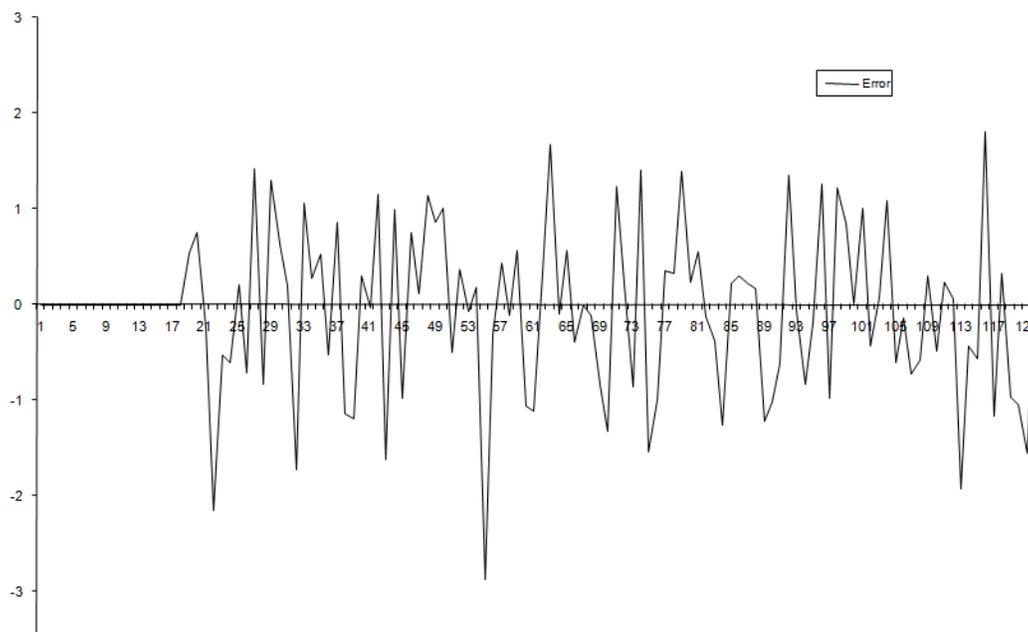
$$a_t = Z_t - (\delta + \sum_{i=1}^p \theta_i Z_{t-i})$$

y

$$F_t = \delta + \sum_{i=1}^p \phi_i Z_{t-i} + \sum_{j=1}^q \theta_j a_{t-j}$$

donde F_t es el valor pronosticado del modelo, ϕ_i y θ_j son los coeficientes de AR y MA respectivamente y δ es una constante.

Una vez que se tiene un modelo AR para una serie, podemos construir un modelo ARMA, a partir de obtener otro modelo AR para la serie obtenida al considerar los errores a_t entre la serie original y su modelo AR. Al agregar al modelo AR otra componente adicional que considere los términos autorregresivos correspondientes al error, obtenemos el modelo ARMA completo. En la gráfica 13 se muestra un ejemplo del error para una serie.



Gráfica 13. Ejemplo de una ST correspondiente al error.

El procedimiento para construir los modelos ARMA se realizara en dos etapas. Primero se construirá un modelo AR para la serie original, y posteriormente se considera la serie del error a_t a la que se le encontrará otro modelo AR. Con estas dos componentes se construye un modelo ARMA para la serie original. En ambos procedimientos, la etapa más importante es la de definir cuántos términos se requieren para cada modelo. Una descripción más detallada del procedimiento para construir los modelos ARMA se hará en la sección 4.4 de este capítulo.

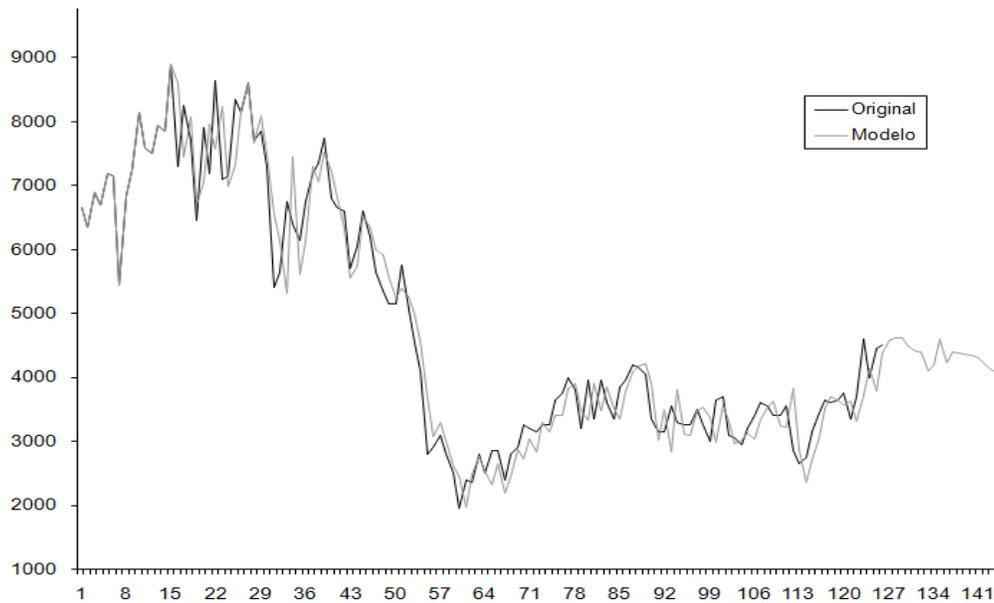
A partir de este capítulo se cambiará la notación para presentar los modelos ARMA de una serie. Para esto se indicará a que parte de la expresión de AR o MA corresponden, y las constantes ϕ_i y θ_j representarán los términos de la expresión correspondiente, es decir, no se escribirán los términos F_{t-i} ni a_{t-j} .

4.2 El fenómeno del Retardo en el Pronóstico.

Analizando las gráficas de los modelos construidos con esta metodología para los ejemplos de NN3-Complete, se detectó un fenómeno que visualmente aparece como si la gráfica del modelo fuera casi igual que la de la serie original, pero con un desplazamiento de una unidad a la

derecha. En el caso de NN3-Complete este fenómeno lo presentaron los 20 ejemplos: 51,64,66,74,80,82,83,84,85,86,88,89,90,91,92,95,100,105,107 y 109.

Dado que en la competencia los primeros 50 ejemplos correspondían a series de 50 valores (al parecer construidas por expertos), y los últimos 61 ejemplos eran series de 150 términos (al parecer de fenómenos reales), suponemos que el 34 % de los ejemplos reales de NN3 presentan este comportamiento. Una vez que se caracteriza el fenómeno del RP concluiremos que un gran porcentaje de series reales lo presentan por lo cual los modelos construidos con esta metodología darán mejores resultados al aplicarlos a estas series. A continuación presentamos en la gráfica 14 un ejemplo de este fenómeno correspondiente al modelo AR del ejemplo 74 obtenido con nuestra metodología.



Gráfica 14. Ejemplo 74 de NN3-Complete.

Este fenómeno lo hemos definido como *el retardo en el pronóstico (RP)*, pues es equivalente a predecir en un cierto momento lo que ocurrió en el momento anterior.

4.2.1 El procedimiento de Adelantar el Pronóstico.

El fenómeno del RP lo podemos aprovechar modificando la gráfica de los modelos lineales obtenidos al aplicar un desplazamiento de una unidad a la izquierda de su gráfica. A este procedimiento lo hemos definido como *adelantar el pronóstico (AP)* y lo formalizamos a continuación.

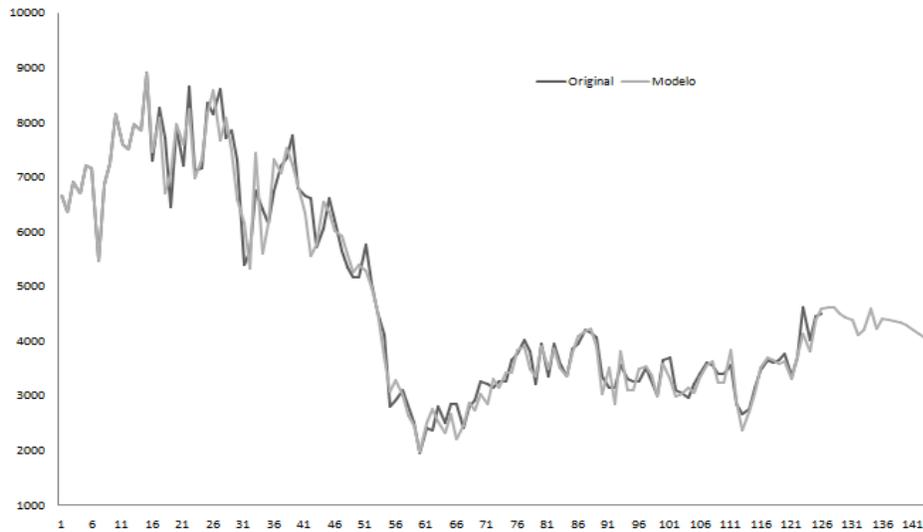
Definición: Sea una serie de tiempo con modelo AR o ARMA

$$F_t = \delta + \sum_{i=1}^p \phi_i Z_{t-i} + \sum_{j=1}^q \theta_j a_{t-j}$$

Definimos el adelanto en el pronóstico como la operación:

$$F_t = F_{t+1} \quad \text{para } t > \text{máximo } (p,q) \quad (18)$$

Cuando a un modelo AR o ARMA le aplicamos esta operación diremos que es un modelo lineal AR o lineal ARMA *con AP* respectivamente. En la gráfica se muestra el modelo lineal del ejemplo 74 con AP.



Gráfica 15. Ejemplo 74 de NN3-Complete a la que se le aplicó el adelanto en el pronóstico.

Un primer resultado que se tiene es que si a una serie que presenta RP se le aplica el AP, entonces el valor de RSS para estos modelos es menor que el de los modelos ARMA originales. Esto se origina por el hecho de que al desplazar la gráfica del modelo una unidad a la izquierda, que es lo que significa la operación (18), casi se encima con la gráfica de la serie original. Extrapolando este comportamiento a la región de pronóstico se espera que lo mismo pase y que los valores del modelo lineal con AP sean una mejor aproximación que los del modelo lineal. Esto último nos lleva a suponer que los modelos lineales con AP tendrán mejor capacidad de pronóstico. A manera de ejemplo, en la Tabla IV se muestra la mejora con los modelos lineales con AP para 10 de los ejemplos de NN3 que presentan RP.

Tabla IV
Comparación de RSS para modelos lineales y lineales con AP.

Ejem.	AR	RSS	RSS AP	% mejora
51	$0.9968 + 0.8129_1 + 0.6594_{12} - 0.4799_{13}$	3767.009	3379.4502	10.28
64	$0.3894 + 0.9199_1 + 0.6778_{12} - 0.5999_{13}$	3899.114	3495.0649	10.36
66	$0.9984 + 0.9202_1 + 0.5858_{12} - 0.5103_{13}$	3893.0544	2803.1406	27.99
74	$0.9993 + 0.9448_1 + 0.5226_{12} - 0.4800_{13}$	4894.1655	3340.5911	31.74
86	$-0.9991 + 0.6235_1 + 0.2161_2 + 0.1907_{17}$	4114.1499	1917.1523	53.40
88	$0.9979 + 0.7001_1 + 0.1531_{11} + 0.1438_{18}$	2449.5383	1265.7606	48.32
89	$0.9995 + 0.8914_1 + 0.2169_{12} - 0.1079_{13}$	1247.8290	339.8757	97.27
105	$1.9980 + 0.9099_1 + 0.3104_{11} - 0.2225_{13}$	1513.984	664.8109	56.08

La mejora en los modelos aquí presentados varía entre el 10.28% al 97.27% con un promedio de 48.48%, y es de esperarse que conforme el porcentaje sea mayor, la capacidad de pronóstico del modelo aumente en una proporción semejante. Cabe resaltar que cuando se tiene un modelo AR con 4 términos, es muy difícil mejorar substancialmente el valor del RSS, aumentando términos al modelo AR, o incluyendo términos de la parte de promedios móviles.

4.3 El Origen del Retardo en el Pronóstico.

Con el fin de analizar qué es lo que origina el fenómeno de RP se utilizó un algoritmo de clasificación basado en reglas [38]. Estos algoritmos se aplican a problemas donde se tiene un número finito de instancias agrupadas en varias clases y caracterizada por un conjunto de valores en ciertos atributos. El propósito de los mismos es encontrar reglas que establecen límites de valores para los atributos que definan, para cada instancia, la pertenencia a cada clase, con la idea de que después se puedan aplicar a instancias que no aparecían en un principio. Cabe señalar que si se quiere que el conjunto de reglas sea lo más aplicable posible, se requiere obtenerlo de un mayor número de ejemplos, donde éstos presenten el comportamiento más diverso posible.

Las reglas que aparecen en esta sección se han construido con un algoritmo llamado 2G el cual es presentado en el Anexo 1 y del cual soy coautor [11]. Dicho algoritmo presenta contribuciones originales, pues está basado en el uso de técnicas de reconocimiento supervisado de patrones en la etapa de preprocesamiento de la información, y construye las reglas utilizando árboles de decisión. Además, este algoritmo se ha probado en varios ejemplos previamente tratados en la literatura, y en todos los casos se han mejorado los resultados reportados. Una descripción de este algoritmo se dará en el anexo 1 de este documento.

El algoritmo 2G produce nuevo conocimiento que se representa por medio de reglas que tienen una estructura diferente que la obtenida por los árboles de decisión tradicionales porque este algoritmo usa diferentes atributos y, en la mayoría de los casos, le basta con dos condiciones para evaluar cada regla.

En nuestro caso construiremos dos conjuntos de modelos, uno que si presenta RP y otro que no y posteriormente aplicaremos el algoritmo 2G para obtener reglas que nos digan cuando un modelo si presenta RP y cuando no. Para desarrollar este procedimiento se tomaron los modelos de todos los ejemplos de NN3 que presentaban el fenómeno del RP, y lo primero que se observó es que si la serie original es $\{Z_t\}$, todos tenían en su componente AR al término Z_{t-1} . A todos estos primeros ejemplos se les consideró como miembros de una misma clase. Posteriormente se formó otra clase con los modelos de los ejemplos (a partir del número 50) que tenían el término Z_{t-1} pero que no presentaban el fenómeno de RP. Con esto se obtuvieron las dos clases que se utilizaron.

A continuación, después de revisar los ejemplos, se consideró que la aparición del fenómeno del RP estaba determinada por el tamaño de los valores de los coeficientes más grandes del modelo, y con esta idea como base, para cada ejemplo, se construyó un vector con tres componentes. La primera, que llamaremos *yesterday*, era el valor del coeficiente de Z_{t-1} , la segunda que llamaremos *other*, era el valor absoluto más grande de todos los términos del modelo AR (salvo los términos δ y Z_{t-1}). La tercera componente era 1 si presentaba el fenómeno RP, y 0 en caso contrario. Al aplicar el algoritmo 2G se obtuvieron las siguientes 6 reglas:

if yesterday <=0.4673 and other <=0.8118 then 0
 if 0.4672<=yesterday <=0.5616 and other <=0.2602 then 1
 if 0.5615<=yesterday <=0.5833 and 0.3366 <= other <=0.6277 then 0
 if 0.5832<=yesterday <=0.6519 and other <=0.2602 then 1
 if 0.6518<=yesterday <=0.6789 and 0.2947 <= other <=0.6277 then 0
 if 0.6788 <=yesterday and other <=0.6613 then 1

A continuación depuramos estas reglas tomando en cuenta los rangos de valores de las variables *yesterday* y *other* y el comportamiento del fenómeno con lo cual reducimos a las siguientes reglas que determinan cuando se presenta RP en el conjunto de ejemplos de NN3. Cabe señalar que un conjunto de reglas que caractericen el fenómeno en más ST, requiere que este procedimiento se aplique con conjuntos más grandes, y que tengan ejemplos más diversos de comportamiento de ST que los que se presentan en NN3.

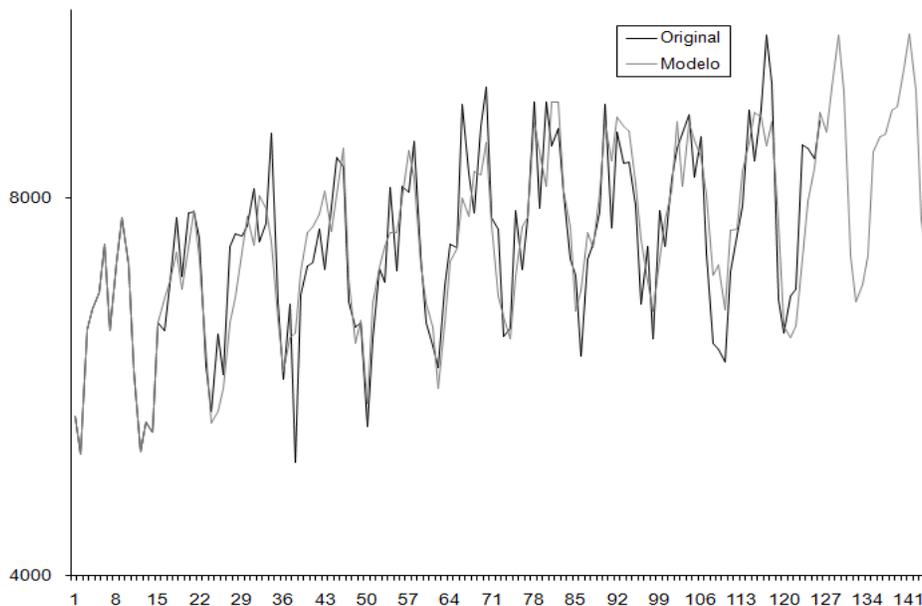
if 0.6788 <=yesterday and other <=0.6613 then 1 (19)
 if 0.4674 <= yesterday <=0.6519 and other <=0.2602 then 1.

Estas reglas en el fondo indican que el proceso RP se va a presentar cuando el valor actual del proceso sea muy semejante al valor anterior *yesterday* del mismo.

Es de señalar que es importante que el modelo que presenta el RP se ha obtenido por el procedimiento de optimización aquí señalado, pues si no es así puede tenerse un modelo con RP, pero con poca capacidad de predicción, como es el caso del ejemplo 55 de NN3, donde la solución óptima que encontramos es:

$$\text{AR: } 0.9995 + 0.1825_1 + 0.1495_{11} + 0.6854_{12}$$

Cuya gráfica aparece a continuación:

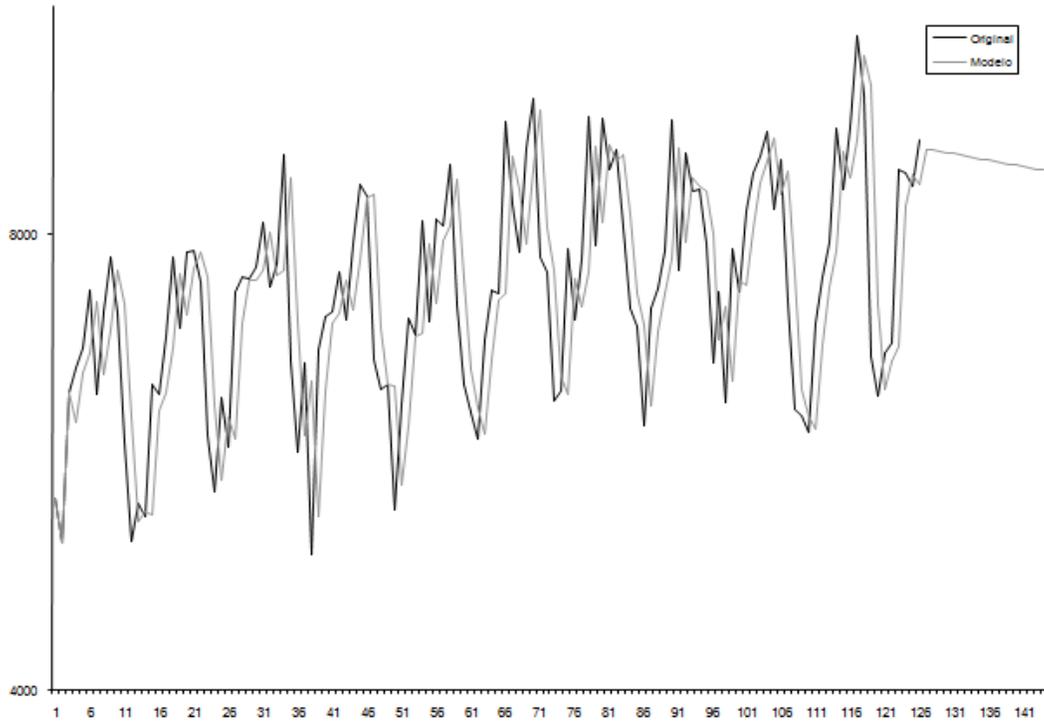


Gráfica 16. Ejemplo 55 de NN3 con modelo AR: $0.9995 + 0.1825_1 + 0.1495_{11} + 0.6854_{12}$

tiene un valor de RSS igual a 4938.354 y si reproduce los patrones de comportamiento del proceso y otra solución AR para el mismo problema:

$$\text{AR: } 23.8735 + 0.7988_1 + 0.1969_2$$

Con un valor de RSS igual a 8612.835 y que aparece en la Gráfica 17 si presenta el fenómeno de RP pero pierde la información sobre los patrones de comportamiento de la serie por lo lejano que esta del óptimo y en consecuencia no es útil para efectos de pronósticos.



Gráfica 17. Ejemplo 55 de NN3 con modelo AR: $23.8735 + 0.7988_1 + 0.1969_2$

Por último, es importante resaltar que el fenómeno del RP se puede presentar con modelos obtenidos con otras metodologías para construcción de modelos lineales, pero al parecer con nuestra metodología su aparición es más frecuente, debido quizás, a los límites que se establecen para las variables del modelo. Por otra parte, por la caracterización que origina que aparezca el fenómeno del RP, es de esperarse que se presente mucho en series que aparezcan en economía, finanzas, fenómenos físicos y químicos, etc., en donde el valor en un cierto instante esta muy fuertemente determinado por el valor que se presentó en el instante anterior.

4.4 Comparación con otras metodologías.

Para evaluar el desempeño de un modelo sobre una ST se dividen los datos en dos conjuntos llamados de entrenamiento y de prueba. El conjunto de entrenamiento tiene los primeros valores de la serie (aproximadamente el 90% del total) y el de prueba los últimos 10%. La información del conjunto de entrenamiento se utiliza para elegir el modelo y evaluar los parámetros. Una vez

elegido el modelo correspondiente se evalúa su capacidad de pronóstico sobre el conjunto de prueba, y cuando se tienen distintas propuestas de modelos, es común elegir al que mejor resultado dé sobre el conjunto de prueba. Para esta evaluación se pueden utilizar varias medidas de desempeño [22]. En nuestro caso utilizaremos de preferencia RSS.

Para construir los modelos con nuestra metodología se procede de la siguiente forma.

- 1) En esta primera etapa se calcula la parte AR del modelo. Para esto, a partir de $K=2$ se construyen modelos AR con K términos y se prueba el desempeño sobre el conjunto de prueba. Una vez que se tiene el primer valor de K donde el RSS del modelo es menor que los valores obtenidos para los términos $K-1$ y el $K+1$ se considera que la parte AR del modelo tiene los K términos ya encontrados y se pasa a la segunda etapa.
- 2) Se calcula la serie del error obtenido entre la serie original y la serie calculada por el modelo obtenido en la etapa anterior. A esta nueva serie se le aplica el mismo procedimiento que aparece en el punto anterior y se obtiene la parte correspondiente a la componente de promedios móviles MA del modelo ARMA. Pudiera darse el caso de que al incluir las componentes MA del modelo se tengan peores aproximaciones en el conjunto de prueba que las obtenidas con la parte AR. En este caso el modelo solo tendría componente AR.
- 3) Se revisan si el modelo AR obtenido en la etapa 1) cumple con las reglas (19) que establecen que se presenta el fenómeno RP y en caso de ser así es posible realizar el corrimiento de la gráfica una unidad a la izquierda según se señala en (18) siempre y cuando con este procedimiento se mejore el resultado.

En el anexo 2 se presenta una descripción de este procedimiento.

Los resultados obtenidos con esta metodología al compararse con otros modelos obtenidos con otras metodologías se detallan en los siguientes párrafos. En particular se hacen comparaciones con las metodologías presentadas en [8] y [18].

Para realizar una prueba del desempeño de los modelos que aparecen en [18], se utilizarán las series: A, B, C, D, E y F, que aparecen en el libro de BJ [4], utilizadas y presentadas en el capítulo 3.

En [18] se presentan los resultados de construir varios modelos lineales para estas series. El primero es el clásico de BJ, y los demás se aplican cuando los modelos de BJ no cumplen con el postulado de que el error es un ruido blanco. En [30] se señala que el uso de estos últimos modelos mejora entre un 8% y un 13% la capacidad de predicción del modelo cuando el error no es un ruido blanco. Enseguida presentamos la relación de estas metodologías para modelos lineales

- Modelo ARIMA standard. Aquí se aplica la metodología tradicional de BJ donde las componentes principales son los modelos autorregresivos con promedios móviles que son lineales en la serie de tiempo $\{Z_t\}$, y el ruido blanco $\{a_t\}$ [4].
- Mínimos cuadrados ordinarios (OLS por sus siglas en inglés *Ordinary Least Squares*). Se usa cuando la distribución del error presenta el problema de leptocurtosis y permite disminuir el error en la predicción [21].

- Menor Desviación Absoluta (LAD por sus siglas en inglés *Least Absolute Deviation*). Se usa minimizando la suma de los valores absolutos en lugar de la suma de cuadrados. Esto se hace para reducir la influencia de los errores extremos [21].
- Distribución-t Generalizada (GT por sus siglas en inglés *Generalized t-distribution*). Aquí se minimiza la función objetivo con respecto a los parámetros pero suponiendo que el error tiene una distribución t [29].
- Distribución exponencial beta generalizada de segunda clase (EGB2 por sus siglas en inglés *Exponential Generalized beta distribution of the second kind*). Aquí se supone que los errores tiene una distribución de este tipo [29].

Adicionalmente en [18] se presentan los resultados de dos modelos de redes neuronales, uno heurístico (Heuristic NN), y otro basado en Algoritmos genéticos (GANN) que vienen implementados en el software comercial BioComp System's NeuroGenetic Optimizer®.

Para hacer comparaciones con los modelos anteriormente señalados, se usarán los mismos tamaños del conjunto de entrenamiento, y de los conjuntos de prueba que aparecen en [18], en donde si el número de elementos de la serie es mayor que 100, los tamaños de conjuntos de prueba se establecen como 10. En el caso de que sean menores o iguales a 100, los conjuntos de prueba tendrán tamaño 5. El tamaño de los conjuntos de entrenamiento es el tamaño de la serie original menos el número de elementos del conjunto de prueba.

Con nuestra metodología se obtuvieron los modelos de la tabla V, donde para cada ejemplo se presenta la componente AR y en caso de ser necesario la MA. Cabe señalar que cuando se indica en la última columna de la tabla "AP" se aplicó el corrimiento presentado en (18).

Tabla V
Soluciones a los problemas de Box Jenkins.

Serie	AR	MA	AP
A	$1.1035 + 0.5648_1 + 0.1919_6 + 0.1245_9 + 0.0544_{13}$	$-0.2271_1 + 0.1046_2 + 0.0514_3$	no
B	$0.8302 + 1.1274_1 - 0.1685_2 + 0.0644_4 - 0.0258_6$	$0.0460_2 - 0.0576_5 + 0.1381_6 + 0.0587_7$	si
C	$.8425_1 - 0.8488_2$		no
D	$0.7609 + 0.8997_1 + 0.0511_{12} - 0.0335_{16}$		si
E	$1.9993 + 1.0051_1 - 0.2590_3 + 0.1538_{10}$		si
F	$1.9996 + 0.6555_2 + 0.2938_3$		no

En las tablas siguientes se muestran los resultados de las distintas metodologías presentadas en [18] y las obtenidas con nuestros algoritmos. En la tabla VI se usa como criterio de comparación la suma de valores absolutos de los errores y en la tabla VII la suma de cuadrados de los errores. Los resultados de nuestros modelos se presentan en la línea que dice "Lineal AP" (AP por las siglas de *Adelanto en el Pronóstico*) y el lugar que se obtiene al compararse con los otros modelos aparece en la línea "lugar". Cabe señalar que en cada grupo de comparaciones, salvo en un ejemplo, los resultados obtenidos con nuestra metodología son mejores que los

obtenidos con los métodos estadísticos comparados y además tienen buenos resultados al compararse con los que utilizan redes neuronales.

Tabla VI
Comparación de modelos respecto a suma de valores absolutos de errores.

	Serie A	Serie B	Serie C	Serie D	Serie E	Serie F	Serie G
Lineal AP	3.9	75	5.9	2.87	87	46	173
Heuristic NN	4.519	88.312	9.138	2.942	98.873	43.966	
GA NN	3.705	72.398	6.684	2.952	69.536	36.4	
ARIMA ML	4.005	78.855	11.247	3.114	116.485	49.161	
OLS	3.937	83.17	10.74	3.08	114.8	45.5	
LAD	3.96	79.47	10.3	3.066	117.6	44.46	
GT	3.937	80.68	10.25	3.064	106.5	44.59	
EBG2	4.017	81.01	10.3	3.066	111.8	44.5	
LUGAR	1	2	1	1	2	6	

Tabla VII
Comparación de modelos respecto a suma de cuadrados de errores.

	Serie A	Serie B	Serie C	Serie D	Serie E	Serie F	Serie G
Lineal AP	2.33	797	7.9	1.519	2399	750	3545
Heuristic NN	3.029	1170.352	10.977	1.607	2864.098	1025.026	
GA NN	2.104	787.691	6.413	1.269	1579.876	600.1	
ARIMA ML	2.455	938.864	16.877	1.506	4055.694	799.42	
OLS	2.371	988.2	15.32	1.506	3094.8	956.5	
LAD	2.398	911.4	14.05	1.488	3224.2	975	
GT	2.371	939.5	13.97	1.485	2689.3	973.4	
EBG2	2.473	947.1	14.05	1.488	2942.2	975.9	
LUGAR	2	2	2	6	2	2	

A continuación se presentan los resultados de comparar nuestro método con los reportados en [8]. En dicho artículo se comparan las metodologías:

- Metodología de Holt-Winters. Esta metodología es muy utilizada por su simplicidad y la precisión de sus pronósticos sobre todo con series de tiempo periódicas. Está basada en cuatro ecuaciones básicas que representan la regularidad, tendencia, periodicidad y pronóstico de la serie [7].
- Metodología de Box-Jenkins, que ya ha sido ampliamente comentada en secciones anteriores [4].
- Método de pronósticos evolutivos. Que es una metodología basada en programación evolutiva [8].
- Algoritmos meta evolutivos. Es una meta heurística que utiliza dos niveles de arquitectura, en el primero se elige el modelo ARMA de que se trate y en el segundo se estiman los parámetros correspondientes [8].

Para realizar una prueba del desempeño de los modelos, se utilizaron algunas de las series que aparecen en [23] que se denominan como: *Passengers*, que es una serie (144 datos) que representa el número de pasajeros mensuales en una aerolínea. *Paper*, esta serie (120 datos) representa la venta mensual de papel en Francia. *Deaths*, que es una serie (169 datos) que representa las muertes y daños en carreteras de Alemania. *Maxtemp*, representa las temperaturas máximas (240 datos) en Melbourne, Australia. *Chemical*, que es una serie (198 datos) de lecturas de concentraciones de un químico en un reactor. Los conjuntos de entrenamientos de estas series contienen el 90% de los datos y los 10% restantes están en el conjunto de prueba.

Con nuestra metodología se obtuvieron los modelos que aparecen en la tabla VIII. Cabe señalar que de estos ejemplos ninguno presenta RP.

Tabla VIII
Soluciones con nuestra metodología.

Serie	AR	MA
Passengers	$1.3400 + 0.9087_1 + 1.0612_{12} - 0.9633_{13}$	
Paper	$6.2323 + 0.9583_{12}$	
Deaths	$1.9941 + 0.9053_{12} + 0.0832_{14}$	$4.3636 + 0.405_2$
Maxtemp	$0.7046 + 0.3362_1 - 0.0668_7 + 0.4060_{11} + 0.2910_{12}$	$18.662 + 0.1045_5 - 0.1857_{11} - 0.1930_{12}$
Chemical	$0.5419 + 0.6081_1 + 0.3753_7 - 0.0144_{15}$	

En la tabla IX se comparan los resultados para estas ST. Los resultados de nuestros modelos se presentan en la columna que dice "Lineal AP" y el lugar que se obtiene al compararse con los otros modelos aparece en la columna "lugar".

Tabla IX
Comparación con otras metodologías.

Series	Holt Winter	Box-Jenkins	Heuristic		Lineal	
			Evolutionay	Meta-Evolutionay	AP	Lugar
passengers	16.5	17.8	21.9+ -1.2	17.2+ - 0.2	16.3	1
paper	49.2	61	60.2+ - 2.2	52.5+ - 0.1	5.59	1
deaths	135	144	135.9+ - 1.7	137+ - 2	140	3
maxtemp	0.72	1.07	0.95+ - 0.02	0.93+ - 0.4	0.94	2
chemical	0.35	0.36	0.36+ - 0.0	0.34+ -0.0	0.34	1

De los resultados presentados en las tablas de esta sección, concluimos que los modelos construidos con nuestra metodología superan a todos los modelos obtenidos con los métodos estadísticos, y son competitivos con los métodos no lineales aquí presentados. Además, cabe agregar que esta metodología es totalmente automatizable y permite modelar ST que otras metodologías tradicionales no pueden.

Por otra parte, es de resaltar es que el fenómeno de RP debe de aparecer en muchas ST en áreas como finanzas, economía, industria etc., en donde el valor actual del proceso que modela la ST es muy parecido al valor anterior. Además, el hecho de conseguir nuevos modelos lineales AP

con mejor capacidad de aproximación, y que sean aplicables en muchas situaciones del mundo real (finanzas, economía, industria, etc.), permite suponer que en estas áreas los modelos lineales AP pudieran convertirse en mejores opciones que los tradicionales modelos lineales.

El fenómeno de RP no es exclusivo de esta metodología, pues se podría presentar con otras técnicas para obtención de modelos lineales que den resultados que cumplan con las reglas (19). Como un ejemplo de esto se tiene que los modelos construidos por BJ para sus serie B y D (no se presentan aquí) presentan este fenómeno y también se presenta en uno de los ejemplos que aparecen en [24]. También es de señalar que habría que hacer un estudio con más ejemplos del fenómeno para obtener reglas que lo caractericen mejor.

Capítulo 5

Modelos Autorregresivos de Funciones de Transferencia Discretas

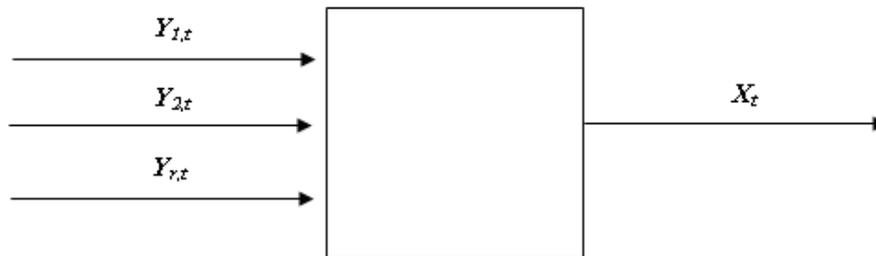
En este capítulo se presenta una propuesta metodológica para construir buenos modelos lineales AR para las FTD de un proceso. Para encontrar dichas funciones, es necesario resolver otro problema de optimización no lineal que permita calcular el valor de una de las variables de salida en cualquier instante de tiempo, como una combinación lineal de los valores históricos de las variables de entrada y posiblemente de la misma salida. De igual manera que en el caso de ST, es necesario primero definir cuantas variables y cuales son las más importantes para resolver el problema utilizando una técnica de optimización adecuada.

En este capítulo se presenta un algoritmo heurístico para calcular nuestro equivalente a modelos AR para FTD. La herramienta principal que se utilizara para esto es una modificación del algoritmo presentado en la sección 3.3.2 para la construcción de modelos AR de ST. Además, se muestran los resultados de modelar tres ejemplos de FTD con esta metodología. En especial es importante resaltar la mejora de resultados que obtenemos con nuestra metodología al compararlos con una metodología de lógica difusa en un problema ya tratado en la literatura [21] que aquí presentamos. Cabe señalar que en estos tres ejemplos siempre se encuentran soluciones que presentan un fenómeno similar al RP pero para FTD.

5.1 Modelos Autorregresivos.

Respecto a la construcción de funciones de transferencia, existen metodologías clásicas tratadas en los textos de control que sobre todo están basadas en la aplicación de leyes físicas o utilizan identificación experimental. Un buen texto basado en aspectos estadísticos para construcción de modelos lineales de las funciones de transferencia es [4].

Antes de presentar el algoritmo para la construcción de modelos AR para FTD recordaremos que en la construcción del algoritmo de la sección 3.3.2 se tenían dos etapas principales. En la primera se estimaban cuántos términos era necesario considerar para capturar el comportamiento de la ST, y posteriormente se aplicaban los AGA en varias etapas para buscar las componentes más importantes, y en esta última etapa, los límites de todas las variables se tomaban en el intervalo $[-1,+1]$. A continuación describiremos el procedimiento para construir modelos AR para una FTD, que modela situaciones del tipo que aparece en la gráfica 18 (varias entradas, una salida), y que es similar al anteriormente comentado, donde se tienen r entradas y el valor de la entrada i en el momento t es $Y_{i,t}$. El valor de la salida en el momento t es X_t .



Gráfica 18. Sistema a modelar.

Para construir nuestro modelo debemos de tener una idea primero del número p de términos en la historia del proceso que consideramos que son importantes. Ya con esto buscamos construir un modelo de la forma:

$$F_t = \delta + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{j=1}^r \sum_{i=1}^p s_{ji} Y_{j,t-i}$$

Donde el pronóstico de la salida en el momento t es F_t , y depende de los p valores anteriores de X y de Y_1, Y_2, \dots, Y_r .

Para encontrar nuestro modelo aplicamos un algoritmo análogo al desarrollado en la sección 3.3.2 que se utilizará para encontrar un modelo con K términos para un problema de $p(r+1)+1$ incógnitas donde r es el número de entradas del proceso de que se trate y que tiene una sola salida. La única variación es que en este caso no se tiene algún resultado que nos sirva de base para establecer como límite de las variables el intervalo $[-1,+1]$, por lo cual, después de realizar varias pruebas se encontró que el intervalo $[-10,+10]$ daba buenos resultados. También en este caso, al igual que en toda la tesis, la función a optimizar es RSS.

Por el gran número de variables disponibles para construir nuestro modelo de pronósticos, resulta que este problema tiene muchas soluciones distintas, pero con casi el mismo valor de RSS. Por lo anterior se propone utilizar los siguientes criterios para elegir el modelo más adecuado.

5.1.1 Elección del modelo Autorregresivo.

Seleccionar al modelo que:

- Tenga el menor número de términos de los valores de la salida.
- Tenga el mayor número de términos en las entradas que están más correlacionadas con la salida.
- Los términos de las variables más correlacionadas con la salida sean los más recientes posibles.

Si visualmente el modelo construido presenta el fenómeno del RP, se puede aplicar un procedimiento análogo al definido en (15) para tener un nuevo modelo lineal. Una indicación sobre si presenta este fenómeno o no la puede dar el valor del primer término no

constante de la componente AR del modelo. Si esta componente corresponde al primer termino del retardo y tiene valor alto, esto pudiera ser un indicador de que se presenta el fenómeno de RP.

5.2. Ejemplos de modelos lineales para Funciones de Transferencia Discretas.

A continuación veremos algunos ejemplos donde se construyen modelos lineales AR para FTD. Para estimar la precisión de los modelos se calcula el porcentaje de error [22], que se obtiene a partir de la formula:

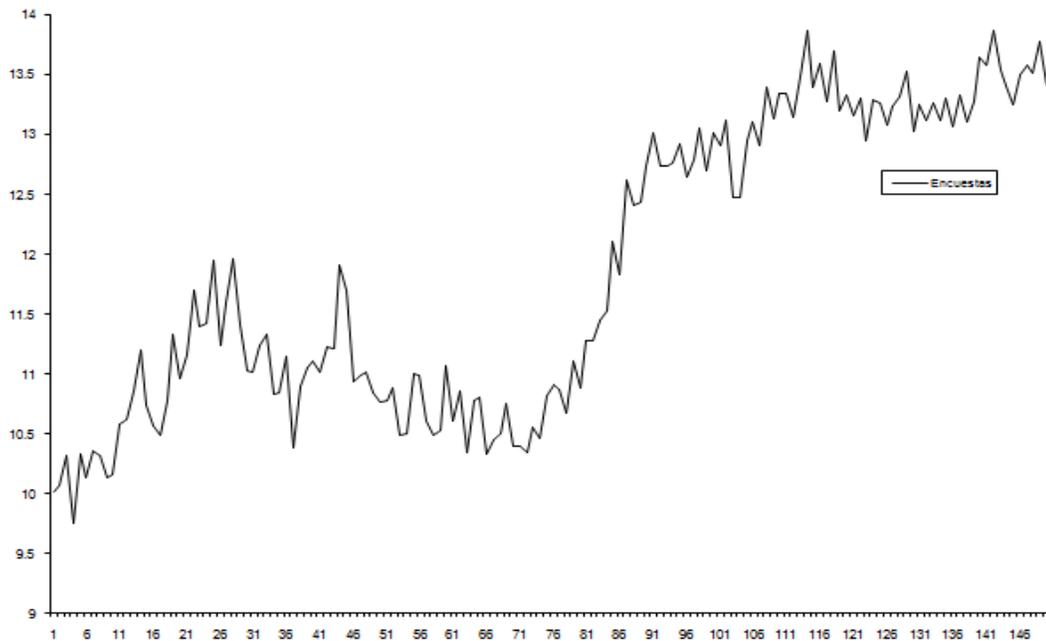
$$\frac{1}{N} \sum \left| \frac{F_t - X_t}{X_t} \right| \times 100\%$$

Donde N es el tamaño del conjunto de prueba. Dicha fórmula la aplicamos en los dos primeros ejemplos calculándola antes y después de realizar el adelanto en el pronóstico.

5.2.1 Primer ejemplo.

Este ejemplo corresponde a los 150 datos de la tabla M que aparece en [4] y corresponden a estimar las ventas diarias de un cierto producto en base a los resultados de unas encuestas.

A continuación aparece la gráfica de resultados de las encuestas:



Gráfica 19. Resultados de las encuestas.

Para calcular el modelo AR de este problema se tomara $p=10$ es decir se considera que la venta en un cierto día es función de las ventas y los resultados de las encuestas de los diez días anteriores. Se busca construir un modelo AR que a lo más tenga $K=6$ términos. Después de aplicar el algoritmo pidiendo 5 soluciones se obtuvieron los resultados:

$$\begin{array}{r}
 2.1099 + 1.5527_1 - 0.5610_2 + 1.8731_{1,0} - 0.8326_{1,4} - 2.5679_{1,9} \\
 0.9084 + 0.9980_1 + 0.8680_{1,2} - 0.0802_{1,3} - 0.0631_{1,5} - 0.3814_{1,7} \\
 -1.1815 + 1.2778_1 - 0.2718_3 + 3.6173_{1,0} + 3.0969_{1,5} - 3.1691_{1,7} \\
 1.1461 + 0.9969_1 - 3.6855_{1,1} - 0.5012_{1,4} - 2.4859_{1,5} - 3.2005_{1,7} \\
 0.5842 + 0.7502_1 + 0.2496_2 - 0.2062_{1,0} - 0.4759_{1,3} + 0.4733_{1,6}
 \end{array}
 \quad
 \begin{array}{r}
 16.865 \\
 17.322 \\
 16.293 \\
 17.332 \\
 19.195
 \end{array}$$

En donde los subíndices i,j representan al valor de $Y_{i,t-j}$ y el subíndice k representa el valor de X_{t-k} .

Primero se probó la solución con menor valor de RSS pero que visualmente no presenta el fenómeno de RP:

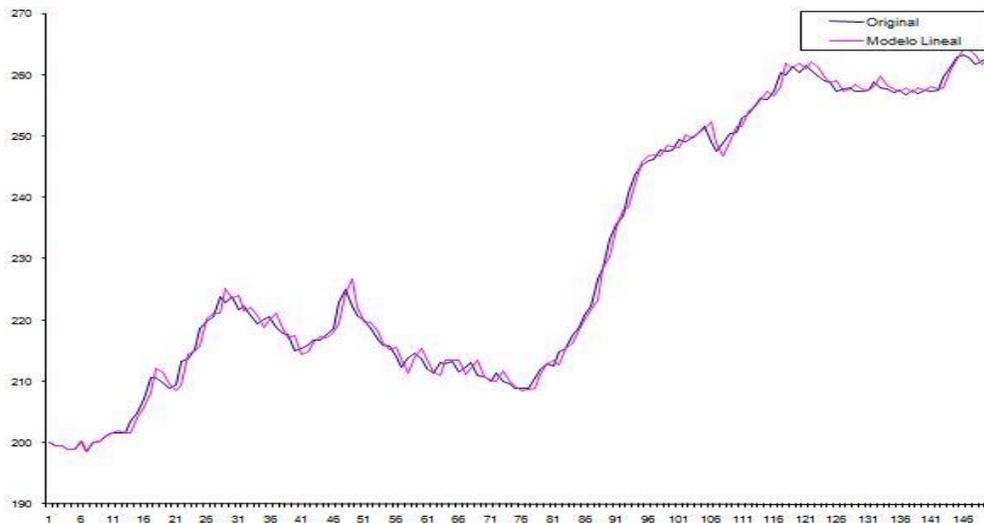
$$-1.1815 + 1.2778_1 - 0.2718_3 + 3.6173_{1,0} + 3.0969_{1,5} - 3.1691_{1,7} \quad 16.293$$

Con un valor de RSS igual a 16.293 y un porcentaje de error de 0.4374. Después de adelantar el pronóstico los valores fueron RSS igual a 9.5582 y porcentaje de error 0.2586.

Después se probó la solución:

$$1.1461 + 0.9969_1 - 3.6855_{1,1} - 0.5012_{1,4} - 2.4859_{1,5} - 3.2005_{1,7} \quad 17.332$$

Que si presenta visualmente retardo en el pronóstico. La gráfica original tenía un valor de RSS igual a 17.3352 y un porcentaje de error de 0.4647. Después de aplicar el adelanto del pronóstico el valor de RSS fue de 5.1740 con un porcentaje de error de 0.1744. A continuación se presenta la gráfica de las ventas y del modelo sin aplicar retardo en el pronóstico.

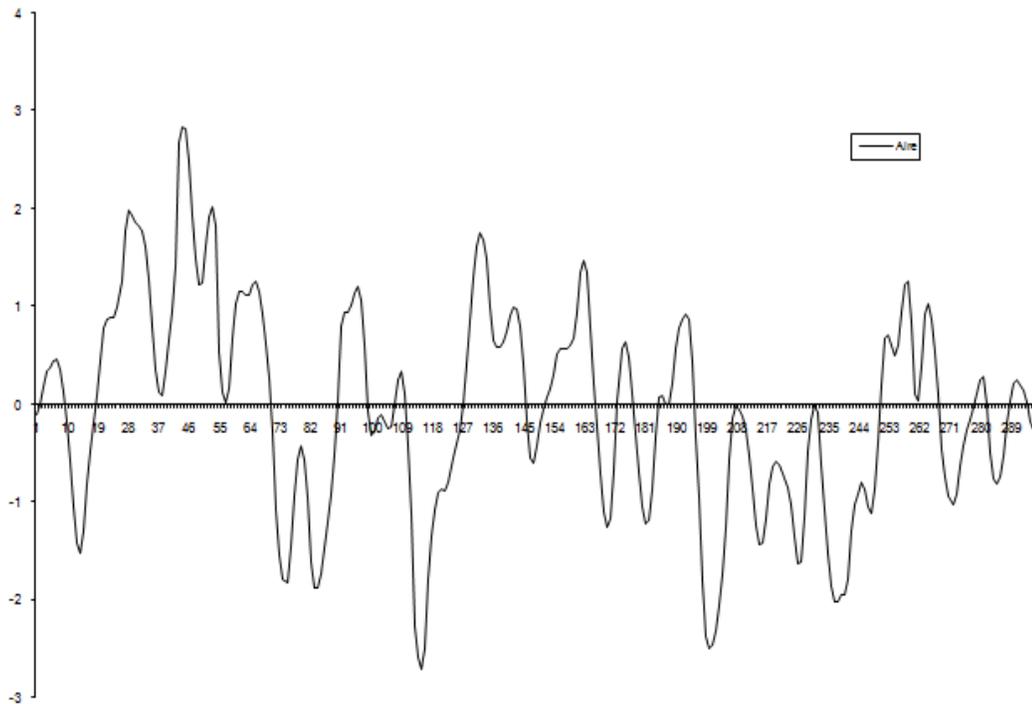


Gráfica 20. Ventas diarias de un producto.

5.2.2 Segundo ejemplo.

Los 296 datos de este ejemplo corresponden a la tabla J que aparece en [4]. Estos datos corresponden a un reactor químico donde se tiene una entrada que es la velocidad de alimentación de un gas medida en cu. ft./min. y la salida es el %CO₂ en los gases de salida. El problema es construir un modelo AR que prediga los valores del %CO₂.

A continuación aparece la gráfica de la velocidad de entrada del gas:



Gráfica 21. Velocidad de entrada del gas.

Para calcular el modelo AR de este problema se tomara $p=10$ es decir se considera que el %CO₂ en un cierto minuto es función de las velocidades de entrada del gas de los diez minutos anteriores. Para este problema se busca construir un modelo AR que a lo más tenga $K=6$ términos. Después de aplicar el algoritmo pidiendo 5 soluciones se obtuvo el resultado:

$$\begin{array}{l}
 1.3639 + 0.9749_1 - 0.1455_{1,0} - 0.2011_{1,2} + 0.0475_{1,6} + 0.0464_{1,8} \\
 3.6344 + 0.9997_1 - 0.0674_5 - 2.3234_{1,2} + 3.6442_{1,7} - 0.4263_{1,8} \\
 1.4582 + 0.9731_1 + 0.5577_{1,2} + 0.6037_{1,4} + 1.0197_{1,5} + 2.9946_{1,7} \\
 0.5742 + 2.2672_1 - 1.2780_2 + 0.5546_{1,5} + 2.0781_{1,8} + 4.1985_{1,9} \\
 0.4304 + 1.4559_1 - 0.4639_2 - 3.2584_{1,2} - 3.8913_{1,4} - 0.1004_{1,5}
 \end{array}
 \begin{array}{l}
 12.732 \\
 11.697 \\
 12.732 \\
 8.965 \\
 8.452
 \end{array}$$

Primero se probó la solución:

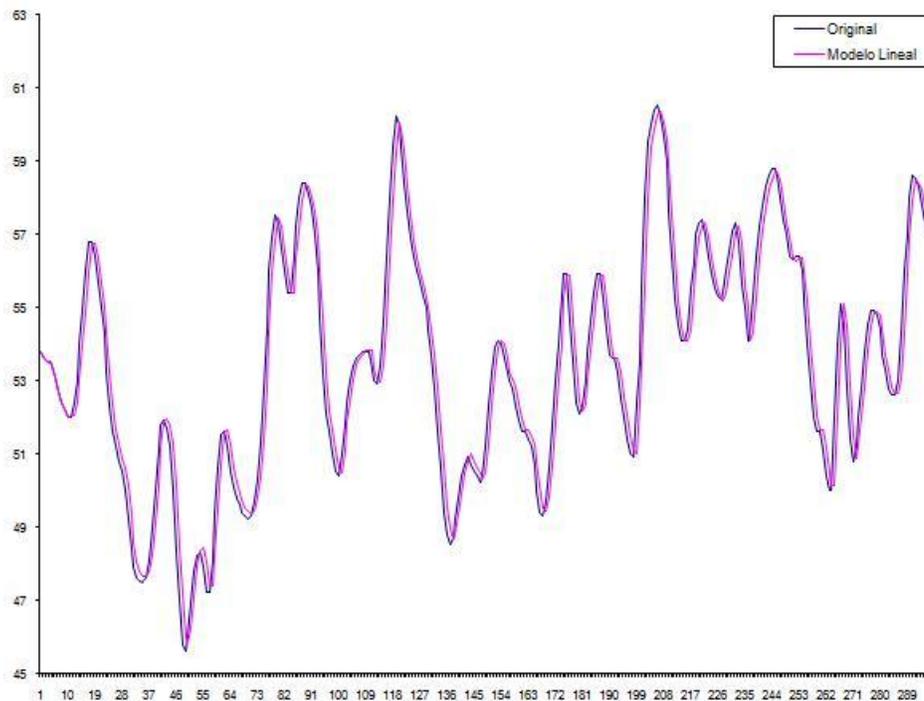
$$0.4304 + 1.4559_1 - 0.4639_2 - 3.2584_{1,2} - 3.8913_{1,4} - 0.1004_{1,5} \quad 8.452$$

Que no presenta visualmente RP, pero que al aplicarle el adelanto del pronóstico nos da una gráfica con RSS igual a 5.9085 y porcentaje de error de 0.5086

Por otra parte, con la solución:

$$1.3639 + 0.9749_1 - 0.1455_{1,0} - 0.2011_{1,2} + 0.0475_{1,6} + 0.0464_{1,8} \quad 12.732$$

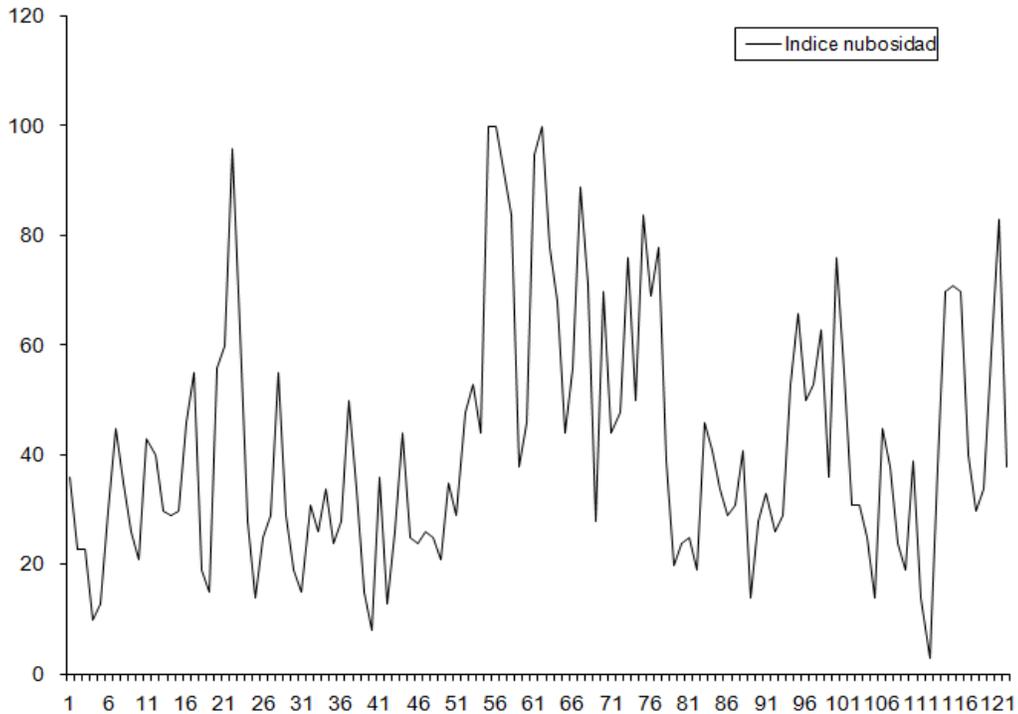
Que si presenta RP, al aplicarle la operación de adelanto se obtuvo una gráfica con RSS de 1.5236 y un porcentaje de error de 0.1259. En la gráfica siguiente aparece la solución que presenta RP.



Gráfica 22. %CO₂ en los gases de salida.

5.2.3 Tercer ejemplo.

Este ejemplo está tomado de [25] y consiste en modelar la FTD de un proceso en donde la entrada es el índice de nubosidad promedio y la salida es la temperatura promedio ambas medidas diariamente en la ciudad de Taipei en Taiwán durante el periodo del 1 de Junio al 30 de Septiembre del 2006. Este problema ya había sido tratado antes en la literatura [41] y en [20] mejoran los resultados obtenidos en [41]. En la gráfica 23 se presenta la variación del índice de nubosidad.



Gráfica 23. Índice de nubosidad diario.

Para calcular el modelo AR de este problema se tomara $p=10$ es decir se considera que la temperatura en un cierto día es función de la temperatura y la nubosidad de los diez días anteriores. Además, se busca construir un modelo AR que a lo más tenga $K=6$ términos. Después de aplicar el algoritmo pidiendo 5 soluciones se obtuvo el resultado:

$$\begin{array}{r}
 5.1314 + 0.8184_1 - 1.6563_{1,2} - 1.7819_{1,4} - 1.9469_{1,6} + 1.6118_{1,7} \quad 9.868 \\
 4.6164 + 0.8369_1 - 1.9952_{1,0} - 1.2249_{1,4} + 1.8506_{1,8} + 0.3868_{1,9} \quad 9.863 \\
 4.5350 + 0.8392_1 - 4.3708_{1,0} - 2.6610_{1,1} + 1.7124_{1,5} + 0.1693_{1,8} \quad 9.861 \\
 5.4569 + 0.8068_1 + 4.5787_{1,2} + 2.4489_{1,5} - 2.6078_{1,6} + 1.8640_{1,7} \quad 9.876 \\
 4.2705 + 0.8507_1 - 3.5614_{1,1} - 2.6190_{1,5} - 1.2269_{1,6} + 3.9375_{1,8} \quad 9.885
 \end{array}$$

Donde los subíndices indican la entrada y el coeficiente correspondiente.

Para elegir el modelo más conveniente aplicaremos las reglas de la sección 5.1.1. Respecto a las primeras dos reglas todos los modelos son igualmente elegibles pero respecto a la tercera, el modelo más conveniente es el:

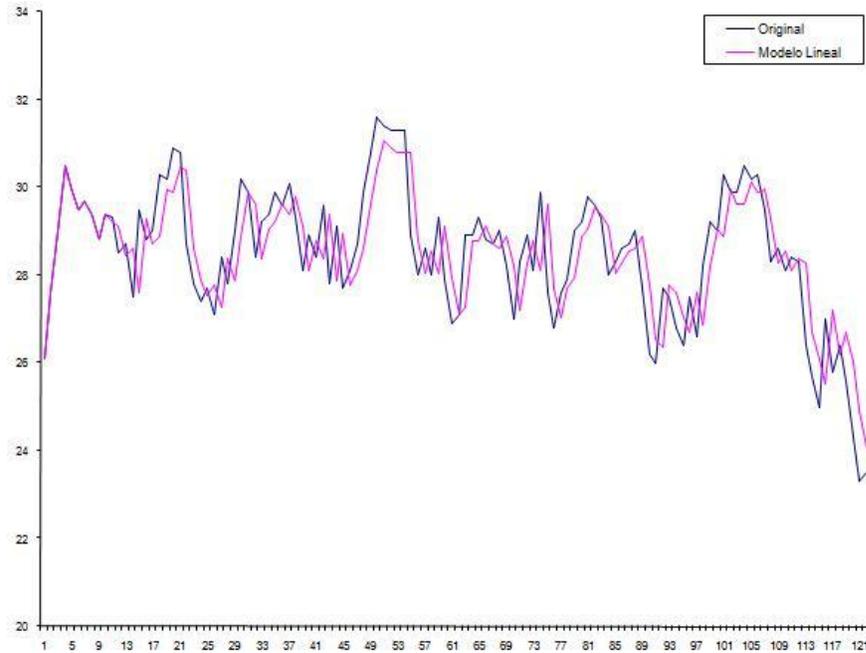
$$4.5350 + 0.8392_1 - 4.3708_{1,0} - 2.6610_{1,1} + 1.7124_{1,5} + 0.1693_{1,8} \quad 9.861 \quad (20)$$

Por ser el que tiene variables más recientes. Cabe señalar que pudiera haber otros modelos con distintas variables que nos dieran exactamente el mismo valor de RSS. Como un ejemplo de esto se tiene que

$$4.5587 + 0.8384_1 + 0.9863_{1,0} - 2.0666_{1,1} - 1.6692_{1,2} - 3.4359_{1,3} + 0.5499_{1,7} \quad 9.861$$

Nos da los mismos valores de RSS con una precisión de tres cifras decimales.

Con el modelo (20) se tiene la siguiente gráfica para el pronóstico de temperatura que presenta el fenómeno de retardo en el pronóstico.



Gráfica 24. Temperatura diaria.

En [25] se analiza este problema y se le construyen modelos difusos para ST de alto-orden (High-Order) de órdenes del 1 al 8, concluyendo que los mejores resultados se obtienen con modelos de segundo orden. Para compararnos con los resultados del artículo, haremos una estimación del porcentaje de error, para los meses de Junio a Septiembre. Mientras menor sea el porcentaje de error mejor es el modelo.

En la Tabla X se presenta una comparación de los porcentajes de error del método que denotaremos por TFHOFTS (por las siglas de *Two-Factors High-Order Fuzzy Time Series*), con el método que desarrollamos con nuestra metodología que denotaremos OA (por las siglas de *Our Algorithm*)

Tabla X
Porcentaje de errores mensuales.

Meses	TFHOFTS	OA
Junio	0.80	0.37
Julio	0.96	0.67
Agosto	1.07	0.48
Septiembre	1.01	0.95

La comparación del mes de Junio no es muy significativa, pues en este caso nuestro modelo empieza a pronosticar a partir del noveno día, y el otro modelo a partir del cuarto. Además, la comparación del mes de Septiembre, en nuestro caso es sobre 29 días por el fenómeno del RP, pero no consideramos que esto tenga un impacto significativo en la estimación de este mes. Al observar los porcentajes de error obtenidos con nuestro modelo concluimos que en este ejemplo nuestro método tiene una mayor precisión en el pronóstico.

Por último es de señalar que esta metodología para construir modelos lineales AR para FTD es totalmente automatizable. Además, al parecer el fenómeno de RP es bastante frecuente en el modelado de FTD por lo cual suponemos que los modelos lineales construidos con nuestra metodología pueden convertirse en una excelente alternativa para construir mejores modelos y en consecuencia pudieran tener un amplio rango de aplicaciones en el área de control industrial.

Capítulo 6

Conclusiones y trabajo futuro

6.1 Conclusiones

Las principales aportaciones de la tesis son:

- El desarrollo de una versión original de Algoritmos Genéticos Autoadaptables que permite encontrar buenas soluciones para problemas de optimización reales, no lineales, con variables acotadas y sin restricciones.
- El desarrollo de dos algoritmos heurísticos para encontrar modelos Autorregresivos (AR) de ST que son la base para construir modelos Autorregresivos y de Promedios Móviles (ARMA por sus siglas en inglés *Autorregresive and Movable Average*) para las mismas.
- Dado que las técnicas clásicas para medir bondad de ajuste de los modelos lineales no se aplican a los modelos aquí desarrollados, se construyó una prueba alternativa basada en prueba de hipótesis para comparación de medias y de varianzas.
- El descubrimiento y caracterización de un fenómeno que no aparece reportado en la literatura al que se ha denominado *retardo en el pronostico* (RP). Dicho fenómeno permite construir modelos lineales con mayor capacidad de pronóstico comparados con los modelos lineales tradicionales.
- Aplicar los algoritmos desarrollados para ST en el caso de FTD con el fin de encontrar buenos modelos lineales para éstas.

A partir de lo anterior podemos obtener varias conclusiones. La primera es que la metodología aquí desarrollada basada en plantear la construcción de modelos lineales como un problema de optimización, donde la construcción del problema esta orientada por la teoría clásica de ST, es correcta pues permite construir mejores modelos distintos de los obtenidos por los métodos tradicionales.

Otra conclusión es que el hecho de elegir los AGA como alternativa para resolver los problemas aquí planteados es muy importante pues permite explorar el espacio de soluciones de nuestro problema y encontrar las variables más significativas para la solución del mismo. Además, la versión de AGA desarrollada ha demostrado ser muy robusta al resolver muchos problemas distintos sin necesidad de ajuste de parámetros.

Como un resultado que no se tenía contemplado se descubrió y caracterizó el fenómeno de RP, el cual nos permitió la construcción de nuevos modelos lineales para ST y FTD, que en algunos casos son mejores alternativas, comparados con otros modelos lineales y no lineales. Además, estos nuevos modelos tienen un gran potencial de aplicación en áreas como control industrial, economía, finanzas, etc. En lo particular, yo pienso que el RP es una característica que tiene el fenómeno de que se trate, pero que solo es detectada si el modelo se construye con una metodología adecuada, sobre todo en la selección de variables y el establecimiento de los límites de los mismos.

Por último, es de resaltar que el hecho de tener una metodología totalmente automatizada y con posibilidad de modelar fenómenos que otras metodologías no pueden, abre un mundo de posibilidades sobre todo en el desarrollo de sistemas de cómputo para modelación y control de procesos.

6.2 Trabajo Futuro

El trabajo aquí desarrollado deja algunos puntos que a futuro es necesario afinar. A continuación presentamos una relación de los más importantes:

- Un primer punto sería el de buscar nuevos intervalos para los parámetros autoadaptables y revisar los criterios de paro de las etapas en las que se aplican los SAGA con el fin de que la resolución de los problemas se realice en menos tiempo.
- Un segundo punto sería el realizar pruebas substituyendo la metodología AGA de optimización por alguna otra (Recocido Simulado, Búsqueda Tabú, GRASP, etc.) con el fin de evaluar si es posible mejorar las soluciones obtenidas en calidad y tiempo de cómputo.
- Otro punto sería realizar más pruebas en otros ejemplos de modelación de FTD para determinar si es necesario agregar componentes del tipo MA y de ser así completar el desarrollo de la metodología en ese aspecto.
- Por otra parte hace falta investigar más sobre la elección de un modelo entre un conjunto de posibles soluciones para FTD que tenga la mejor capacidad de pronóstico.
- Por último, respecto al fenómeno de RP es necesario realizar una caracterización más completa para ST y desarrollar también una caracterización para FTD que actualmente no existe.

Bibliografía

- [1] Agapie, A., Agapie, A. "Forecasting the Economic Cycles Based on an Extension of the Holt-Winters Model. A Genetic Algorithm Approach" *Proceedings of the IEEE Computational Intelligence for Financial Forecasting*. Pp. 96-99. 1997.
- [2] Bäck Thomas. "The interaction of mutation rate, selection, and self-adaptation within genetic algorithm". *Proc. 2nd Conf. on Parallel Problem Solving from Nature* (Brussels,1992) Elsevier Amsterdam. 1992.
- [3] Bäck Thomas. "Self-adaptation in genetic algorithms". *Proc. 1st Eur. Conf. on Artificial Life*. MIT Press. Cambridge, MA. 1992.
- [4] Box George E., Jenkins Gwilym M. *Time Series Analysis: Forecasting and Control*. Holden-Day, INC. Oakland, Calif. USA 1976.
- [5] Carvalho Alexandre X., Tanner Martin A. "Mixtures-of-Experts of Autoregressive Time Series: Asymptotic Normality and Model Specification". *IEEE Transactions on Neural Networks*. Vol. 16, No. 1. January 2005. Pp. 39-56.
- [6] Chai, C., Chuek, C., M. DP. Huat, T. "Time Series Modeling and Forecasting using Genetic Algorithms" *Proceedings of the First International Conference on Knowledge-Based Intelligent Electronic Systems*. Vol 1. Pp. 260-268. 1995.
- [7] Chatfield Chris. *The Series Forecasting*. CRC Press. USA. 2000.
- [8] Cortez Paulo, Rocha Miguel, Neves Jose. "Evolving Time Series Forecasting ARMA Models". *Journal of Heuristics*. Vol. 10. No. 4. Pp.415-429. 2004.
- [9] Cortez, P., Rocha, M., Neves, J.: "A Meta-Genetic Algorithm for Time Series Forecasting." *Proceedings of the Workshop on Artificial Intelligence Techniques for Financial Time Series Analysis*, 10th Portuguese Conference on Artificial Intelligence, Pp. 21-31. 2001.
- [10] Cortez, Paulo, Rocha, Miguel, Neves, Jose.: "Genetic and Evolutionary Algorithms for Time Series Forecasting". *Proceedings of the 14th International conference on Industrial and engineering applications of artificial intelligence and expert systems: engineering of intelligent systems*. Pp.393-402. 2001.
- [11] Cota O. Maria de G, Flores Pérez Pedro. "El Algoritmo 2G". Manuscrito. Depto. De Matemáticas. Universidad de Sonora.

- [12] Eiben Ágoston E. Hinterding Robert, Michalewicz Zbigniew. Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*. Vol. 3 No. 2. Pp. 124-141. 1999.
- [13] Gaetan Carlo. "Subset ARMA Model Identification Using Genetic Algorithm". *Journal of Time Series*. Vol. 21. Issue 5. Pp.559-570. 2000.
- [14] Garduño Ramon, Morales Luis B., Flores P. Pedro. "Dinámica de Procesos Biológicos no Covalentes a Nivel Molecular". *Revista Mexicana de Física*. Vol 46. Suplemento 2. Pp. 135-141. 2000.
- [15] Garduño Ramon, Morales Luis B., Flores P. Pedro. "About Singularities at the Global Minimum of Empiric Force Fields for Peptides". *Journal at Molecular Structure (Theochem)*. 543 . 277-284. 2001.
- [16] Goldberg D. E. *Genetic Algorithms in Search, optimization and Machine Learning*. Addison Wesley, Reading, 1989.
- [17] Guerrero Víctor M. Guzmán. *Análisis Estadístico de Series de Tiempo Económicas*. Thomson Editores. México DF. 2003.
- [18] Hansen James V. McDonald James B., Nelson Ray D. "Time Series Prediction with Genetic Algorithms designed neural networks: An empirical comparison with modern statistical models". *Computational Intelligence*, Vol. 15. No. 3. Pp.171-184. 1999.
- [19] Holland, John H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor. 1975.
- [20] Huang, C., H. Yang. "A Time Series Approach To Short Term Load Forecasting Through Evolutionary Programming Structures." *Proceedings of the EMPD'95 International Conference*. Pp. 583-588. 1995.
- [21] Huber Peter J. *Robust Statistics*. Wiley-IEEE. USA. 2004.
- [22] Hyndman Rob, Koehler Anne. "Another look at measures of forecast accuracy". *International Journal of Forecasting*, Vol. 22, Issue 4. Pp. 679-688. 2006.
- [23] Hyndman, R (2003). Time Series Data Library. Disponible en: <http://www.robjhyndman.com/TSDL/>, y consultada el 21 de Octubre del 2008.
- [24] Kozlowski, B. " Wavelet Enhanced Analytical and Evolutionary Approaches to Time Series Forecasting". *ICANNGA (1), Lecture Notes in Computer Science*, Vol. 4431. Springer. Pp 49-57. 2007.
- [25] Li-Wei Lee, Li-Hui Wang, Shyi-Ming Chen, Yung-Ho Leu. "Handling Forecasting Problems Based on Two-Factors High-Order Fuzzy Time Series". *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, Vol. 14. No. 3. Pp. 468-477. 2006.

- [26] Makridakis Spyros, Wheelwright Steven C. *Manual de Técnicas de Pronosticos*. Ed. Limusa. Mexico, DF. 1997.
- [27] Makridakis Spyros, Wheelwright Steven C, Hyndman Rob J. *Forecasting: Methods and Applications*. John Wiley & Sons. New York. USA. Third edition 1998.
- [28] Makridakis Spyros, Hibon Michél. “The M3-Competition; results, conclusions and implications”. *International Journal of Forecasting*. Vol.16. Pp. 451-476. 2000.
- [29] McDonald James, Newey, W. K. “Partially adaptive estimation of regression models via the generalized t distribution”. *Econometric Theory*. Vol. 4. Pp. 428-457. 1988.
- [30] McDonald James, Yexiao Xu. “Some forecasting applications of partially adaptive estimators of ARIMA models”. *Economics Letters*, Vol. 45, Issue 4. Pp. 155-160. 1994.
- [31] Medeiros C. Marcelo, Veiga Álvaro. “A Hybrid Linear-Model for Time Series Forecasting”. *IEEE Transactions on Neural Networks*, Vol. 11, No. 6. Pp.1402-1412. 2000.
- [32] Medeiros C. Marcelo, Veiga Álvaro. “A Flexible Coefficient Smooth Transition Time Series Model”. *IEEE Transactions on Neural Networks*, Vol. 16, No. 1. Pp. 97-113. 2005.
- [33] Michalewicz Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer. Berlin. 1994.
- [34] Miller Brad L., Goldberg David E. “Genetic Algorithms, Tournament Selection, and the Effects of Noise”. *Complex Systems*. Vol. 9. Pp. 193-212.1995.
- [35] Montgomery Douglas C. *Diseño y Análisis de Experimentos*. LIMUSA WILEY, México DF. 2006.
- [36] Página oficial del Journal of Time Series Analysis. Consultada el 01/01/2007 (disponible en: <http://www.blackwellpublishing.com/journal.asp?ref=0143-9782>).
- [37] Palit Ajoy K. Popovic Dobrivoje. *Computational Intelligence in Time Series Forecasting: Theory and engineering applications*. Springe-Verlag London. 2005.
- [38] Quinlan J., *Induction of Decision Trees*, Kluwer Academic Publishers, Machine Learning, Hingham, MA, USA, Volume 1, Issue 1. Pp. 81-106. 1986.
- [39] Rolf, S., Sprave, J., and Urfer, W. "Model Identification and Parameter Estimation of ARMA Models by Means of Evolutionary Algorithms." *Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering (CIFEr'97)*. Pp. 237-243. 1997.
- [40] Sanchez-Corrales V.M. Flores-Perez Pedro, J.A. Valera. Perez-Tello M. “Mass Balance Calculations in Copper Flash Smelting by Means of Genetic Algorithms”. *Journal of Metals*. Vol. 56, No.12, Pp. 29-32, 2004.

- [41] S.M. Chen, J.R. Hwang. "Temperature prediction using fuzzy time series". *IEEE Trans. Syst. Man., Cybern. B, Cybern.* Vol. 30, No. 2. Pp. 263-275. 2000.
- [42] Swee Chiang Chiam, Kay Chen Tan and Abdullah Al Mamun. "Multiobjective, Evolutionary Neural Networks for Time Series Forecasting". *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO. Lecture Notes in Computer Science.* Vol. 4403. Springer. Pp. 346-360. 2007.
- [43] Szpiro George G. "Forecasting chaotic time series with genetic algorithms". *Physical Review E.* Vol. 5. No. 3. Pp. 2557-2568. 1997.
- [44] Weigend Andreas S, Gershenfeld Neil A. Editores de: *Time Series Prediction: Forecasting the future and Understanding the Past.* Addison-Wesley Publishing Company. Reading MA. USA. 1994.
- [45] Yadavalli Vansi K., Dahule Rahul K., Tambee Sanjeev S., Kulkarni B. D. "Obtaining functional form for chaotic time series using genetic algorithm". *CHAOS* Vol. 9 No. 3. Pp. 789-794. 1999.
- [46] La información aparece en: <http://www.cis.hut.fi/~lendasse/competition/competition.html>. consultada el 23 de agosto del 2009.
- [47] La información aparece en: <http://www.neural-forecasting-competition.com/NN5/index.htm> consultada el 23 de agosto del 2009.
- [48] La información aparece en: <http://www.neural-forecasting-competition.com/> consultada el 23 de agosto del 2009.
- [49] La información aparece en <http://www.neural-forecastingcompetition.com/NN3/results.htm>, consultada el 26 de Noviembre del 2007.

ANEXO 1

Algoritmo 2G

En esta sección se presenta el algoritmo que hemos denominado 2G, el cual es un algoritmo de área de representación y extracción de conocimientos basado en reglas y que presenta aportaciones originales al aplicar técnicas de reconocimiento supervisado de patrones, tanto en la etapa de preprocesamiento de la información, como en las de construcción de árboles de decisión y generación y evaluación de reglas.

Con dicho algoritmo se han realizado pruebas en varios ejemplos reportados en la literatura y se han obtenido mejores resultados. El algoritmo 2G produce nuevos conocimientos que se representan a través de reglas con una estructura distinta a la que se produce utilizando árboles tradicionales, pues utilizan menos atributos y solo se requiere de dos condiciones para evaluarse.

En este anexo la numeración de las referencias bibliográficas y de tablas es exclusiva del mismo.

A Introducción.

El algoritmo 2G esta basado en árboles de decisión [5] [15] [16] [28], y resulta muy eficiente en el tratamiento de problemas con atributos que utilizan valores reales, aunque se puede utilizar con problemas de variables discretas o con mezcla de reales y discretas. Las diferencias principales con otros algoritmos de este tipo radican en que genera árboles de poca profundidad, e implementa procedimientos para generar reglas que, al aplicarse en el conjunto de entrenamiento, están exentas de error. Por otra parte, para su evaluación se requieren a lo más de dos condiciones por regla, y aplica un mecanismo que permite disminuir el número de reglas. Las reglas que aquí aparecen permiten construir lo que hemos denominado “patrones virtuales no explícitos”, que consisten en que un ejemplo puede ser clasificado, a pesar de que no exista una regla que se aplique en el, utilizando partes de reglas distintas que clasifican en una misma clase.

Para evaluar el desempeño de este algoritmo, se eligieron varios problemas que son clásicos en el área, y se realizaron pruebas comparativas con el código libre que se proporciona para aplicar los algoritmos C4.5 [24], See5 [18] y J48 [19]. Además, se compararon los resultados obtenidos por el algoritmo 2G con los de otras metodologías que aparecen en varias publicaciones [2] [6] [12] [15] [16] [17] [28], y en sitios oficiales de internet [21] [22] [23] [25] [26]. Es de resaltar que en todas las comparaciones realizadas se obtuvieron mejores resultados con 2G que los existentes a la fecha. Por cuestiones de espacio, estos resultados no se presentan en esta sección.

Para efectos de organización, este anexo se divide en tres secciones: la primera corresponde a esta introducción; la segunda describe el algoritmo 2G; y por ultimo, se presentan las conclusiones del tema.

B Algoritmo 2G.

La construcción del algoritmo 2G se basa, al igual que en los algoritmos C4.5 y See5, en los conceptos de entropía y ganancia de información, pero estos se utilizan en una forma distinta, la cual se describe a continuación.

B.1 Entropía y Ganancia de Información.

Para estimar los grados de Entropía y ganancia de información, es necesario contar con un conjunto de valores por atributo que son agrupados por una medida que los identifica con una clase. En este caso, se seleccionan de los valores continuos, los máximos de cada grupo que pertenece a una misma clase, y se procede a aplicar el siguiente modelo que se construye en base a los siguientes conceptos:

- Dominio.- Nombre que permite identificar a un conjunto de valores con propiedades homogéneas [47].
- Universo del discurso.- Entorno donde se define el problema representado por el producto cartesiano de un conjunto finito de dominios [47].
- Atributo o variable del problema.- Objeto con características propias que puede tomar valores existentes en el universo del discurso asociados a un determinado dominio [47].
- Ejemplo de entrenamiento.- Representado por una fila del registro correspondiente al conjunto de los atributos.
- Clase.- Define la generalización de un objeto en particular, y representa un patrón o un prototipo de una familia de objetos concretos. Una instancia de la clase para este caso se representa a través de un ejemplo de entrenamiento [45].

Un conjunto de patrones se considera un grupo si tienen características homogéneas y diferenciadas de los demás en el espacio muestral de representación. En este trabajo, para la selección de atributos que permitan agrupar instancias de la misma clase, se utiliza el concepto de entropía o medida de la incertidumbre que existe en un conjunto de valores [1]. Para esto se aplican las especificaciones que se describen a continuación:

- S conjunto de ejemplos de entrenamiento original.
- Sr conjunto reducido de ejemplos de entrenamiento.
- p número total de ejemplos de S o de Sr .
- e instancia de p .
- c clase a la que corresponde e .
- k número total de clases.
- z índice del valor de la clase c asociado con e y $z = \{1, \dots, c\}$.
- n número total de ejemplos de una clase de e_1
- l número de atributos del conjunto S o de Sr .
- A conjunto de atributos.
- Ar conjunto de atributos seleccionados en $G(Ar)$.
- t es el número total de ejemplos de A o de Ar .
- v conjunto de posibles valores que puede tomar A en el universo del discurso de Ar .

- m número de ejemplos de v .
- T número de valores de v en A o de v en Ar .
- $prob_i$ probabilidad de que un ejemplo e_i de Ar pertenezca a una clase c con $i = \{1, \dots, t\}$.

Además, se definen las funciones: $H(Sr)$ y $H(Ar)$ para la entropía conjunta, $H(v)$ la entropía por atributo y $G(Ar)$ para la ganancia de información de Ar , por medio de las siguientes expresiones:

$$H(Sr) = \sum_{i=1}^p - prob_i * \log_2(prob_i)$$

$$H(Ar) = \sum_{r=1}^T H(Sr) - prob_r * H(v)$$

$$H(v) = \sum_{j=1}^l - prob_j * \log_2(prob_j)$$

$$G(Ar) = \sum_{x=1}^l H(S) - H(Ar)$$

Bajo este esquema, se toma como entrada un conjunto de ejemplos de entrenamiento que son procesados para crear un árbol de decisión, y se generan reglas que se utilizan para validar nuevas instancias del problema que se trate. Este es una metodología clásica que ya aparece en la literatura [9] [24].

B.2 Discretización de valores.

Para discretizar los valores de los atributos, se toma como base el método implementado en los algoritmos C4.5, See5 y J48, pero se realizan operaciones distintas orientadas a obtener mayor precisión en los resultados, mismas que se describen a continuación:

- a) Tradicionalmente, del conjunto de entrenamiento, se toman los valores de un atributo y la clase a que corresponden, y se ordenan de menor a mayor. En nuestro caso, se realiza el mismo procedimiento pero se evitan redundancias.
- b) Para representar la lista de valores y la existencia de instancias por clase, que denominamos “ninst”, se utiliza una matriz donde se asigna un 0 a la falta de instancias, y un 1 en caso contrario.
- c) Para obtener el grado de entropía y ganancia en la información, se eligen los valores máximos de los grupos donde “ninst” toma un valor mayor que 0 en las mismas clases, y el grupo resultante no se reduce.

Como ejemplo, a continuación se muestran en el Vector 1, valores registrados para un atributo ‘X’, que pertenecen a las clases ‘-1’ o ‘1’, que con el método tradicional se ordenaría de la siguiente forma:

Vector 1.- Valores de atributo 'X' ordenados con método tradicional.

Valor	65	70	70	80	85	86	86	90	99	100
Clase	-1	1	-1	-1	1	1	1	-1	-1	1
	x			x			x		x	x

Donde 'x' representa en el Vector 1 y en el Vector 2 a los valores que son seleccionados para participar en el proceso de aplicación de alguna heurística de selección de atributos, como puede ser utilizar el criterio de la media, la mediana, entropía, etc.

Por otra parte, con el fin de alcanzar mayor precisión en la generación de reglas, en nuestro caso se aplica un procedimiento para evitar redundancia en los datos, que consiste en construir una matriz donde las columnas se relacionan con los valores, y las filas con las clases correspondientes.

En el ejemplo que se analiza, tomando como base los valores del Vector 1, se construye un Vector 2, asignándole un '1' a la casilla donde existen instancias del valor para alguna de las clases, y en caso contrario se asigna un '0'.

Después, se seleccionan los valores que marcan lo que denominamos "patrón de rompimiento de clases", que consiste en revisar en forma ascendente las casillas que corresponden a los valores del vector que conectan con la identificación de una clase, y se van seleccionando los valores donde esta casilla presenta un cambio entre '1' y '0', y se incluyen además aquellos para los cuales la clase no está bien definida, es decir, el valor tiene participación en más de una clase.

Vector 2.- Valores de atributo 'X' ordenados con nuestro método.

Valor	65	70	80	85	86	90	99	100
Clase -1	1	1	1	0	0	1	1	0
Clase 1	0	1	0	1	1	0	0	1
	x	x	x		x		x	x

Destacando en este procedimiento, que la principal diferencia radica en incluir los valores que están relacionados con más de una clase, agrupándolos para ser representados como si se tratara de un conjunto de valores de clases distintas a las existentes, a lo que hemos denominado "clase virtual".

B.3 Selección de atributos.

A diferencia de algoritmos como el C4.5, See5 y J48, en nuestro caso se obtiene el mayor grado de información de los atributos en una sola ocasión, y del orden obtenido, para provocar mayor variabilidad en los datos se intercambia el primer atributo por el segundo, y se continúa con la creación del árbol de decisión correspondiente.

Cabe mencionar que al utilizar métodos distintos como el descrito en el punto anterior, se obtiene un orden muy homogéneo de la selección de atributos que reportan algoritmos como el C4.5, See5 y J48, y se alcanza una mayor precisión en los resultados obtenidos en el proceso de clasificación con el algoritmo 2G.

Para efectos de ejemplificar, en la tabla A.I se presenta una comparación de resultados sobre varios problemas reportados en la literatura [1] [6] [12] [15] [16] [21] [22] [23] [25] [26] [28]:

Tabla A.I
Comparación de resultados.

Nombre del problema	dsi	dsa	dc	Porcentaje de error (%)				
				AdaBoost	See5	C4.5	J48	2G
Cancer de seno de Wisconsin	699	10	2	2.96	3.01	4.87	5.44	0.14
Diabetes de los indios Pimas	768	8	2	24.75	26.03	29.29	26.17	0.26
Problema del Sonar	208	60	2	21.35	14.53	28.97	28.84	0
Problema del Vino	168	13	3	3.14	2.25	8.83	6.18	2.25
Problema de la Ionosfera	351	34	2	6.61	6.23	7.96	8.55	1.14

dsi = Número of instancias
dsa = Número de atributos
dc = Número de clases

Los mejores resultados aparecen resaltados en color gris en la Tabla A.I.

B.4 Resumen del algoritmo 2G.

Para aplicar el algoritmo 2G es necesario seguir los siguientes pasos:

- a) Registro de información:
 - Clases (c).
 - Atributos (Ar).
 - Valores de atributos (v).
 - Ejemplos de entrenamiento (e).
- b) Calculo de los valores:
 - p, k, n, l, t, m, T .
 - Entropía Conjunta representada por $H(Sr)$.
 - Por cada atributo, se aplica $H(Ar)$.
 - Por último, por única vez, se aplica $G(Ar)$ y se seleccionan en Ar los cuatro atributos que presentan mayor grado en ganancia en información.
- c) Construcción del árbol con Ar :
 - Se toman los mayores valores representativos del atributo, que cumplen con la siguiente regla:
“De cada posible combinación de clases, donde existe coincidencia en que el número de elementos es mayor que 0, se toma el valor máximo como representativo del grupo”.
 - Se restringe que el nivel del árbol no exceda de cuatro niveles.
 - Se realiza una permutación entre el primer y segundo nodo seleccionados.

- e) Generación de reglas:
- Se generan reglas exactas ordenadas por un índice ascendente, que se forman únicamente por dos condiciones resultantes de las posibles combinaciones de pares de atributos contiguos que presentan grado de error = 0.
 - Para cada regla se fija un valor mínimo y uno máximo.
 - En lugar de los métodos tradicionales de ‘poda’, en el caso de valores continuos, se realiza una reducción de reglas basada en la pertenencia de valores a un rango de valores continuos, donde se agrupan elementos de una misma clase.
- f) Evaluación del conjunto de prueba:
- Para aumentar el grado de exactitud en la clasificación del conjunto de prueba, se permite que un ejemplo sea clasificado por distintas reglas de una misma clase, con las siguientes restricciones:
- Se asigna la primera regla a la condicional de un ejemplo donde los rangos de los valores coinciden, tomando en cuenta el orden ascendente del índice que le corresponda a la misma.
 - Un ejemplo puede ser clasificado por una regla.
 - Un ejemplo puede ser clasificado por dos reglas que pertenezcan a una misma clase, siempre y cuando los atributos que clasifican sean distintos. A esto último lo hemos denominado “patrones virtuales no explícitos” (ver Figura B.1).
 - Para la evaluación de nuevos ejemplos se toma como referencia la estructura del árbol, utilizando en primera instancia las reglas generadas por pares de nodos contiguos, y después, en caso de no haber clasificado la totalidad de los mismos, se utilizan pares de atributos intercalados en los distintos niveles, y como último paso, se combina el primer nodo con el último, o se aplican reglas simples sobre alguno de los atributos que forman parte del árbol de decisión que ha sido generado previamente.

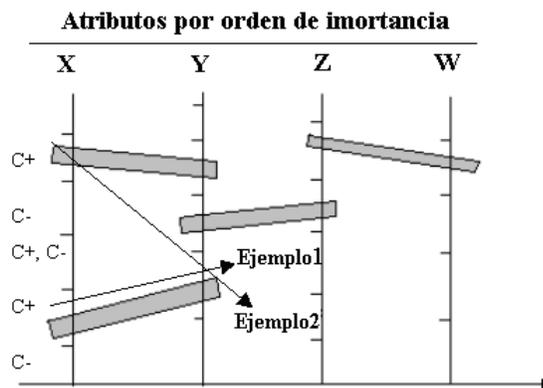


Fig. B.1.- Patrones virtuales no explícitos.

C Conclusiones

La selección de los valores de atributos es muy importante por el impacto que presenta en los resultados, por tal razón, en lugar de buscar disminuir el número de valores representativos en el proceso de discretización de los datos, se ha implementado un procedimiento que al agrupar datos en lo que hemos denominado “clase virtual”, permite evaluar lo que hemos llamado “patrón de rompimiento de clases”.

En cuanto a la selección de atributos, hemos utilizado el método tradicional de agruparlos por orden de ganancia de información y los resultados son muy parecidos a los que reportan algoritmos como C4.5, See5 y J48 de Weka, con la diferencia de que al intercambiar el primer atributo por el segundo, se busca que aparezca un “ruido” que cause mayor variabilidad en los datos ya que, como hemos mencionado, a mayor variabilidad de los datos se obtendrá mayor precisión. Por otro lado, los árboles de decisión no exceden un nivel de profundidad de cuatro niveles, lo que los hace más eficientes en su tratamiento.

En el proceso de generación de reglas, la principal diferencia consiste en que éstas se conforman sólo de una o dos condiciones, y para su reducción, se agrupan por valores continuos que pertenecen a una misma clase, lo que disminuye el tiempo para evaluar una nueva instancia y conservar el nivel de eficiencia en la asignación de clase.

Para la evaluación de nuevas instancias sin perder el buen grado de clasificación, el permitir que un ejemplo sea clasificado por atributos y reglas distintas, con la salvedad de que sean de la misma clase, amplía el esquema de clasificación. A esto lo hemos denominado “patrón virtual no explícito”.

Por último, cabe mencionar que hemos comprobado que el hecho de ampliar los esquemas de tratamiento de datos, en lugar de restringirlos, y a la vez implementar criterios que permiten disminuir el tiempo de cómputo, tanto en el proceso de clasificación como en el de evaluación, aumenta el grado de precisión en la asignación de nuevas instancias a una determinada clase.

Referencias Bibliográficas

- [1] Ahmad A., Dey L, “A feature selection technique for classificatory analysis”, *Journal Title: Pattern Recognition Letters*, Volume 26, Issue 1, Pp. 43-56. 2005.
- [2] *Conference*. University of California, Berkeley; UC Berkeley. 2006.
- [3] Beltrán N., Duarte M., Salah S., Bustos M., Peña N., Loyola M. & Jalocha, J. W., N, “Feature selection algorithms using Chilean wine chromatograms as examples”, *Journal of Food Engineering* 67. Elsevier SCI. (4): Pp. 483-490. 2005.
- [4] Brachman R.J. & Anand T., *The Process Of Knowledge Discovery In Databases: A Human-Centered Approach*. In *Advances In Knowledge Discovery And Data Mining*, eds. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, AAAI Press/The MIT Press, Menlo Park, CA., Pp. 37-57. 1996.
- [5] Comley W., Allison L. & Fitzgibbon J., “Flexible Decision Trees in a General Data-Mining Environment”, *Fourth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL-2003)*, Hong Kong. 2003.
- [6] Díaz M., Fernández M.z & Martínez A., “See5 Algorithm versus Discriminant Analysis. An Application to the Prediction of Insolvency in Spanish Non-life Insurance Companies”, *Documentos de trabajo de la Facultad de Ciencias Económicas y Empresariales*. Universidad Complutense de Madrid. Pp. 04-12. 2004.

- [7] Felici G., Vercellis C., *Mathematical Methods for Knowledge Discovery and Data Mining, Information Science Reference*, Hershey • New York, Pp. 1-5, 244-255. 2008.
- [8] Joyanes, L., *Programación orientada a objetos*, Mc Graw Hill, Pp. xvii-xix. 1999.
- [9] Kantardzic, M., *Data Mining: Concepts, Models, Methods, and Algorithms*, Wiley-Interscience, Pp. 1-56. 2003.
- [10] Kim, Y., Street, W., and Menczer, F., “Feature selection in Data Mining, Data Mining: Opportunities and Challenges”, *Idea Group Publishing, Hershey, PA*, Pp. 80-105. 2002.
- [11] Kira, K. & Rendell, L. A., “The feature selection problem: Traditional methods and a new algorithm”. *10th National Conference on Artificial Intelligence*. Cambridge, MA, USA. 1992.
- [12] Kohavi, R., Li, C.-H., “Oblivious decision trees, graphs, and top-down pruning”, *Fourteenth International Joint Conference on Artificial Intelligence*. San Francisco, CA. 2005.
- [13] Miguel, A., Piattini, M., Marcos, E., *Diseño de Bases de datos Relacionales*, Alfa-Omega – ra-ma, Pp. 39, 95. 2000.
- [14] Peña Daniel, *Análisis de Datos Multivariantes*, Mc Graw Hill, Madrid, España. Pp. 61-65. 2002.
- [15] Quinlan J., “Induction of Decision Trees”, *Kluwer Academic Publishers, Machine Learning*, Hingham, MA, USA, Volume 1, Issue 1. Pp. 81-106. 1986.
- [16] Quinlan R & Cameron J., “Oversearching and Layered Search in Empirical Learning”, *International Joint Conference on Artificial Intelligence*. Montreal, Quebec, Canada. 1995.
- [17] Quinlan R., *Comparing Connectionist and Symbolic Learning Methods*. Cambridge, Massachussetts. pp. 445-456. 1994.
- [18] URL: <http://www.rulequest.com/see5-info.html>.
- [19] URL: <http://www.cs.waikato.ac.nz/ml/weka/>.
- [20] URL: <http://www.grappa.univ-ille3.fr/~torre/guide.php?id=accueil>.
- [21] URL: <http://kdd.ics.uci.edu/>.
- [22] URL: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
- [23] URL: <http://www.ailab.si/orange/doc/datasets/breast-cancer-wisconsin.htm>.
- [24] URL: <http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html>.
- [25] URL: <http://archive.ics.uci.edu/ml/datasets.html>.
- [26] URL: <http://www.rulequest.com/Personal/>.

- [27] Wang L, Fu X., *Data Mining with Computational Intelligence*, Springer Berlin Heidelberg, New York. Pp. 1-2, 117-137. 2005.
- [28] Xian G., Zhu Z., Hoppus M. & Fleming M., “Application of Decision-Tree Techniques to Forest Group and Basal Area Mapping Using Satellite Imagery and Forest Inventory Data”, *Integrated Remote Sensing at the Global, Regional and Local Scale, ISPRS Commission I Mid-Term Symposium in conjunction with Pecora 15/Land Satellite Information IV Conference*. Denver, CO USA. 2002.
- [29] Yang Y & Wu X, “Parameter tuning for induction-algorithm-oriented feature elimination”, *Intelligent Systems*, IEEE, Volume 19, Issue 2. Pp. 40-69. 2004.
- [30] Zhang, P., Brijesh V., Kuldeep K., “Neural vs. statistical classifier in conjunction with genetic algorithm feature selection in digital mammography”, *Sch. of Inf. Technol., Evolutionary Computation, CEC '03*, Bond Univ., Gold Coast, Qld., Australia. 2003.

ANEXO 2

Descripción de Algoritmos

Dado que la extensión del código de los algoritmos que aquí se describen es muy grande, solamente se presentan una descripción simbólica de los aspectos más importantes que proporcionan detalles de comprensión de la metodología aplicada que ha sido descrita en este documento. Esta descripción se hace tomando en cuenta los siguientes puntos:

- Solo se describirá código para las funciones llamadas procedimientos.
- Los procedimientos pueden utilizar funciones que se presentan sin escribir su código.
- Las funciones y procedimientos pueden devolver vectores.
- Puede haber igualdad entre vectores a ambos lados del símbolo =.
- En todo momento se sobreentiende el tamaño de los vectores.
- Se usara una notación parecida a la del lenguaje C.

A continuación pasamos a describir los algoritmos.

A Algoritmos referenciados en el capítulo 3.

Descripción de variables globales:

K	Número de términos del modelo AR.
periodo	Número de retrasos a considerar del modelo autorregresivo.
importantes[K]	Vector con los índices de las K variables importantes del modelo.
sol[periodo]	Vector para guardar soluciones.
mejor_solucion[periodo]	Vector para guardar el mejor valor de las soluciones.
Z_0	Serie original.
Z_1	Serie de primeras diferencias.
Z_2	Serie de segundas diferencias.
Z_3	Serie de terceras diferencias.
Z_T	Serie de trabajo.

Descripción de funciones:

`var()` : Es la función varianza.

`min()` : Es la función mínimo. Se introducen los términos a comparar como parámetros y se devuelve el lugar del valor mínimo.

`Autocorr(,)` : Es la función de autocorrelación. En la primera componente va el valor de la serie de que se trate y en la segunda el desplazamiento temporal.

`máximo_local(R, 30)` : Devuelve la coordenada del primer máximo local de los 30 valores de R (distintos del primero). En caso de que este no exista devuelve 4.

`fijar_limites1(i)` : Se fijan los límites para aplicar los algoritmos genéticos. Si $i=1$ se aplica en la primera etapa de acuerdo a (12). Si $i=2$ se aplica en la segunda etapa después de hacer cero los límites de las variables que no importan, para esto se usa la información de índices que está en el vector `importantes[K]`. Si $i=3$ se estiman los límites para la serie original después de aplicar el **resultado 1**. Si $i=4$ se hacen todos los límites $[-1,1]$. Si $i=5$ se hacen cero los límites de las variables que no importan y los demás son $[-1,1]$. Si $i=6$ se amplían los límites de la solución para aplicar la tercera etapa.

`aplica_geneticos1()` : Se aplican los algoritmos genéticos sobre Z_T de acuerdo a la descripción de la primera etapa. Devuelve un vector de tamaño periodo.

`aplica_geneticos2()` : Se aplican los algoritmos genéticos sobre Z_T de acuerdo a la descripción de la segunda etapa. Devuelve un vector de tamaño periodo.

`abs()` : Función valor absoluto.

`ordena(V)` : Función que se aplica a un vector V y devuelve otro del mismo tamaño pero con los índices que corresponden a las variables de mayor a menor.

`corta(V,K)` : Función que se aplica a un vector V y devuelve otro de tamaño K con las primeras componentes de V.

`evalua_solucion()` : Función que se aplica a una solución del problema y devuelve el valor de RSS.

Se elige con que serie se va a trabajar

```
procedimiento eleccion_serie_trabajo() {  
    int i, eleccion;  
    float v[4];
```

```

    for (i=0;i<4;i++) v[i]=var(Zi);
    eleccion=min(v[0], v[1], v[2], v[3]);
    ZT= Zeleccion;
}

```

Se calcula cuantas variables se van a considerar en el problema

```

procedimiento calcula_periodo() {
    int i;
    float R[30];
        /* R[k] guardará el valor de autocorrelación
        de la serie ZT con desplazamiento temporal k */

    for (i=0;i<30;i++) R[i]=Autocorr(ZT, k);
    periodo= maximo_local (R, 30)+1;    /* Es +1 porque se incluye el valor de δ */
}

```

Se realiza la primera etapa de genéticos. Se calculan 10 soluciones y se suman los valores absolutos de ellas.

```

procedimiento primera_etapa_geneticos() {
    float soluciones[10, periodo];
    int i,j ;

    // aqui se calculan y guardan 10 soluciones de la primera etapa
    for (i=0;i<10;i++) soluciones[i, periodo]=aplica_geneticos1( )
        /* aquí se suman los valores absolutos de estas 10 soluciones y se guardan en
        sol[periodo] */
        for (i=0;i<periodo;i++){
            for (j=0;j<10;j++) sol[i]= sol[i]+ abs(soluciones[j,i ]); }
}

```

Se calculan cuales son las K variables importantes del sistema.

```

procedimiento calcula_importantes() {
    int lugares[periodo];
        lugares[]=ordena(sol)
        importantes[]=corta(lugares,K);
}

```

Se realiza la segunda etapa de genéticos.

```
procedimiento segunda_etapa_geneticos( ) {  
float soluciones[5, periodo], valores_solucion[5];  
int i, j ;  
    // aqui se calculan y guardan 5 soluciones de la segunda etapa  
    for (i=0;i<5;i++) {  
        soluciones[i, periodo]=aplica_geneticos2( );  
        valores_solucion[i]=evalua_solucion(soluciones[i, periodo]); }  
        j=min(valores_solucion);  
        mejor_solucion=soluciones[j,periodo];  
    }  
}
```

Se calcula una solución con el algoritmo I,

```
procedimiento algoritmo_I( ) {  
    eleccion_serie_trabajo( );  
    calcula_periodo( );  
    fijar_limites(1);  
    primera_etapa_geneticos( );  
    calcula_importantes( );  
    fijar_limites(2);  
    segunda_etapa_geneticos( );  
    fijar_limites(3);  
     $Z_T = Z_0$ ;  
    mejor_solucion= aplica_geneticos2( );  
}
```

Se calcula una solución con el algoritmo II.

```
procedimiento algoritmo_II( ) {  
    eleccion_serie_trabajo( );  
    calcula_periodo( );  
    periodo=periodo+eleccion;  
     $Z_T = Z_0$ ;  
    fijar_limites(4);  
    primera_etapa_geneticos( );  
    calcula_importantes( );  
    fijar_limites(5);  
    segunda_etapa_geneticos( );  
    fijar_limites(6);  
}
```

```
mejor_solucion= aplica_geneticos2( );  
}
```

B Algoritmo referenciado en el capítulo 4.

En esta sección se describe el algoritmo para estimar el modelo ARMA de una serie. La decisión de aplicar el adelanto en el pronóstico no se incluye en este algoritmo.

Descripción de variables globales:

K número de variables del modelo.
 K_{AR} número de variables autorregresivas del modelo.
 K_{MA} número de variables de promedios móviles del modelo.
 Z_0 Serie original.
 Z_T Serie de trabajo.
 F_T Serie de pronóstico.

modelo[K] : Aquí se guardan las variables del modelo.

modeloAR[K_{AR}] : Vector que corresponde a la solución autorregresiva del modelo.

modeloMA[K_{MA}] : Vector que corresponde a la solución de promedios móviles del modelo.

calcula_modelo(K) : Función que calcula un modelo AR para la serie Z_T con K términos tomando en cuenta el conjunto de entrenamiento. El valor del modelo se guarda en el vector modelo[K].

guarda_solucionAR() : Aquí se guarda la solución de la parte AR en modeloAR[K_{AR}]

guarda_solucionMA() : Aquí se guarda la solución de la parte MA en modeloMA[K_{MA}]

calcula_pronosticoAR() : Se calculan los valores para la serie F_T tomando en cuenta el modelo modeloAR[K_{AR}].

calcula_pronosticoMA() : Se calculan los valores para la serie F_T tomando en cuenta los modelos modeloAR[K_{AR}] y modeloMA[K_{MA}].

evalua_modeloAR() : Función que evalúa el modelo modelo[K] para la serie Z_T y el pronóstico F_T tomando en cuenta el conjunto de prueba. La función devuelve el valor del modelo.

evalua_modeloMA() : función que evalúa el modelo modelo[K] para la serie Z_T y el pronóstico F_T tomando en cuenta el conjunto de prueba. La función devuelve el valor del modelo.

Aquí se calcula la parte autorregresiva del modelo.

```
procedimiento calcula_modeloAR() {  
    float valor; K=2, valor=infinito;  
    calcula_modelo( K);  
    calcula_pronosticoAR( );  
    while(valor>evalua_modeloAR()){  
        guarda_solucionAR( );  
        valor= evalua_modeloAR( ); K=K+1;  
        calcula_modelo(K); }  
}
```

Aquí se calcula la parte de promedios móviles del modelo.

```
procedimiento calcula_modeloMA() {  
    float valor; K=2, valor=infinito;  
    calcula_modelo(K);  
    calcula_pronosticoMA( );  
    while(valor>evalua_modeloMA()){  
        guarda_solucionMA( );  
        valor= evalua_modeloMA( ); K=K+1;  
        calcula_modelo(K); }  
}
```

Este procedimiento calcula el modelo ARMA.

```
procedimiento calcula_modelo_ARMA() {  
    calcula_modeloAR( );  $K_{AR} = K$ ;  
    modeloAR[  $K_{AR}$  ]= modelo[ K ];  
    calcula_pronosticoAR( );  
     $Z_T = F_T - Z_0$ ; // aquí se toma como serie de trabajo la del error  
    calcula_modeloMA( )  
     $K_{MA} = K$ ;  
    modeloMA[ $K_{MA}$  ]= modelo[ K ];  
}
```
