

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO - CHILE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**“TAKAGI SUGENO MODELO CON
ALGORITMO PSO PARA PRONÓSTICO
DE CAPTURAS DE ANCHOVETAS EN EL
NORTE DE CHILE”**

Roderich Frank Henríquez Inzulza

Diciembre, 2009.

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**“TAKAGI SUGENO MODELO CON
ALGORITMO PSO PARA PRONÓSTICO
DE CAPTURAS DE ANCHOVETAS EN EL
NORTE DE CHILE”**

Roderich Frank Henríquez Inzulza

Profesor Guía: **Nibaldo Rodríguez Agurto**

Profesor Co-referente: **Rodrigo Alfaro Arancibia**

Carrera: **Ingeniería de Ejecución en Informática**

Diciembre, 2009.

Dedicatoria

Dedicado al esfuerzo de mis padres y a mis hermanos,
que me apoyaron en todo momento.

Agradecimientos

A Dios, mis padres y a mi profesor guía.

Resumen

Resumen

En esta tesis se presenta la investigación y el desarrollo de un sistema de predicción basado en un modelo de inferencia difusa de tipo Takagi Sugeno con un algoritmo de optimización por enjambre de partículas (PSO) para el pronóstico de capturas de anchovetas en el norte de Chile, en la cual se considera el análisis de las capturas de los meses previos para entregar el pronóstico del mes siguiente. Los resultados de la investigación confirman que las tecnologías antes mencionadas pueden ser utilizadas como un buen estimador para la predicción de capturas de anchovetas.

Palabras claves: Optimización, pronóstico, difusa.

Abstract

This thesis presents the research and development of a prediction system based in a fuzzy forecast model of Takagi Sugeno type with a particle swarm optimization (PSO) algorithm for the catch forecasting of anchovy in the north area of Chile, witch considering the analysis of the previous months for to give the next month forecast. The results of the research confirm that the technologies before mentioned can be used as a good estimator for the catch forecasting of anchovy.

Key words: Optimization, forecasting, fuzzy.

Índice de Contenido

Capítulo 1: Introducción	1
1.1 Introducción.....	1
1.2 La Anchoqueta.....	2
1.3 Hábitat.....	2
1.4 Reproducción.....	3
1.5 Aporte nutritivo.....	3
1.6 Importancia del problema.....	3
1.7 Objetivos.....	4
Capítulo 2: Optimización por enjambre de partículas (PSO).....	5
2.1 Introducción.....	5
2.2 Organización del texto.....	6
2.3 El enjambre y su inteligencia de optimización.....	6
2.4 Modelo tradicional.....	7
2.5 Parámetros del PSO.....	9
2.6 Topologías local y global.....	11
2.7 Actualización síncrona y asíncrona.....	12
2.8 Exploración y explotación de PSO.....	12
2.9 PSO con restricción de velocidad mínima.....	13
2.10 PSO Quantum.....	14
Capítulo 3: Modelo difuso Takagi Sugeno.....	15
3.1 Introducción.....	15
3.2 Organización del texto.....	15
3.3 Takagi Sugeno.....	16
3.4 Datos de entrada del modelo.....	16
3.5 Reglas del modelo.....	16
3.6 Conjuntos difusos del modelo.....	17
3.7 Salida del modelo.....	18
Capítulo 4: Simulación y evaluación de rendimiento.....	19
4.1 Introducción.....	19
4.2 Organización del texto.....	19
4.3 Métricas de evaluación.....	20
4.4 Plan de pruebas.....	20
4.5 Pruebas con PSO Tradicional.....	21
4.5.1 Topología de 2 entradas.....	22
4.5.2 Topología de 4 entradas.....	28
4.5.3 Topología de 8 entradas.....	34

4.6	Análisis de resultados con PSO Tradicional.....	40
4.7	Pruebas con PSO Quantum.....	41
4.7.1	Topología de 2 entradas.....	42
4.7.2	Topología de 4 entradas.....	44
4.7.3	Topología de 8 entradas.....	46
4.8	Pruebas de las variables del sistema.....	48
4.8.1	Pruebas para la cantidad óptima de épocas.....	48
4.8.2	Pruebas para la cantidad óptima de partículas.....	50
4.8.3	Pruebas para el valor de la variable de peso inercial.....	51
4.8.4	Pruebas para el valor de la variable de influencia cognitiva C_1	51
4.8.5	Pruebas para el valor de la variable de influencia social C_2	51
4.8.6	Pruebas para el valor de la variable de velocidad máxima.....	52
Capítulo 5: Resultados.....		53
5.1	Introducción.....	53
5.2	Configuración óptima del sistema.....	53
5.2.1	Implementación Takagi Sugeno con PSO Tradicional.....	53
5.2.2	Implementación Takagi Sugeno con PSO Quantum.....	56
5.3	Análisis de resultados.....	58
Capítulo 6: Conclusiones.....		59
Referencias.....		61
Anexo A – Código Fuente.....		64
	Main Class.....	64
	Takagi.....	64
	Enjambre.....	69

Glosario

- **PSO** : Particle Swarm Optimization
- **QPSO** : Quantum Particle Swarm Optimization
- **ACO** : Ant Colony Optimization
- **GA** : Genetic algorithm
- **TS** : Takagi Sugeno
- **SID** : Sistema de inferencia difusa
- **ECM** : Error cuadrático medio
- **MAE** : Mean absolute error
- **SAG** : Servicio Agrícola Ganadero.
- **SERNAP** : Servicio Nacional de Pesca.
- **SERNAPESCA** : Servicio Nacional de Pesca.

Índice de Figuras

Figura 1.1	Imagen de un ejemplar de anchoveta.....	2
Figura 3.1	Función de membresía de tipo gaussiana.....	17
Figura 3.2	Diagrama de flujo del modelo TS en su etapa forward.....	18
Figura 4.1	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.2 obtenido en el Random 5, con un R^2 de 75,93%.	23
Figura 4.2	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 75,93%.	23
Figura 4.3	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.2 obtenido en el Random 10, con un R^2 de 75,95%.	23
Figura 4.4	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 10, con un R^2 de 75,95%.	23
Figura 4.5	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.3 obtenido en el Random 8, con un R^2 de 71,45%.	25
Figura 4.6	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 8, con un R^2 de 71,45%.	25
Figura 4.7	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.3 obtenido en el Random 3, con un R^2 de 81,29%.	25
Figura 4.8	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 81,29%.	25
Figura 4.9	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.4 obtenido en el Random 9, con un R^2 de 65,96%.	27
Figura 4.10	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 65,96%.	27
Figura 4.11	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.4 obtenido en el Random 1, con un R^2 de 84,26%.	27
Figura 4.12	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 1, con un R^2 de 84,26%.	27
Figura 4.13	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.5 obtenido en el Random 10, con R^2 de 68,74%.	29
Figura 4.14	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 10, con un R^2 de 68,74%.	29
Figura 4.15	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.5 obtenido en el Random 4, con un R^2 de 69,05%.	29
Figura 4.16	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 4, con un R^2 de 69,05%.	29
Figura 4.17	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.6 obtenido en el Random 3, con R^2 de 68,40%.	31

Figura 4.18	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 68,40%.	31
Figura 4.19	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.6 obtenido en el Random 7, con un R^2 de 70,78%.	31
Figura 4.20	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 7, con un R^2 de 70,78%.	31
Figura 4.21	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.7 obtenido en el Random 2, con R^2 de 63,49%.	33
Figura 4.22	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 2, con un R^2 de 63,49%.	33
Figura 4.23	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.7 obtenido en el Random 6, con un R^2 de 75,19%.	33
Figura 4.24	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 6, con un R^2 de 75,19%.	33
Figura 4.25	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.8 obtenido en el Random 6, con R^2 de 64,41%.	35
Figura 4.26	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 6, con un R^2 de 64,41%.	35
Figura 4.27	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.8 obtenido en el Random 5, con un R^2 de 65,61%.	35
Figura 4.28	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 65,61%.	35
Figura 4.29	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.9 obtenido en el Random 6, con R^2 de 64,73%.	37
Figura 4.30	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 6, con un R^2 de 64,73%.	37
Figura 4.31	Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 5.9 obtenido en el Random 3, con un R^2 de 65,55%.	37
Figura 4.32	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 65,55%.	37
Figura 4.33	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.10 obtenido en el Random 8, con R^2 de 42,79%.	39
Figura 4.34	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 8, con un R^2 de 42,79%.	39
Figura 4.35	Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 5.10 obtenido en el Random 3, con un R^2 de 65,22%.	39
Figura 4.36	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 65,22%.	39
Figura 4.37	Gráfico del peor resultado estimado con el entrenamiento PSO tradicional para la data observada v/s la data estimada que corresponde a la data de testing desde el año 2002 hasta el año 2007.	40
Figura 4.38	Gráfico del mejor resultado estimado con entrenamiento PSO tradicional para la data observada v/s la data estimada que corresponde a la data de testing desde el año 2002 hasta el año 2007	41

Figura 4.39	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.12 obtenido en el Random 9, con R^2 de 59,84%.	43
Figura 4.40	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 59,84%.	43
Figura 4.41	Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 5.12 obtenido en el Random 5, con un R^2 de 88,9%.	43
Figura 4.42	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 88,9%.	43
Figura 4.43	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.13 obtenido en el Random 10, con R^2 de 26,35%.	45
Figura 4.44	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 10, con un R^2 de 26,35%.	45
Figura 4.45	Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 5.13 obtenido en el Random 9, con un R^2 de 72,92%.	45
Figura 4.46	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 72,92%.	45
Figura 4.47	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.14 obtenido en el Random 5, con R^2 de -18,85%.	47
Figura 4.48	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de -18,85%.	47
Figura 4.49	Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 5.14 obtenido en el Random 4, con un R^2 de 55,48%.	47
Figura 4.50	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 4, con un R^2 de 55,48%.	47
Figura 4.51	Gráfico del descenso del error de la mejor partícula del enjambre en cada una de las épocas del sistema, con un total de 500 épocas	48
Figura 4.52	Gráfico del descenso del error de la mejor partícula del enjambre en cada una de las épocas del sistema, con un total de 1000 épocas.	49
Figura 4.53	Gráfico del descenso del error de la mejor partícula del enjambre en cada una de las épocas del sistema, con un total de 2000 épocas	49
Figura 4.54	Gráfico de la cantidad de partículas y el resultado de su coeficiente de determinación	50
Figura 5.1	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 6.2 obtenido en el Random 9, con R^2 de 71,66%.	55
Figura 5.2	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 71,66%.	55
Figura 5.3	Gráfico de la data estimada v/s la data observada de mejor resultado de la tabla 6.2 obtenido en el Random 3, con un R^2 de 91,54%.	55
Figura 5.4	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 91,54%.	55
Figura 5.5	Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 6.4 obtenido en el Random 9, con R^2 de 59,84%.	57
Figura 5.6	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 59,84%.	57

Figura 5.7	Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 6.4 obtenido en el Random 5, con un R^2 de 88,9%.	57
Figura 5.8	Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 88,9%.	57

Índice de Tablas

Tabla 1.1	Aporte nutritivo de la anchoveta.....	3
Tabla 2.1	Algoritmo PSO Tradicional.....	8
Tabla 4.1	Plan de pruebas del sistema para determinar la cantidad de entradas y la cantidad de reglas del modelo.	20
Tabla 4.2	Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas y 1 regla de PSO tradicional con 10 partículas.	22
Tabla 4.3	Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas y 2 regla de PSO tradicional con 10 partículas.	24
Tabla 4.4	Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas y 3 reglas.	26
Tabla 4.5	Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 1 regla y 10 partículas.	28
Tabla 4.6	Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 2 reglas y 10 partículas.	30
Tabla 4.7	Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 3 reglas y 10 partículas.	32
Tabla 4.8	Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 1 regla y 10 partículas.	34
Tabla 4.9	Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 2 reglas y 10 partículas.	36
Tabla 4.10	Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 3 reglas y 10 partículas.	38
Tabla 4.11	Plan de pruebas del sistema para determinar la efectividad de PSO Quantum.	41
Tabla 4.12	Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas, 3 reglas y 10 partículas.	42
Tabla 4.13	Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 3 reglas y 10 partículas.	44
Tabla 4.14	Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 3 reglas y 10 partículas.	46
Tabla 4.15	Evaluación de R^2 y ECM con distinta cantidad de épocas del sistema para determinar la cantidad óptima de épocas.	48
Tabla 4.16	Evaluación de R^2 y ECM con distinto tamaño de población del PSO	50
Tabla 4.17	Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo del peso inercial w .	51

Tabla 4.18	Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo de la variable de influencia cognitiva C_1 .	51
Tabla 4.19	Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo de la variable de influencia social C_2 .	51
Tabla 4.20	Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo de la variable de velocidad v_{\max} .	52
Tabla 5.1	Mejor configuración del sistema para Takagi Sugeno con PSO Tradicional.	53
Tabla 5.2	Resultados de la mejor configuración para Takagi Sugeno con PSO Tradicional.	54
Tabla 5.3	Mejor configuración del sistema para PSO Quantum.	56
Tabla 5.4	Resultados de la mejor configuración para Takagi Sugeno con PSO Quantum.	56

Introducción

1.1 Introducción

Hoy en día, una de las actividades de extracción más importantes de nuestro país es la pesca, ya que nuestro el territorio marítimo gracias a sistema de corrientes marinas de Chile-Perú, sustenta uno de los ecosistemas marinos más productivos de todo el mundo. Es por ello que las pesqueras invierten gran cantidad de recursos en para mejorar la extracción de especies, sin embargo, algunos de los grandes problemas a los cuales se ven enfrentados es el desplazamiento de los banco de peces y la cantidad de abundancia de peces en cierto período de tiempo, lo cual provoca un desperdicio de esfuerzo, recursos y de mano de obra si se escoge mal la cantidad de flota pesquera y/o el tiempo destinado a la extracción.

Las pesquerías de recursos pelágicos sitúan a Chile como una potencia mundial en términos de captura. Dichos desembarques se sustentan principalmente en la extracción de unas pocas especies pelágicas y demersales. En nuestro país, los principales recursos pelágicos son: jurel, anchoveta, sardina española, caballa y sardina común, los cuales representan una fracción relevante de la captura total (SERNAPESCA, 2001) [1].

En 1996 cerca de un 20% de las capturas de especies pelágicas mundiales se realizaron en la zona norte de nuestro país, la cual representa un 0,09% de la superficie del océano mundial. Varios factores importantes sustentan la alta productividad biológica observada en este sistema de borde oriental. Estos incluyen la corriente subsuperficial de Perú-Chile, la cual transporta aguas ricas en nutrientes y pobres en oxígeno en dirección sur y la persistente sugerencia costera generada por la acción de los vientos predominantes, cuyo principal efecto consiste en el afloramiento de aguas ricas en nutrientes hacia la superficie, favoreciendo la alta producción de fitoplancton disponible para el zooplancton y los peces. Observaciones directas de corrientes medidas sobre el talud continental frente a Chile, han revelado una compleja estructura de variabilidad en la banda intraestacional, estacional e interanual. Mucha de esta variabilidad aparece forzada remotamente desde el Pacífico ecuatorial, por las oscilaciones de Madden-Julian en la banda intraestacional por los vientos ecuatoriales en la banda estacional y por El Niño y la oscilación del sur en la banda interanual. Esta variabilidad vinculada a la dinámica ecuatorial abre ahora un potencial de predictibilidad de las condiciones oceanográficas frente a Chile con varios meses de anticipación.

Chile es uno de los principales países pesqueros a nivel mundial, con aproximadamente 5 millones de toneladas de capturas anuales. Las especies pelágicas constituyen cerca de un 90% de las capturas anuales, donde la pesquería pelágica de la zona norte de Chile (18°21'S- 24°00'S) contribuye con un 48% (valor promedio entre 1951-1998; SAG, 1957-77, SERNAP, 1978-1994 y SERNAPESCA, 1995-97) lo cual la posiciona como la segunda área más importante del país. Esta pesquería se realiza desde la

década del 50, primero y esencialmente sobre la Anchoqueta (*Engraulis ringens*), especie en que nos centraremos en nuestro proyecto. Esta pesquería colapsa a mediados de la década del 70, siendo reemplazada por la de la Sardina (*Sardinops sagax*).

Después de 1975 también aumentan las capturas de Jurel (*Trachurus murphy*) y Caballa (*Scomber japonicus*), representando en el norte de Chile el 15% y 4% de los desembarques del período 1976-2002; en tanto la Anchoqueta y Sardina representan el 34% y 46% respectivamente (SAG, 1976-77; SERNAPESCA, 1978-2002).

1.2 La Anchoqueta

La Anchoqueta es familia de los Clupeidos, ubicada en el género *Engraulis*. Es una de las especies pelágicas de mayor importancia debido a los grandes volúmenes de captura anual en el ámbito mundial. Viven en aguas oceánicas cuya temperatura se encuentra entre 14,5 y los 20°C. La anchoqueta vive entre los 3 o 4 años de edad y en su etapa adulta, alcanza una longitud que oscila entre los 12 y 16 centímetros [2].

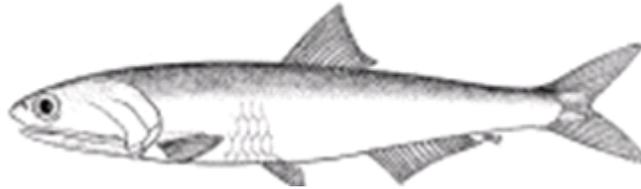


Figura 1.1: Imagen de un ejemplar de Anchoqueta

Las anchoquetas se alimentan del abundante plankton existente en la zona norte de Chile y sur de Perú, producto de la corriente peruana, como no existe ni en abundancia ni calidad en ninguna otra parte del mundo. Durante la primavera y el verano normales, la anchoqueta se encuentra dentro de una franja costera hasta los 36 a 54 km de la costa; en el otoño e invierno llega a desplazarse hasta los 144 km, y en algunas ocasiones, hasta 180 km de la costa.

1.3 Hábitat

Los cardúmenes de anchoquetas se desplazan en aguas superficiales de hasta 50 metros de profundidad en el día y suben en la noche. Se las encuentra entre los 03°30' Sur y los 37°00' Sur. En este espacio hay dos grandes zonas donde se encuentran anchoquetas, la franja norte-centro del Perú que es la más importante y la franja del sur del Perú-norte de Chile que es menos poblada. Las aguas en las que vive la anchoqueta debe tener entre 15 y 21 grados de temperatura y la salinidad debe encontrarse entre los 34,5 y 35,1 UPS. Es decir para que la anchoqueta viva y se reproduzca normalmente, debe tener a su disposición alimento, aguas con temperatura adecuada y salinidad moderada a la profundidad normal para ellas. Cuando las aguas superficiales se calientan, las anchoquetas se profundizan hasta cierto límite, porque con la profundidad la salinidad aumenta y también escasea el

plankton. Por esa razón, si ocurre algún cambio climático, los cardúmenes se desplazan buscando el hábitat adecuado.

1.4 Reproducción

Se reproducen en cualquier época del año, sin embargo, los mayores desoves se producen, uno al final del invierno y otro al final del verano. Una hembra adulta produce millares de huevos durante su vida, desovando en la superficie y hasta 50 metros de profundidad.

1.5 Aporte nutritivo

La anchoveta es una excelente fuente de proteína animal de alta calidad. Su alto contenido de lisina y otros aminoácidos esenciales la hacen especialmente adecuada para el complemento de dietas ricas en carbohidratos. Es un recurso muy rico en micro nutriente que no son usualmente encontrados en alimentos básicos [3].

GRASA (100GRS)	8.2
HUMEDAD (100GRS)	70.8
PROTEINAS (100GRS)	19.1
SALES MINERALES (100GRS)	1.2
CALCIO (100GRS)	77.1
POTASIO (100GR)	241.4
SODIO (100GRS)	78.0
MAGNESIO (100GRS)	31.3
CALORIAS (100GRS)	185.0
OMEGA 3 (100GRS)	90.9

Tabla 1.1: Aporte nutritivo de la Anchoveta

Es muy recomendable el consumo en los recién nacidos, son esenciales para la formación del tejido nervioso y en la función visual; en los niños de edad escolar para aumentar su coeficiente intelectual; en adultos y ancianos para mejorar su memoria; en las mujeres previene el cáncer de mama y al colon.

1.6 Importancia del problema

La importancia de resolver el problema planteado es poder generar una herramienta que ayude en la toma de decisiones en las tareas de extracción de la anchoveta en el norte de nuestro país, logrando así resultados óptimos en la extracción de este importante recurso natural y a la buena disposición de los recursos destinados a ella.

Como ya se mencionó anteriormente, la anchoveta es un recurso natural sumamente importante, el cual provee de mucho aporte nutritivo al ser humano. Debido a su gran abundancia, constituye una fuente importante de ingresos a nuestro país, como también entrega una fuente considerable de trabajo para la población del norte de nuestro país.

Además es importante generar nuevos sistemas de pronóstico, ya que la competitividad en la extracción de este recurso es un factor de éxito primordial para las empresas pesqueras del norte de Chile.

1.7 Objetivos general y específicos

General

Desarrollar y evaluar un modelo fuzzy Takagi Sugeno usando algoritmo de optimización de enjambre de partículas (PSO) para pronosticar las capturas mensuales de anchovetas en el área norte de Chile.

Específicos

- Comprender el modelo fuzzy Takagi-Sugeno y el algoritmo PSO
- Diseñar e implementar un modelo de pronóstico fuzzy usando algoritmo PSO.
- Evaluar el rendimiento del modelo de pronóstico propuesto

Optimización por enjambre de partículas

2.1 Introducción

Este capítulo analiza una tecnología que hoy en día tiene una gran importancia en materias de resolución de problemas complejos y optimización. Con el pasar de los tiempos el hombre ha elaborado herramientas para mejorar sus tareas y procesos que están involucrados en su quehacer diario, como por ejemplo modelos lineales matemáticos y estadísticos enfocados a entregar resultados que maximicen la producción y minimicen los costos. Siguiendo con esa línea, en este capítulo se explican los algoritmos de optimización por enjambre de partículas, que son modelos estocásticos de análisis de datos enfocados a la optimización.

Debido al gran potencial de los algoritmos de optimización por enjambre de partículas, sus implementaciones las podemos encontrar en una gran cantidad de áreas, como por ejemplo encontrar la mejor manera de alimentar una agrupación lineal de antenas de transmisión que cubrirá una cierta zona geográfica, que se describe con más detalle en [31], en donde el PSO se utiliza para estimar los fasores de la distribución de corriente de cada uno de los elementos que componen la agrupación, de forma que cumplan con los niveles de radiación permitidos y que la distribución de alimentación sea la adecuada.

Dentro de todo tipo de empresa es muy importante el tema de la, ya que de ella depende, en gran manera, su rentabilidad, tal como podemos apreciar con más detalle en [5]. Es en este campo donde también vemos aplicados los algoritmos de PSO resolviendo problemas de secuenciamiento de tareas (*WorkShop*), para minimizar el tiempo de procesamiento de todas las tareas, minimizar la penalización por tardanzas en las fechas de entrega, minimizar el tiempo ocioso de las máquinas, etc.

Para tales efectos, se analizará el enjambre de partículas y en que consiste, realizando un análisis en profundidad de los principios del PSO como método de optimización. Se analizará los esquemas que se han implementado, con el objeto de estudiar la influencia de los principales parámetros del algoritmo sobre la convergencia de la optimización, buscando una buena relación entre la precisión del modelo y el coste computacional que hay involucrado. Se analizará sus topologías y las formas de actualización del.

2.2 Organización del texto

Para comenzar, en el punto 2.3 se explica qué es un enjambre de partículas, sus orígenes y primeras implementaciones, la lógica empleada en el algoritmo de optimización, sus partículas y la interacción que emplean basándose en modelos colectivos. El punto 2.4 describe la implementación del modelo tradicional de PSO, explicando cómo está conformada una partícula y su función de desplazamiento que permite que las partículas se muevan en el espacio en busca de las soluciones del problema. El punto 2.5 explica con mayor detalle los parámetros mencionados en el punto anterior y describe la importancia de cada uno de ellos, parámetros tales como la función *fitness*, el peso inercial w , velocidad máxima y velocidad mínima, variables de influencia cognitiva y social. El punto 2.6 muestra sus topologías local y global. El punto 2.7 muestra los tipos de actualización de las partículas del enjambre de modo síncrono y asíncrono. El punto 2.8 explica la importancia de la exploración y explotación de PSO. El punto 2.9 describe una implementación de PSO con restricción de velocidad mínima para controlar que la velocidad las partículas no descienda demasiado rápido. El punto 2.10 muestra el modelo PSO Quantum, explicando sus diferencias con PSO tradicional.

2.3 El enjambre y su inteligencia de optimización

Un algoritmo de optimización basado en enjambre de partículas es una técnica metaheurística que se basa en poblaciones de individuos e inspirada en el comportamiento social del vuelo las bandadas de pájaros y el desplazamiento de un cardumen de peces.

Durante esta última década, se ha observado un interés creciente por el desarrollo de métodos para solucionar problemas complejos de optimización uniobjetivo y multiobjetivo, incluyendo su aplicación a problemas combinatorios no lineales [6]. Los algoritmos metaheurísticos constituyen uno de los campos de investigación más activos en optimización. Su progreso está siendo rápido, surgiendo continuamente nuevas ideas en un intento de alcanzar una mayor eficiencia en el proceso de solución.

Actualmente los métodos de optimización heurísticos basados en poblaciones han despertado el interés de la comunidad científica debido a su capacidad para explorar espacios de soluciones multimodales y multidimensionales [7,8], de manera rápida y eficiente. La optimización por enjambre de partículas (PSO) pertenece a las técnicas estocásticas de cálculo evolutivo, considerada unas de las representantes de la rama de la inteligencia de enjambre (Swarm Intelligence), al igual que la técnica optimización por colonia de hormigas (ACO), los cuales son los métodos más populares y utilizados en el área de la inteligencia computacional. Ambos tratan de imitar los comportamientos sociales de un colectivo a partir de la interacción de los individuos entre sí y con el entorno [9].

Los inicios del PSO como método estocástico de optimización global se remontan a los estudios realizados por Kennedy y Eberhart en [10], quienes intentaron simular gráficamente el movimiento sincronizado e impredecible de grupos tales como los bancos de peces o las bandadas de aves, motivados por la capacidad que poseen estos grupos para separarse, reagruparse o encontrar alimento. Paralelamente con trabajos previos en el ámbito de la biología y de la sociología, concluyen que el comportamiento, inteligencia y movimiento de estas agrupaciones (entre las cuales podríamos incluir a los seres humanos con un cierto grado de abstracción) está relacionado directamente con la capacidad para

compartir información y para aprovechar la experiencia acumulada por cada uno de sus congéneres.

En la tecnología utilizada en [10], Kennedy y Eberhart introducen el término general de partícula o agente para representar a los peces, pájaros, abejas, hormigas o cualquier otro tipo de individuos que exhiban un comportamiento social como grupo, en forma de una colección de agentes que interactúan entre sí. De acuerdo con los fundamentos teóricos del método, el movimiento de cada una de estas partículas hacia un objetivo común en dos dimensiones está condicionado por dos factores básicos: la memoria autobiográfica de la partícula o nostalgia, y la influencia social de todo el enjambre. A nivel computacional, como método de optimización, esta filosofía puede extenderse a un espacio d -dimensional de acuerdo con el problema bajo análisis. La posición instantánea de cada una de las partículas de la población en el espacio representa una solución potencial, siendo d el número de incógnitas del problema original. Básicamente, el proceso evolutivo se reduce a mover cada partícula dentro del espacio de soluciones con una velocidad que variará de acuerdo a su velocidad actual, a la memoria de la partícula y a la información global que comparte el resto del enjambre, utilizando una función de aptitud (*fitness*) para cuantificar la calidad de cada partícula en función de la posición que esta ocupe [11].

Más allá de la propia naturaleza del método, los esquemas existentes para su implementación son muy diversos. Dependiendo de cómo se actualicen las posiciones de las partículas surgen las versiones síncrona y asíncrona del algoritmo. Adicionalmente dependiendo de cómo se haga fluir la experiencia acumulada por el enjambre sobre el movimiento de cada una de las partículas que lo integran, se puede distinguir entre PSO local y global. La combinación de estas cuatro variantes resume los esquemas desarrollados e investigados comúnmente, pero en la literatura existen múltiples variantes a los esquemas convencionales, en la mayoría de los casos fruto de modificaciones introducidas por los propios autores para mejorar el rendimiento del algoritmo original en aplicaciones concretas [12].

Desde el punto de vista de su algoritmo, la ventaja principal del PSO es su rápida convergencia, en comparación con otros algoritmos de optimización global, como por ejemplo los algoritmos genéticos (GA), colonias de hormigas, enfriamiento simulado, por mencionar algunos [13-14]. Para aplicar PSO de forma exitosa, uno de los problemas claves se encuentra en como representar la solución en las partículas del PSO. Ésta representación de la solución del problema afectará directamente en la facilidad del uso del algoritmo y su funcionamiento.

2.4 Modelo tradicional

Una partícula está definida como un vector, el cual está compuesto por la velocidad que posee la partícula, su posición, y su memoria [6]. El algoritmo PSO se inicializa con partículas (soluciones) aleatorias, y a medida que se realicen las iteraciones, buscará soluciones más óptimas actualizando cada partícula. Una partícula se considera como un punto dentro del espacio de búsqueda D -dimensional, donde cada dimensión representa una de las incógnitas del problema. El enjambre es representado por $X = (x_{i1}, x_{i2}, \dots, x_{id})$, donde $i = 1, 2, \dots, n$ corresponde la i -ésima partícula, y $d = 1, 2, \dots, D$ corresponde sus dimensiones.

En cada generación (iteración), cada una de las partículas es actualizada de acuerdo a dos valores importantes. El primero valor corresponde a la mejor posición encontrada por

una partícula, conocido como mejor personal p_{best} . El p_{best} de una i -ésima partícula es representado como $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$. El segundo valor corresponde a la mejor posición que ha encontrado el enjambre, conocido como el mejor global p_{gbest} . Ambos valores se actualizan en cada iteración, y se utilizan para ajustar la velocidad con que se mueve una determinada partícula en cada una de las dimensiones. A su vez, la velocidad es utilizada para determinar la nueva posición que alcanzará una partícula en la siguiente iteración del algoritmo. La influencia que tiene la mejor posición personal sobre la velocidad de una partícula se conoce como el factor o componente de cognición, y la influencia de la mejor posición del enjambre se conoce como el componente social [6].

En [11] Eberhart incorporó un factor de inercia w en la ecuación que determina la velocidad (2.1) de la partícula para balancear la búsqueda global y local. La nueva posición de una partícula está definida en (2.2). El algoritmo PSO se presenta en la tabla 2.1.

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \quad (2.1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2.2)$$

Donde r_1 y r_2 corresponden a valores aleatorios independientes en el intervalo $[0,1]$, y los parámetros c_1 y c_2 corresponden a las variables que controlan la influencia de las componentes cognitiva y social. Para prevenir que las velocidades de las partículas incrementen infinitamente, se incorpora un parámetro V_{max} que delimita el rango de velocidad $[-V_{max}, V_{max}]$ que puede tomar una partícula.

$$\begin{aligned} & \text{if } (v_{ij} > V_{max}) \quad \text{or} \quad (v_{ij} < -V_{max}) \\ & \quad v_{ij} = \text{sign}(v_{ij}) \cdot V_{max} \end{aligned} \quad (2.3)$$

La posición $x_{id}(t+1)$ expresada en (3.2) es la nueva posición que adopta la partícula $x_{id}(t)$, lo que se traduce en que (t) es el mes presente y $(t+1)$ es el pronóstico del mes siguiente.

Tabla 2.1 Algoritmo PSO Tradicional.

<p>Paso 1: Configurar el tamaño del enjambre n e inicializar las posiciones y velocidades de las partículas en un espacio D-dimensional.</p>
<p>Paso 2: Para cada partícula, evaluar su <i>fitness</i> de acuerdo algún criterio de optimización (función de costo, mínimos cuadrados, error cuadrático medio).</p>
<p>Paso 3: Comparar el <i>fitness</i> actual de cada partícula con el <i>fitness</i> de su mejor posición personal encontrada hasta el momento. Si el <i>fitness</i> actual resulta ser mejor, entonces se actualiza la mejor posición personal de la partícula p_{id} al valor de la posición actual x_{id}.</p>
<p>Paso 4: Comparar el <i>fitness</i> actual de cada partícula con el mejor <i>fitness</i></p>

encontrado por el enjambre. Si el *fitness* actual resulta ser mejor, entonces se actualiza la mejor posición global p_{gd} al valor de la posición actual x_{id} .

Paso 5: Ajustar las velocidades y posiciones de la partículas de acuerdo a las ecuaciones (2.1) y (2.2). Verificar si las partículas sobrepasan la velocidad máxima definida en (2.3).

Paso 6: Verificar si se cumple algún criterio de termino (máximo número de iteraciones, valor de *fitness* alcanzado, número de intentos de mejoramiento de la solución), si no ir al Paso 2.

El éxito del algoritmo, radica en su capacidad de ajustar las posiciones de las partículas en un área del espacio de soluciones prometedora, de acuerdo a una función objetivo que se desea minimizar o maximizar. Es decir, la función objetivo evalúa la aptitud o *fitness* de una partícula, y se utiliza para guiar la búsqueda durante el proceso de optimización [11].

Las condiciones de término comúnmente usados para finalizar el proceso de optimización son:

1. Número máximo de iteraciones: El proceso de iteración termina después que alcanza un número fijo de iteraciones, como por ejemplo 1000 iteraciones.
2. Número de iteraciones sin mejoras: El proceso de iteración termina después que se alcanza un número fijo de iteraciones en la no se han obtenido alguna mejora en términos de solución.
3. Error mínimo de la función objetivo: El error entre el valor obtenido de la función objetivo y el mejor valor de eficacia (*fitness*) es menor que un umbral prefijo esperado.

A pesar de que el algoritmo de PSO posee un reducido número de parámetros, el mal ajuste de ellos puede provocar que el PSO converja a un óptimo en pocas iteraciones, o a un buen óptimo en muchas iteraciones. A menudo, el PSO podría encontrar un mal óptimo en pocas iteraciones (fenómeno conocido como convergencia prematura), o a un mal óptimo en muchas iteraciones [15-16]. A continuación se estudiará cada parámetro del algoritmo PSO y la influencia que tiene sobre este.

2.5 Parámetros de PSO

Como en cualquier otro método de cómputo evolutivo, el ajuste de los parámetros del PSO condiciona el rendimiento posterior del algoritmo de optimización. Aunque existen múltiples estudios que intentan generalizar la selección de estos parámetros, dicha selección está íntimamente ligada a la naturaleza del problema a optimizar, siendo preciso lograr un balance óptimo entre exploración y convergencia [17]. Al margen de los parámetros intrínsecos del algoritmo, debe resaltarse la importancia que adquiere la función de *fitness*. La utilización de una función de *fitness* que no introduzca una métrica adecuada para pesar la bondad de cada partícula, hará emerger las carencias del PSO que erróneamente pueden asociarse con el algoritmo en sí, aún siendo más propias de los defectos de la propia

función de *fitness* [6]. En esta sección se resumen los valores típicos para los parámetros del algoritmo.

El movimiento de los partículas sobre el espacio de soluciones y, en consecuencia, el rendimiento del algoritmo, está condicionado por el grado de contribución de las tres componentes de la velocidad en la primera ecuación, ya introducidas, y denominadas por este orden como momento, hábito o inercia para considerar la tendencia de la partícula; memoria, nostalgia o auto-aprendizaje para incluir la experiencia de la propia partícula, y cooperación, conocimiento social, conocimiento de grupo o información compartida, para reflejar el intercambio de información y el comportamiento social como grupo [6].

Para acotar la velocidad de la partícula se especifica un valor máximo, V_{max} , que restringe la velocidad en cada dimensión al intervalo $[-V_{max}, V_{max}]$. Si el valor de V_{max} toma valores extremadamente pequeños las partículas explorarán el espacio de soluciones muy lentamente y podrán quedar atrapadas alrededor de soluciones locales, incapaces de librarse de la base de atracción [12]. Sin embargo, si no se introdujese el parámetro de control V_{max} , el enjambre no convergería hacia un punto, sino que sufriría el fenómeno conocido como explosión del PSO, consistente en un comportamiento oscilatorio y creciente de la posición de las partículas, provocando la ineficiencia del PSO como algoritmo de optimización [11].

El efecto de V_{max} está fuertemente vinculado con la naturaleza del problema a optimizar, y no existe ninguna regla que dicte las pautas a seguir en su elección para asegurar la convergencia del enjambre. Con el objetivo de reducir el efecto de V_{max} y perfeccionar el control del alcance de la búsqueda sobre el espacio de soluciones, se incorporó el concepto de peso inercial [6].

El peso de inercia w en la ecuación (2.1), es considerado un parámetro crítico para la convergencia de PSO. El peso inercial se ocupa para controlar el impacto de la historia previa de las velocidades actuales. En consecuencia, el parámetro w regula la compensación entre las capacidades de exploración global (amplias) y locales (cercanas) del enjambre. Un peso inercial grande facilita la exploración global (buscando nuevas áreas). Sin embargo, un valor pequeño para este parámetro tendería a facilitar la exploración local, es decir, ajusta al algoritmo al área actual de búsqueda. Un valor conveniente para este parámetro por lo general proporciona un balance entre las habilidades de exploración global y local, y por consiguiente, causa una reducción del número de iteraciones requeridas para localizar la solución óptima [11]. Al principio, el peso de inercia es asignado como un constante. Sin embargo, algunos resultados experimentales indican que es mejor asignarle un valor grande al principio, para promover la exploración global del espacio de búsqueda, y gradualmente disminuirlo para conseguir soluciones más refinadas.

En [6] se menciona que una buena opción para el parámetro w sería asignarle un valor inicial alrededor 1.0 y reducirlo hacia cero a medida que se itere el algoritmo. Otro método mejor aun, sería usar alguna aproximación adaptable (un controlador difuso por ejemplo), en la cual este parámetro pueda ser sintonizado según las características del problema a considerar.

Los parámetros c_1 y c_2 , en la ecuación (2.1), no son críticos para la convergencia de PSO. Sin embargo, si se sintonizan apropiadamente podrían acelerar la convergencia del algoritmo y el aliviarlo de mínimos locales. Los valores típicos para estos parámetros son $c_1 = c_2 = 2$ [6], pero algunos resultados experimentales indican que se pueden asignar ambos parámetros como constantes $c_1 = c_2 = 1.49$. Los trabajos recientes revelan que podría ser aún mejor, escoger un parámetro cognitivos más grande, c_1 , que un parámetro social, c_2 , siempre que se cumpla la siguiente expresión $c_1 + c_2 \leq 4$ [11]. Sin embargo, existen varias

propuestas de ajustes de estos parámetros de acuerdo al avance del proceso de optimización, para que se adapten de acuerdo al rendimiento del algoritmo mediante chequeos de indicadores tales como: calidad del óptimo según el número de iteraciones, cantidad de intentos de mejora de la solución, entre otros [18,19]. Más adelante se definirán los modos de operación del algoritmo conocidos como exploración y explotación.

Otro parámetro que precisa ser seleccionado con extrema rigurosidad es el tamaño de la población. Poblaciones excesivamente grandes exploran minuciosamente el espacio de búsqueda, pero como contrapartida, el coste computacional asociado con el aumento del número de evaluaciones de la función de *fitness* puede resultar excesivamente elevado. Los estudios paramétricos revelan que una población de alrededor de 30 partículas es suficiente para múltiples problemas, y típicamente se utilizan poblaciones que oscilan entre 10 y 50 partículas, o entre 100 y 200 partículas para abordar problemas más complejos [6].

2.6 Topologías Local y Global

El algoritmo PSO puede ser descrito, de manera general, como una población de vectores cuyas trayectorias oscilan alrededor de una región que está definida por el mejor éxito personal de cada individuo y el éxito de alguna otra partícula. Se han usados varios métodos para identificar otra partícula que influya en el individuo. Los autores de [11], Eberhart y Kennedy, llamaron a los dos métodos básicos como “modelo del óptimo global” y “modelo del óptimo local”. En el modelo del óptimo local, las partículas tienen sola la información de sus vecinos más cercanos del enjambre, mientras que el modelo del óptimo global las partículas consideran la información de todo el enjambre.

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{ld}(t) - x_{id}(t)) \quad (2.4)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2.5)$$

En el modelo del óptimo global, la trayectoria de búsqueda de cada partícula es influenciada por la mejor posición encontrada por cualquier miembro de la población. La mejor partícula atrae a todas las partículas hacia ella. El modelo del óptimo local permite que cada individuo sea influenciado por un reducido número de miembros adyacentes de la población. Las partículas seleccionadas para ser un subconjunto del enjambre no tienen relación directa con el resto de las partículas que pertenecen a otra vecindad. Generalmente el óptimo local abarcan a dos vecinos.

Los autores de [6] declaran que el modelo de óptimo global converge rápidamente hacia las soluciones del problema, pero tiene la debilidad de quedar atrapado en óptimos locales, mientras el modelo de óptimo local tiene la desventaja de converger lentamente hacia la solución del problema. Por ello, los autores de [6] recomiendan usar el modelo de óptimo global fuerte para funciones unimodales, mientras recomiendan a un modelo local variable para funciones multimodales.

2.7 Actualización síncrona y asíncrona

Al atravesar el espacio d -dimensional, las partículas actualizan sus vectores de posición y velocidad de acuerdo a su posición y velocidad actual, su memoria y el conocimiento social. En función del instante dentro del proceso iterativo en el que se realiza la actualización de la memoria de cada partícula y el conocimiento social del grupo, se puede distinguir entre PSO con actualizaciones síncronas y asíncronas de la población.

En el modelo síncrono, todas las partículas se mueven en paralelo. En cada iteración se evalúa el *fitness* de todas partículas, se actualiza su memoria p_{id} y el conocimiento social p_{gd} , y el enjambre se desplaza hacia un nuevo punto tomando como referencia esta dupla de información. Es decir, todas las partículas comparten la misma información acerca de la mejor solución de partida p_{gd} .

Por el contrario, en el PSO asíncrono cada partícula aprovecha, al desplazarse, la información actualizada por sus inmediatos predecesores. Es decir, en cada iteración k , la partícula i -ésima se desplaza hacia un nuevo punto $x_i(k+1)$ utilizando la información de los vectores $p_{id}(k)$ y p_{gd} , actualizados por las $i-1$ partículas previas [12]. Posteriormente, la partícula evalúa la calidad del nuevo punto y actualiza, si procede, las variables $p_{id}(k+1)$ y p_{gd} . Esta información se transmite a las restantes partículas.

Al actualizar la información partícula a partícula, el modelo asíncrono acelera la optimización, aunque la naturaleza del modelo síncrono lo hace susceptible de ser ejecutado en paralelo, sobre múltiples procesadores. Por lo tanto, las distintas topologías y modos de actualización de las memorias de las partículas, hacen que el algoritmo PSO se comporte de diferentes maneras, por lo que encontrar una topología ideal para el problema en cuestión es parte del proceso no trivial de sintonización de parámetros [12].

2.8 Exploración y explotación de PSO

Como se revisó en la sección anterior, la sintonización de parámetros del algoritmo PSO, y de cualquier otro algoritmo evolutivo, es un proceso no trivial y a la vez costoso en términos de tiempo de prueba [19]. Su importancia radica en que de ello depende el buen funcionamiento del algoritmo y la calidad del óptimo encontrado. Uno de los problemas que se presentan en los algoritmos evolutivos, es que en ocasiones quedan atrapados en óptimos locales, es decir, una solución buena (o no muy buena en algunos casos), pero no la mejor existente dentro del espacio de búsqueda. Este fenómeno se conoce dentro de la literatura como convergencia prematura del algoritmo, en donde este es incapaz de salir de la zona en donde encontró un óptimo local, impidiendo que este explore otras regiones del espacio de búsqueda [17].

En relación a lo expresado en el párrafo anterior, se puede concluir que lo ideal es que las partículas (o población) representen soluciones de diferentes zonas del espacio de búsqueda, es decir, lo más diversificado posible. Este concepto se conoce como diversificación de la población, y en términos del algoritmo, se refiere a que las partículas sobrevuelan diferentes zonas del espacio de solución antes de que se comprometan con alguna zona. Por otro lado, una de las características que debe tener todo algoritmo de optimización, es su capacidad de converger a una solución óptima, conocida como convergencia de la población, lo cual se refiere a que las partículas, una vez conglomeradas

en una zona prometedora, sean capaces de mejorar la solución que hayan encontrado, es decir, que logren encontrar y converger a un óptimo cada vez mejor [19,20].

Estas dos características de los algoritmos evolutivos, incluyendo el área de la inteligencia de enjambre, pueden ser representadas por dos modos, distintos pero esenciales, de operar el algoritmo. Estos modos se conocen en la literatura como exploración y explotación [21]. Se define como exploración, a la búsqueda de la solución del problema en el espacio de búsqueda por un rango amplio, es decir, que las partículas vuelen por diferentes zonas del espacio de búsqueda privilegiando la diversificación del enjambre. Por otro lado, se define como explotación a la búsqueda de la solución del problema encontrando un rango (sub-espacio) dentro del espacio de búsqueda, en donde el enjambre trate de tomar todos los valores posibles que comprenden dicho rango, privilegiando la convergencia del enjambre [19].

Los estudios realizados en [11], revelan que valores altos para el coeficiente de inercia, factores de cognición y V_{max} ($0.75 < w < 1$ y $2 < c_1, c_2 < 4$), favorecen el modo exploración del PSO, mientras que valores bajos para estos parámetros ($0.4 < w < 0.75$ y $0.1 < c_1, c_2 < 2$) favorecen el modo explotación del PSO.

Lo ideal es ajustar estos parámetros para que el algoritmo funcione en modo exploración al comienzo de la ejecución, y que cambie al modo explotación una vez transcurridas una cierta cantidad de iteraciones o haya alcanzado un cierto óptimo. Debido a esto, en la actualidad se han propuesto una serie de técnicas adaptativas y auto-adaptables [19-22], en donde los parámetros son ajustados, o mejor dicho se auto ajustan, durante la ejecución del algoritmo, cambiando de modo exploración a explotación y de explotación a exploración, según como sea el rendimiento del algoritmo, al chequear indicadores de rendimiento [23]. Estos indicadores, también conocidos como métricas, son definidos por los usuarios para controlar el progreso del proceso de optimización, mediante el ajuste conveniente de sus parámetros.

2.9 PSO con restricción de velocidad mínima

Los autores de [24] señalan que el algoritmo PSO puede no converger a un mínimo global, debido a que las partículas podrían escaparse del espacio de búsqueda cuando sus velocidades incrementan rápidamente, o podrían detenerse cuando sus velocidades decrecen rápidamente, quedando atrapadas en un óptimo local. Por ello, en [24] se propone la incorporación de un parámetro V_{min} a modo de velocidad mínima, para impedir que las partículas descendan rápidamente su velocidad, y evitar que el algoritmo PSO converja prematuramente. Para ello, los autores toman la misma estructura diseñada para el V_{max} , es decir, después que es actualizada las velocidades de las partículas, revisar si estas bajan su velocidad por debajo de la velocidad mínima según (2.6).

$$if(v_{id} < V_{min}) and (v_{id} > -V_{min})$$

$$v_{id} = sign(v_{id}) \cdot \bar{v} \tag{2.6}$$

Donde \bar{v} representa el promedio de las velocidades de las partículas según (2.7), siendo D las dimensiones de las partículas y M el tamaño del enjambre.

$$\bar{v} = \frac{1}{M \cdot D} \sum_{i=1}^M \sum_{d=1}^D (|v_{id}|) \quad (2.7)$$

La propuesta se probó con 4 funciones de prueba patrón, en las cuales se obtuvo una buena mejora en términos de calidad del óptimo encontrado al evitar la convergencia prematura del algoritmo [13]. Sin embargo, se podría hacer menos costoso el cálculo del promedio de la velocidad del enjambre, al considerar solo el promedio de la dimensión en la cual la velocidad decreció por debajo de la velocidad mínima, entregando así un promedio menos plano al no considerar todo el enjambre.

2.10 PSO Quantum

Para analizar otra posible solución al ajuste de los parámetros del sistema de inferencia difusa Takagi Sugeno, se implementará una variante de PSO, la cual elimina el concepto de función de velocidad y la reemplaza por una función de movimiento (2.8) tomando en consideración la participación del resto del enjambre.

Esta función de movimiento propuesta por los autores de [9] compara la posición actual de una partícula con el promedio de las mejores posiciones del enjambre, a diferencia del PSO tradicional que compara la posición de una partícula con la mejor posición del enjambre.

$$x(t+1) = p \pm \beta \cdot |m_{best} - x(t)| \cdot \ln(1/u) \quad (2.8)$$

Donde m_{best} representa el vector con la media de todas las mejores posiciones encontradas por las partículas a nivel de dimensión. Calcular la media de las mejores posiciones en cada iteración del algoritmo tiene un alto coste computacional, por lo que se debe considerar un número reducido de partículas.

$$m_{best} = \frac{1}{M} \sum_{i=1}^M p_i = \left(\frac{1}{M} \sum_{i=1}^M p_{i1}, \frac{1}{M} \sum_{i=1}^M p_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M p_{id} \right) \quad (2.9)$$

El parámetro p en (2.8) depende de la mejor posición de la partícula p_{id} y de la mejor posición global según:

$$p = \varphi \cdot p_{id} + (1 - \varphi) \cdot p_{gd} \quad (2.10)$$

Donde φ y u son valores aleatorios entre $[0,1]$, β es un coeficiente de expansión-contracción, tomando un valor entre un rango de $[0,1.7]$.

Modelo difuso Takagi Sugeno

3.1 Introducción

Los modelos difusos constituyen una herramienta importante para dar respuestas a problemas que carecen de comportamiento lineal, a los cuales, para dar una explicación u obtener su resultado, no se puede aplicar una fórmula matemática.

El motivo de utilizar del modelo de inferencia difusa Takagi Sugeno se sustenta bajo el argumento de que la cantidad de tonelaje de pesca obtenido en un mes determinado no tiene una relación lineal con respecto al mes anterior o los meses inmediatamente anteriores, lo cual lo convierte en un problema sin linealidad, en el cual, como ya se ha mencionado anteriormente, no se puede aplicar una función matemática o una correlación lineal estadística.

El modelo difuso TS se utilizó en un modelo a escala piloto de un intercambiador de calor de tipo serpentín, tal como se explica en [4]. Es usado para calentar alguna sustancia líquida que se encuentra en el tanque por medio del intercambio de calor con el vapor que pasa a través del tubo serpentín.

Se utiliza en un sistema de control de tolerancia a fallos de un motor de inducción de un vehículo de riel. Como se explica en [27], se necesita regular el arranque y el flujo del motor, donde el arranque y flujo no pueden ser medidos, pero si pueden ser estimados usando un observador bi-lineal que alimenta al controlador difuso TS.

El este capítulo se explica el modelo de inferencia difusa Takagi Sugeno, se analizará sus características y componentes, concluyendo con presentar algunos de los problemas en los cuales se ha implementado.

3.2 Organización del texto

Para comenzar, el punto 3.3 presenta la definición de un modelo de inferencia difusa de tipo Takagi Sugeno y sus componentes. El punto 3.4 explica los datos de entrada del modelo de inferencia difusa. El punto 3.5 explica qué son las reglas del modelo y cómo están formadas, introduce los conceptos de antecedente y consecuente. El punto 3.6 explica los conjuntos difusos del modelo y su función de membresía. El punto 3.7 explica los datos de salida del modelo y presenta un diagrama de cómo se compone el modelo difuso.

3.3 Takagi Sugeno

El modelo Takagi Sugeno (TS) es un sistema de inferencia difusa (SID), en el cual el conocimiento entre el dominio de entrada X y el dominio de salida Y se expresa como un conjunto de reglas de tipo Si – Entonces, en las cuales existe un ‘antecedente’ y un ‘consecuente’. TS posee datos de entrada, reglas, función de membresía y una salida, los cuales se explican con mayor detalle a continuación.

3.4 Datos de entrada del modelo

La entrada al sistema es un vector de datos del tipo

$$X = \{x_1, x_2, x_3, \dots, x_n\} \quad (3.1)$$

Donde n es la cantidad de datos que componen el vector de entrada.

Las entradas al sistema, en el presente proyecto, corresponden a una serie de tiempo, en donde el valor de sus variables representan datos históricos de la cantidad de las captura de anchovetas en el norte de Chile.

3.5 Reglas del modelo

Las reglas del modelo TS constituyen el conocimiento que posee el modelo con respecto al problema en análisis. Las reglas sirven para calcular el fitness que posee el vector de entrada.

Cada regla del modelo está compuesta por un antecedente y un consecuente, de la forma IF “antecedente” THEN “consecuente”, donde el antecedente se representa por

$$A_i(x) = \prod_{j=1}^n \mu_{ij}(x) \quad (3.2)$$

y el consecuente se representa por y_i . En (6.2), el valor de n corresponde a la cantidad de elementos que componen el vector de entrada [30].

Las reglas R_i son del tipo:

$$\begin{aligned} R_1 : & \text{ IF } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{12} \text{ and } \dots \text{ and } x_n \text{ is } A_{1n} \text{ THEN } y_1 = f_1(x) \\ R_2 : & \text{ IF } x_1 \text{ is } A_{21} \text{ and } x_2 \text{ is } A_{22} \text{ and } \dots \text{ and } x_n \text{ is } A_{2n} \text{ THEN } y_2 = f_2(x) \\ & \dots \\ R_r : & \text{ IF } x_1 \text{ is } A_{r1} \text{ and } x_2 \text{ is } A_{r2} \text{ and } \dots \text{ and } x_n \text{ is } A_{rn} \text{ THEN } y_r = f_r(x) \end{aligned} \quad (3.3)$$

Donde r es la cantidad de reglas del modelo. A_{ij} representa una partición difusa, donde i representa el número del conjunto difuso y j es el número de la partición difusa.

El y_i del modelo TS es el resultado de la i -ésima regla del modelo y corresponde a una combinación lineal de las variables de entrada, definida como:

$$y_i = f_i(x) = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \quad (3.4)$$

La variable i toma valores desde 1 hasta r , donde r es la cantidad de reglas del modelo.

3.6 Conjuntos difusos del modelo

Los conjuntos difusos forman parte del antecedente de una regla del modelo TS y se definen como un conjunto de términos lingüísticos A_i (6.5) en el dominio de las variables de entrada X , en otras palabras, cada conjunto difuso está formado por 'n' particiones, donde n es equivalente con la cantidad de valores que componen el vector de entrada.

$$A_i = \{A_{i1}, A_{i2}, \dots, A_{in}\} \quad (3.5)$$

Cada A_{ij} representa una partición difusa, donde i representa el número del conjunto difuso y j es el número de la partición difusa.

La componente principal de los conjuntos difusos es la función de membresía, la cual indica el grado de pertenencia que tiene el valor de una variable de entrada con respecto a un conjunto difuso. El grado de pertenencia es un valor real que pertenece al intervalo $[0;1]$.

La función de membresía del presente proyecto es una función de tipo gaussiana, expresada como:

$$\exp\left(\frac{x - \mu}{\sigma^2}\right) \quad (3.6)$$

Donde μ es el valor de la media y σ es el valor de la desviación estándar.

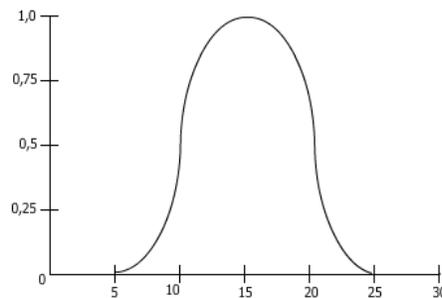


Figura 3.1: Función de membresía de tipo gaussiana.

3.7 Salida del modelo

Como de explico anteriormente, la salida del modelo esta representada por Y (6.7), la cual corresponde a la suma de la contribución que entrega cada una de las reglas [30] del modelo, es decir, para la i -ésima entrada de x_i , la salida total del modelo es

$$Y = \sum_{i=1}^r w_i y_i \quad (3.7)$$

Donde r es el número de reglas y w_i es el grado normalizado del aporte de la regla R_i , expresado por:

$$w_i = \frac{A_i(x)}{\sum_{j=1}^r A_j(x)} \quad (3.8)$$

A continuación se muestra un diagrama explicando el flujo de TS en su etapa forward, indicando en que etapa está ubicada cada una de las ecuaciones mencionadas anteriormente.

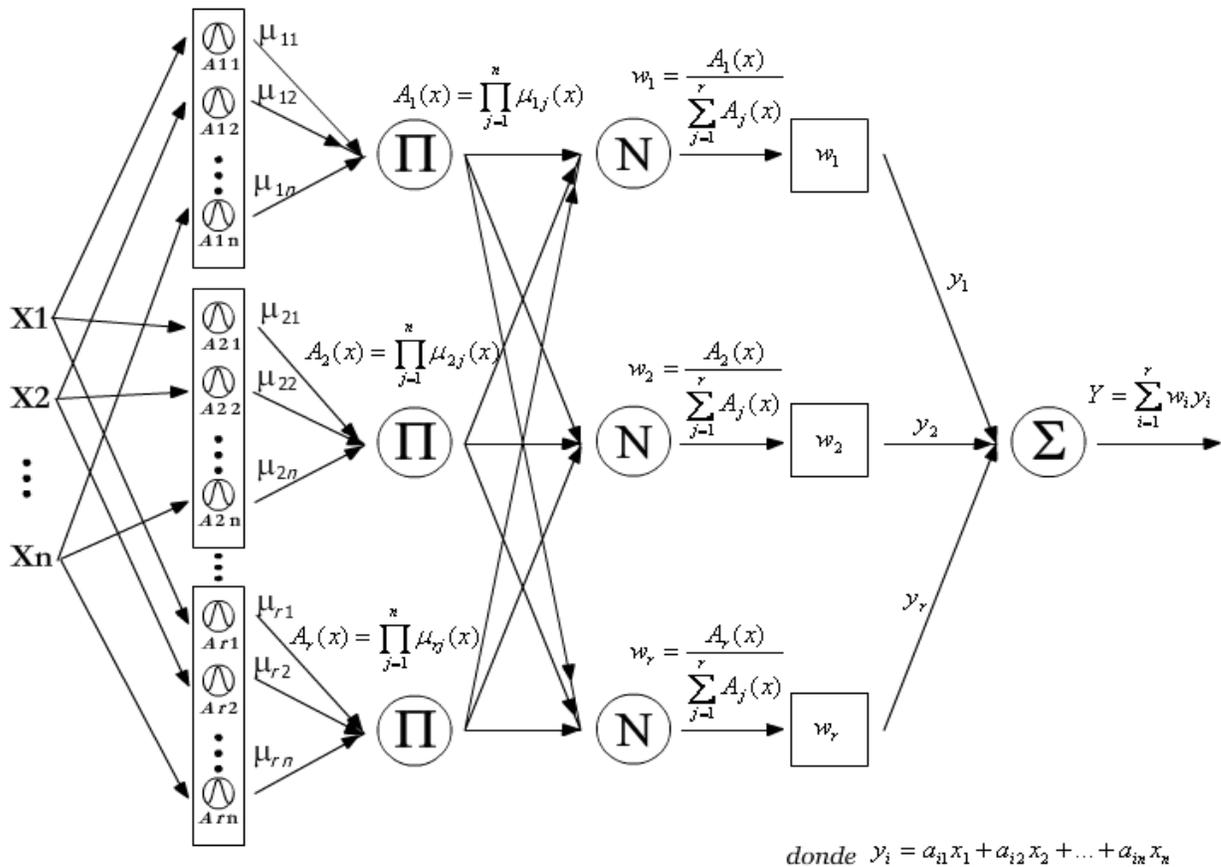


Figura 3.2: Diagrama del flujo del modelo TS en su etapa forward.

Simulación y evaluación de rendimiento

4.1 Introducción

En el presente capítulo se mostrarán los resultados obtenidos en la etapa de testing del modelo con las distintas topologías propuestas, las cuales serán evaluadas con métricas que posteriormente serán definidas. Al inicio de cada prueba se indicará la configuración del sistema, se mostrarán los resultados obtenidos en una tabla de información y posteriormente se realizará un análisis de los resultados de la prueba.

Para concluir el capítulo se hará un análisis de los resultados obtenidos de todas las pruebas realizadas y se entregará la mejor topología encontrada en base a las pruebas realizadas.

4.2 Organización del texto

Para comenzar, en el punto 4.3 se presentan las métricas de evaluación de resultados que se aplicarán al modelo. El punto 4.4 presenta el plan de pruebas para evaluar el modelo. El punto 4.5 presenta las pruebas realizadas al modelo tradicional de PSO, junto con gráficos para el peor resultado y el mejor resultado de los 10 randoms realizados en cada prueba. El punto 4.6 presenta el análisis de resultados obtenidos con PSO tradicional. El punto 4.7 presenta las pruebas realizadas al modelo Quantum de PSO, junto con gráficos para el peor resultado y el mejor resultado de los 10 randoms realizados en cada prueba. Para finalizar el capítulo, en el punto 4.8 se muestran los resultados de las pruebas para encontrar los valores óptimos para las variables de tamaño del enjambre, cantidad de épocas, peso de inercia w , C_1 y C_2 que corresponden a las variables de control de la influencia de las componentes cognitiva y social.

4.3 Métricas de evaluación

Para la evaluación del modelo se utilizarán 3 métricas:

$$\text{Coeficiente de determinación, } R^2 = 1 - \frac{\sum_{i=1}^N (Q_t - \hat{Q}_t)^2}{\sum_{i=1}^N (Q_t - \bar{Q}_t)^2} \quad (4.1)$$

$$\text{Error cuadrático medio, } \text{ECM} = \frac{\sum_{i=1}^N (Q_t - \hat{Q}_t)^2}{N} \quad (4.2)$$

$$\text{MAE, } \text{MAE} = \frac{\sum_{i=1}^N |Q_t - \hat{Q}_t|}{N} \quad (4.3)$$

Donde Q_t representa las capturas observadas en un tiempo t , \hat{Q}_t representa las capturas estimadas en un tiempo t y \bar{Q}_t representa la media de las capturas observadas.

4.4 Plan de pruebas

Para ver los resultados del modelo, se evaluarán las distintas topologías de PSO Tradicional y PSO Quantum, con la data de testing, la cual contiene la cantidad mensual de toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] desde el año 2002 hasta el año 2007.

El plan de pruebas consiste en evaluar por separado cada variable del sistema para determinar su importancia y en que medida influye dentro del sistema. Las pruebas que a continuación se muestran pretenden ajustar la carga del sistema traducido en la cantidad de dimensiones de las partículas del enjambre, la cual esta determinada por la cantidad de entradas del modelo y la cantidad de reglas.

Mientras menor sea la cantidad de dimensiones del modelo, menor será la carga del sistema. El plan de pruebas se detalla a continuación

Cantidad de entradas	Cantidad de reglas
2	1
2	2
2	3
4	1
4	2
4	3
8	1
8	2
8	3

Tabla 4.1: Plan de pruebas del sistema para determinar la cantidad de entradas y la cantidad de reglas del modelo.

Se analizarán configuraciones de 2, 4 y 8 entradas, donde para cada una de ellas también se analizará la cantidad de 1, 2 y 3 reglas. Con esto se espera obtener la información necesaria para poder determinar la mejor configuración del sistema.

4.5 Pruebas con PSO tradicional

Como ya se estudio en el capítulo 2, la función de restricción de velocidad máxima para la velocidad de una partícula esta expresada por

$$\begin{aligned} & \text{if } (v_{id} > V_{\max}) \text{ or } (v_{id} < -V_{\max}) \\ & v_{id} = \text{sign}(v_{id}) \cdot V_{\max} \end{aligned} \quad (4.4)$$

Sin embargo con esta función, muchos de los resultados entregados por el PSO eran de tipo Nan, lo cual indica que es un resultado imposible de calcular. Para ello se realizó una modificación en la función, que finalmente quedó definida como

$$\begin{aligned} & \text{if } (v_{id} > V_{\max}) \text{ or } (v_{id} < -V_{\max}) \\ & v_{id} = \text{sign}(v_{id}) \cdot \bar{v} \end{aligned} \quad (4.5)$$

donde el valor de \bar{v} se define como la suma ponderada del valor absoluto de las velocidades de las dimensiones de cada una de las partículas del enjambre, expresada por

$$\bar{v} = \frac{1}{M \cdot D} \sum_{i=1}^M \sum_{d=1}^D (|v_{id}|) \quad (4.6)$$

con la finalidad de que la nueva velocidad de la dimensión de la partícula quede más cercana al resto de las velocidades del enjambre.

4.5.1 Topología de 2 entradas

	R^2	ECM	MAE
Random 1	0.75943303	0.00353094	0.04796981
Random 2	0.75951571	0.00352972	0.04796972
Random 3	0.75943851	0.00353086	0.04796987
Random 4	0.75951626	0.00352972	0.04796972
Random 5	0.75939644	0.00353147	0.04796984
Random 6	0.75951595	0.00352972	0.04796973
Random 7	0.75951646	0.00352971	0.04796972
Random 8	0.75947777	0.00353028	0.04796963
Random 9	0.75950820	0.00352983	0.04796972
Random 10	0.75951781	0.00352969	0.04796971

Tabla 4.2: Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas y 1 regla de PSO tradicional con 10 partículas.

La tabla 4.2 corresponde a los resultados obtenidos de las pruebas con una configuración de 2 entradas, 1 regla y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba son muy homogéneos en cada uno de los randoms, se aprecia que hay estabilidad de resultados con esta configuración, sin embargo un valor de R^2 de 75,95% es bajo para ser considerado una buena estimación para problema propuesto.

Figura 4.1: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.2 obtenido en el Random 5, con un R^2 de 75,93%.

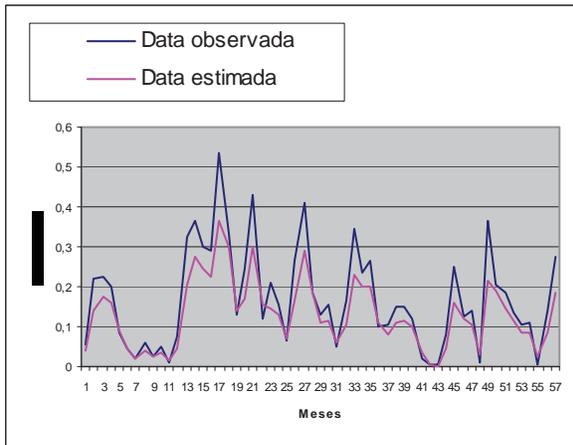
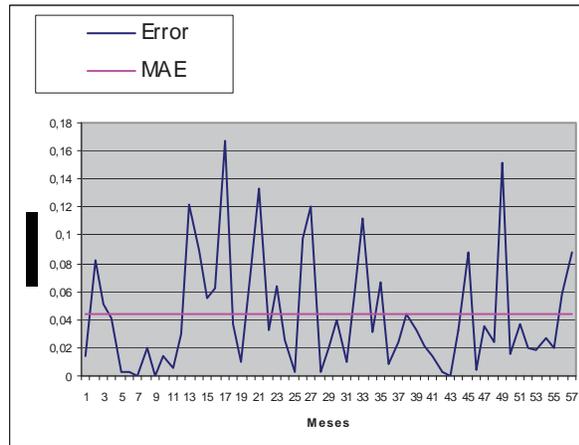


Figura 4.2: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 75,93%.



En las figuras 4.1 y 4.2 correspondientes al Random 5, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre $[0;1]$ con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.3: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.2 obtenido en el Random 10, con un R^2 de 75,95%.

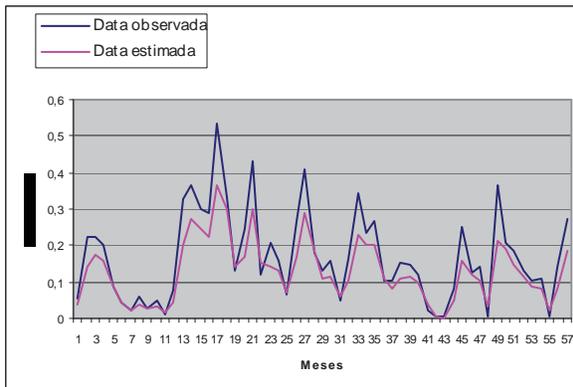
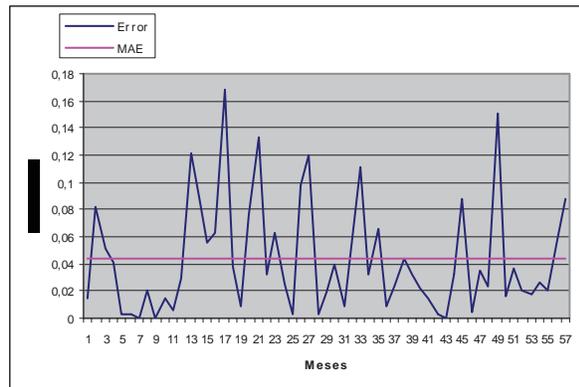


Figura 4.4: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 10, con un R^2 de 75,95%.



En las figuras 4.3 y 4.4 correspondientes al Random 10, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre $[0;1]$ con la data de testing desde el año 2002 hasta el año 2007.

	R^2	ECM	MAE
Random 1	0.79783997	0.00296722	0.08657784
Random 2	0.78482159	0.00315829	0.09879318
Random 3	0.81296836	0.00274517	0.08584466
Random 4	0.81255112	0.00275129	0.08680451
Random 5	0.76979226	0.00337889	0.09142557
Random 6	0.75166210	0.00364500	0.08673017
Random 7	0.77239404	0.00334070	0.08804747
Random 8	0.71456750	0.00418946	0.08700603
Random 9	0.79882008	0.00295283	0.08684461
Random 10	0.78595391	0.00314168	0.08684234

Tabla 4.3: Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas y 2 regla de PSO tradicional con 10 partículas.

La tabla 4.3 corresponde a los resultados obtenidos de las pruebas con una configuración de 2 entradas, 2 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba son bastante homogéneos, se aprecia que los resultados oscilan entre un valor de R^2 de 71,45% y 81,29%, lo cual indica que el modelo está entregando resultados considerablemente buenos, sin embargo el modelo puede seguir mejorándose.

Figura 4.5: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.3 obtenido en el Random 8, con un R^2 de 71,45%.

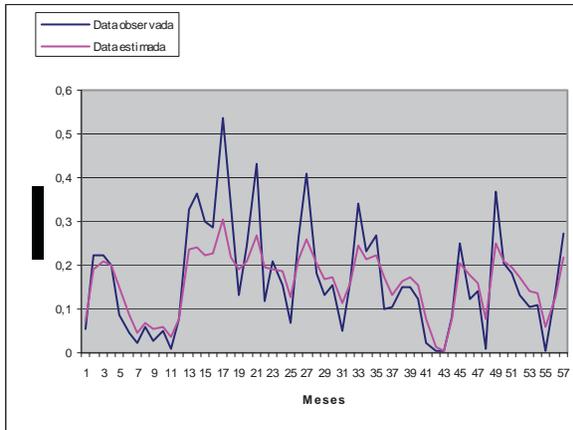
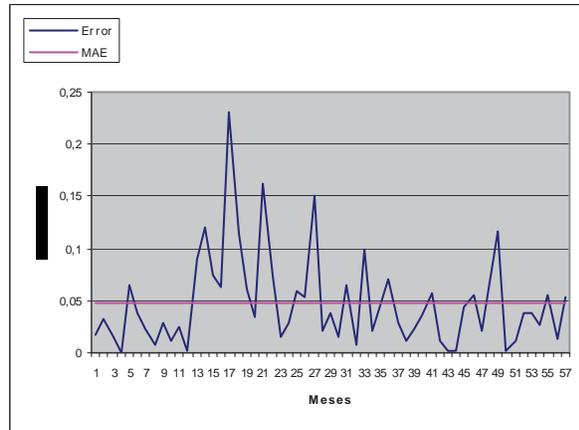


Figura 4.6: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 8, con un R^2 de 71,45%.



En las figuras 4.5 y 4.6 correspondientes al Random 8, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.7: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.3 obtenido en el Random 3, con un R^2 de 81,29%.

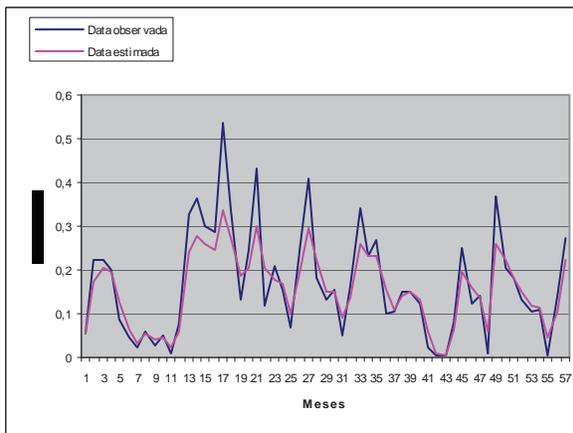
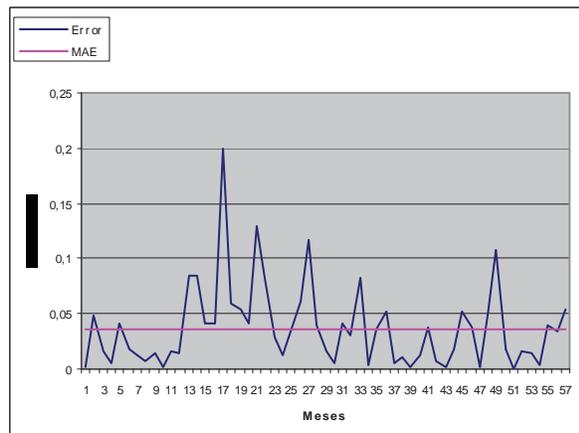


Figura 4.8: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 81,29%.



En las figuras 4.7 y 4.8 correspondientes al Random 3, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

	R^2	ECM	MAE
Random 1	0.84264767	0.00230955	0.03740204
Random 2	0.78145107	0.00320777	0.04744038
Random 3	0.71762400	0.00414459	0.04819477
Random 4	0.67662819	0.00474631	0.05802577
Random 5	0.80769744	0.00282253	0.03670957
Random 6	0.81566122	0.00270564	0.03710657
Random 7	0.72283021	0.00406818	0.04696495
Random 8	0.79022934	0.00307892	0.04726425
Random 9	0.65965112	0.00499550	0.05899611
Random 10	0.70235433	0.00436872	0.05697569

Tabla 4.4: Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas y 3 reglas.

La tabla 4.4 corresponde a los resultados obtenidos de las pruebas con una configuración de 2 entradas, 3 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba no son muy homogéneos, el sistema entrega resultados para R^2 en un rango de 65,96% y 84,26%, lo cual indica que los resultados de la configuración no son muy homogéneos, sin embargo el valor de R^2 de 84,26% es bastante bueno considerando la poca carga del sistema.

Figura 4.9: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.4 obtenido en el Random 9, con un R^2 de 65,96%.

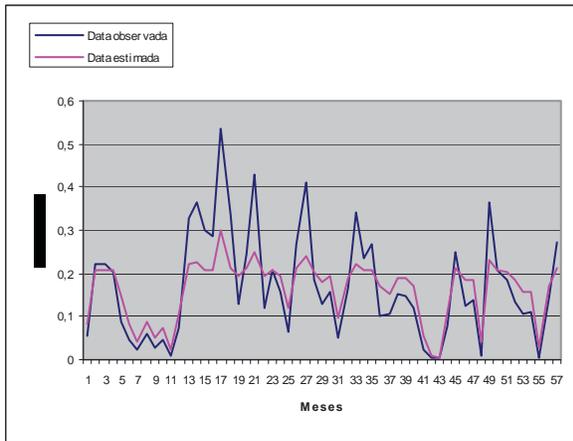
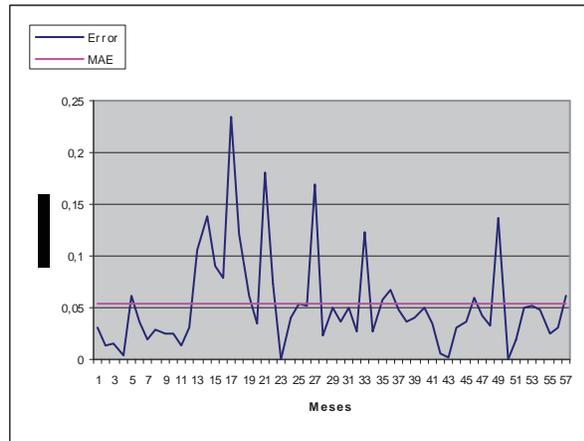


Figura 4.10: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 65,96%.



En las figuras 4.9 y 4.10 correspondientes al Random 9, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre $[0;1]$ con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.11: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.4 obtenido en el Random 1, con un R^2 de 84,26%.

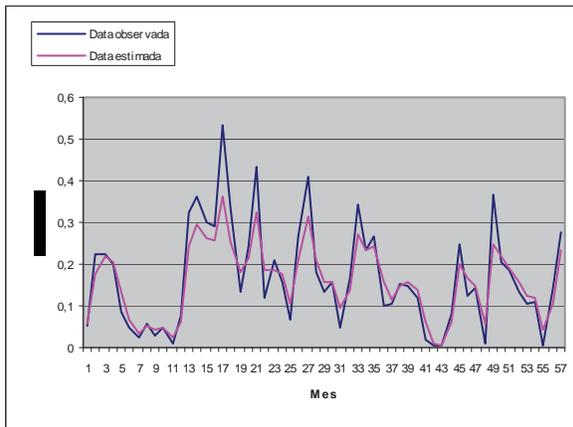
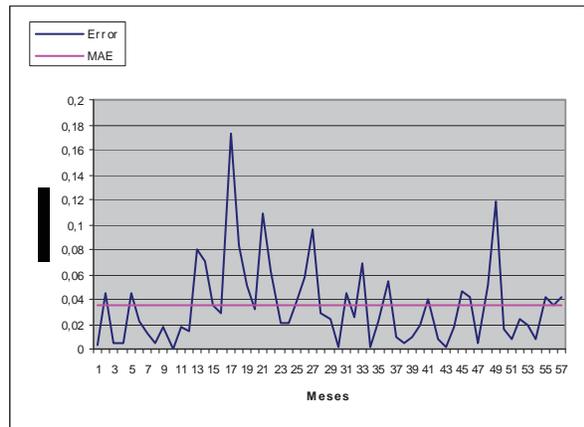


Figura 4.12: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 1, con un R^2 de 84,26%.



En las figuras 4.11 y 4.12 correspondientes al Random 1, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre $[0;1]$ con la data de testing desde el año 2002 hasta el año 2007.

4.5.2 Topología de 4 entradas

	R^2	ECM	MAE
Random 1	0.68951376	0.00463393	0.08798695
Random 2	0.68856830	0.00464804	0.08797608
Random 3	0.68957150	0.00463307	0.08798570
Random 4	0.69057224	0.00461813	0.08797442
Random 5	0.68988975	0.00462832	0.08799820
Random 6	0.68889328	0.00464319	0.08797341
Random 7	0.68999594	0.00462673	0.08797101
Random 8	0.68917721	0.00463895	0.08797509
Random 9	0.68876367	0.00464513	0.08800092
Random 10	0.68748595	0.00466420	0.08798922

Tabla 4.5: Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 1 regla y 10 partículas.

La tabla 4.5 corresponde a los resultados obtenidos de las pruebas con una configuración de 4 entradas, 1 regla y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba son muy homogéneos, pero el desempeño del sistema es muy bajo considerando que su mejor valor en la bondad de ajuste de R^2 fue de 68,99%.

Figura 4.13: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.5 obtenido en el Random 10, con R^2 de 68,74%.

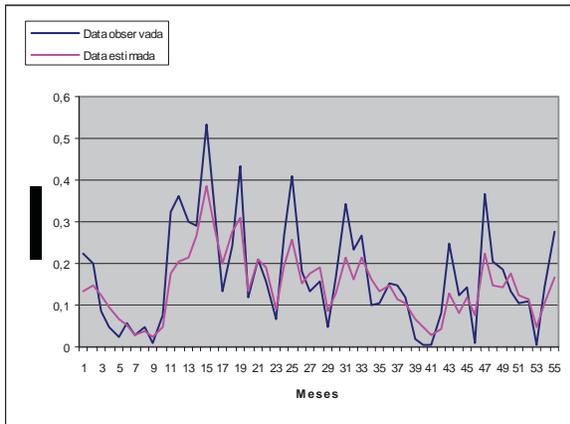
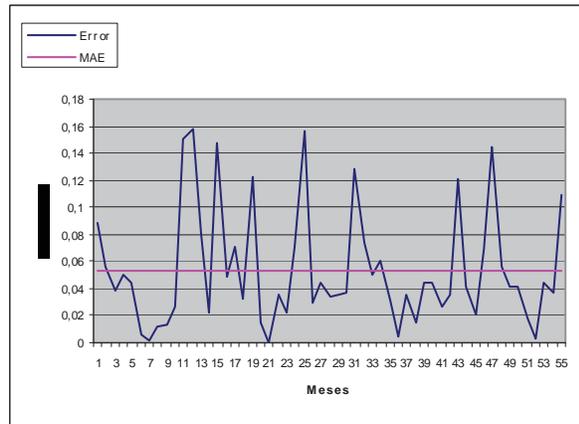


Figura 4.14: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 10, con un R^2 de 68,74%.



En las figuras 4.13 y 4.14 correspondientes al Random 10, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.15: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.5 obtenido en el Random 4, con un R^2 de 69,05%.

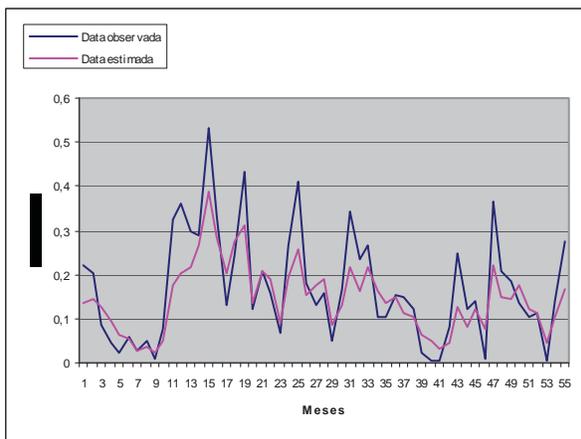
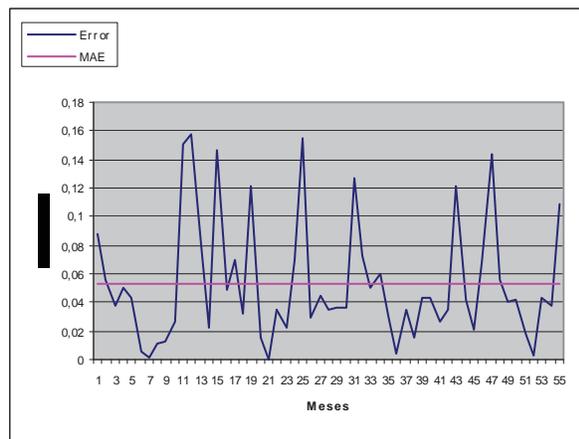


Figura 4.16: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 4, con un R^2 de 69,05%.



En las figuras 4.15 y 4.16 correspondientes al Random 4, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

	R^2	ECM	MAE
Random 1	0.68862900	0.00464714	0.08795552
Random 2	0.68937538	0.00463600	0.08796910
Random 3	0.68408518	0.00471495	0.08837587
Random 4	0.68831232	0.00465186	0.08798445
Random 5	0.68899344	0.00464170	0.08799250
Random 6	0.68909383	0.00464020	0.08802021
Random 7	0.70785174	0.00436024	0.04952204
Random 8	0.68915738	0.00463925	0.08798840
Random 9	0.68827449	0.00465243	0.08799646
Random 10	0.68917896	0.00463893	0.08798044

Tabla 4.6: Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 2 reglas y 10 partículas.

La tabla 4.6 corresponde a los resultados obtenidos de las pruebas con una configuración de 4 entradas, 2 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba son muy homogéneos, pero el desempeño del sistema es muy bajo considerando que su mejor valor en la bondad de ajuste de R^2 fue de 70,78%.

Figura 4.17: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.6 obtenido en el Random 3, con R^2 de 68,40%.

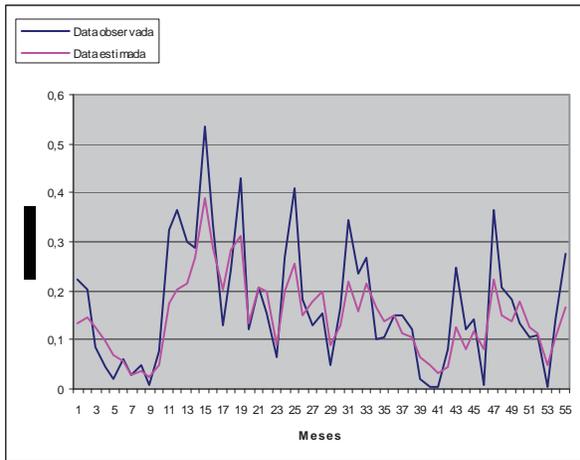
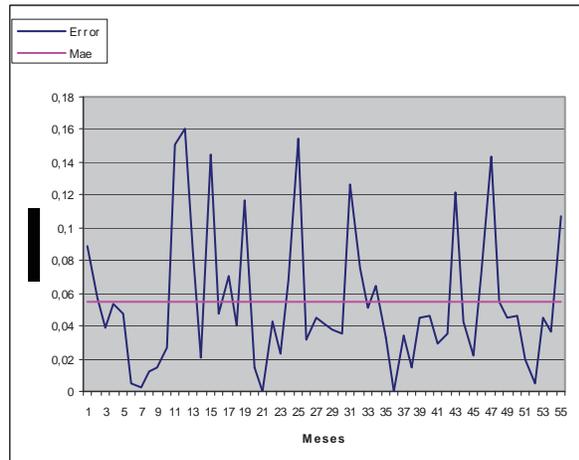


Figura 4.18: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 68,40%.



En las figuras 4.17 y 4.18 correspondientes al Random 3, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.19: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.6 obtenido en el Random 7, con un R^2 de 70,78%.

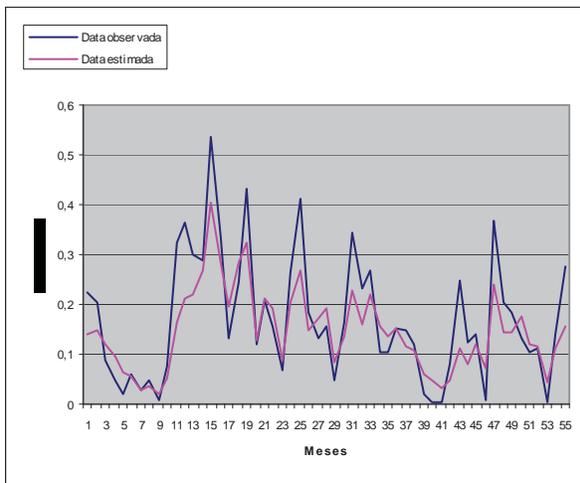
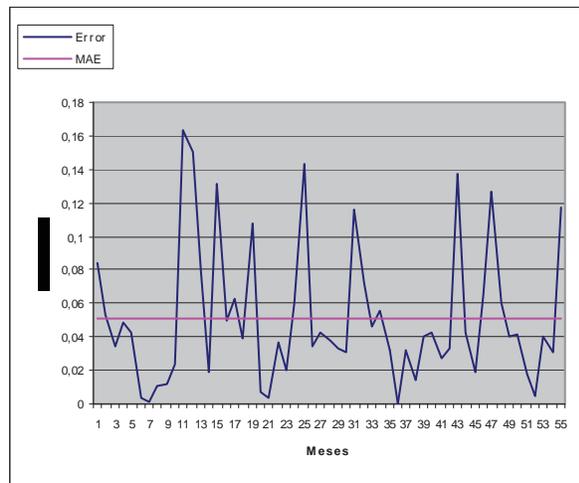


Figura 4.20: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 7, con un R^2 de 70,78%.



En las figuras 4.19 y 4.20 correspondientes al Random 7, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

	R^2	ECM	MAE
Random 1	0.64765864	0.00525861	0.05809457
Random 2	0.63496283	0.00544809	0.05873848
Random 3	0.68932017	0.00463682	0.05599743
Random 4	0.70548077	0.00439563	0.05407502
Random 5	0.70817874	0.00435536	0.05490503
Random 6	0.75194333	0.00370218	0.04567875
Random 7	0.68927238	0.00463753	0.05598404
Random 8	0.68920039	0.00463861	0.05597911
Random 9	0.68891794	0.00464282	0.05598295
Random 10	0.68982247	0.00462932	0.05597228

Tabla 4.7: Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 3 reglas y 10 partículas.

La tabla 4.7 corresponde a los resultados obtenidos de las pruebas con una configuración de 4 entradas, 3 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba no son óptimos, ya que el mejor valor obtenido para la bondad de ajuste de R^2 fue de 77,92%.

Figura 4.21: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.7 obtenido en el Random 2, con R^2 de 63,49%.

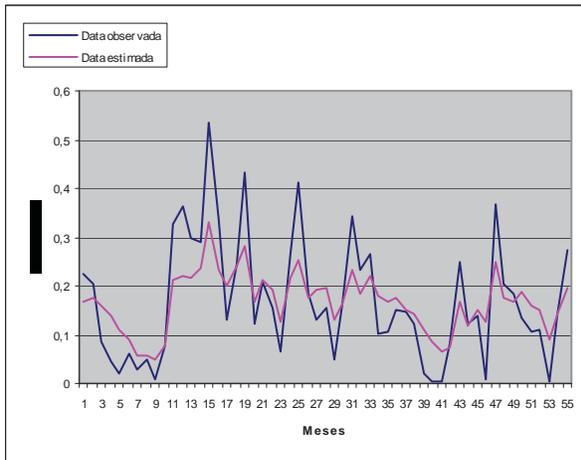
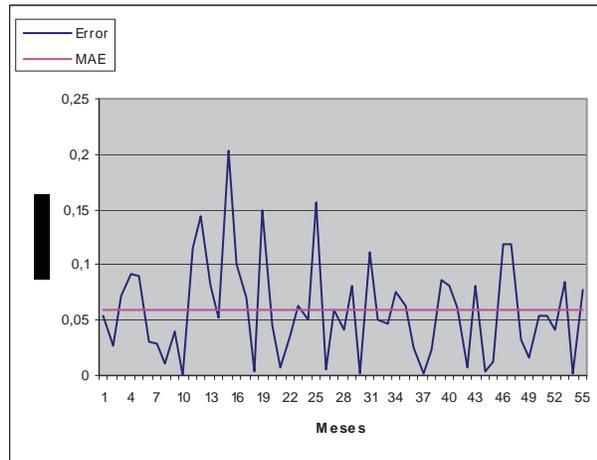


Figura 4.22: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 2, con un R^2 de 63,49%.



En las figuras 4.21 y 4.22 correspondientes al Random 2, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.23: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.7 obtenido en el Random 6, con un R^2 de 75,19%.

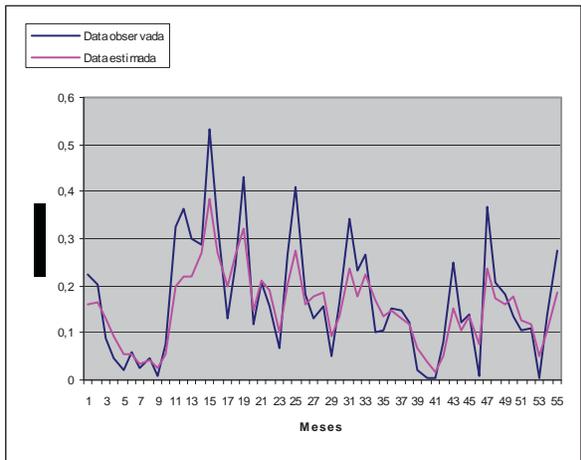
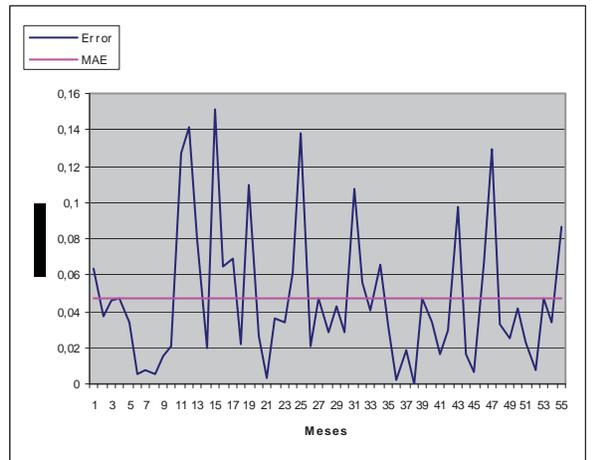


Figura 4.24: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 6, con un R^2 de 75,19%.



En las figuras 4.23 y 4.24 correspondientes al Random 6, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

4.5.3 Topología de 8 entradas

	R^2	ECM	MAE
Random 1	0.64593194	0.00551773	0.08773318
Random 2	0.64725013	0.00549719	0.08778182
Random 3	0.65041468	0.00544787	0.08766213
Random 4	0.65101241	0.00543856	0.08767711
Random 5	0.65618271	0.00535799	0.08771028
Random 6	0.64417343	0.00554514	0.08773651
Random 7	0.65032399	0.00544929	0.08765185
Random 8	0.65417513	0.00538927	0.08777292
Random 9	0.65303566	0.00540703	0.08768150
Random 10	0.65011125	0.00545260	0.08769851

Tabla 4.8: Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 1 regla y 10 partículas.

La tabla 4.8 corresponde a los resultados obtenidos de las pruebas con una configuración de 8 entradas, 1 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba no son buenos, ya que el mejor valor obtenido para la bondad de ajuste de R^2 fue de 65,61%.

Al parecer la configuración de 8 entradas no entrega buenos resultados y al compararla con los resultados de las demás configuraciones, se podrá determinar si esta configuración sirve o no, pero todo indica que con esta configuración el rendimiento del sistema será bajo.

Figura 4.25: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.8 obtenido en el Random 6, con R^2 de 64,41%.

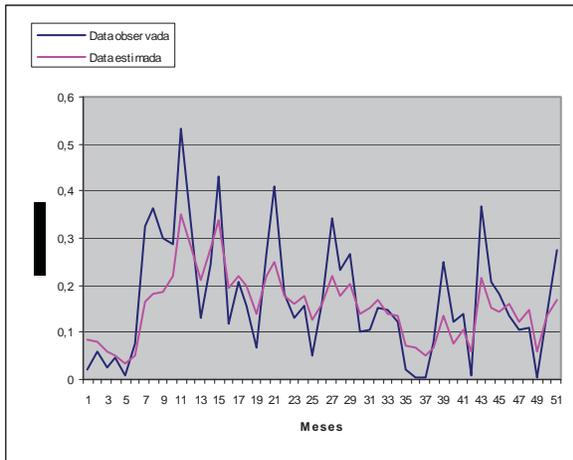
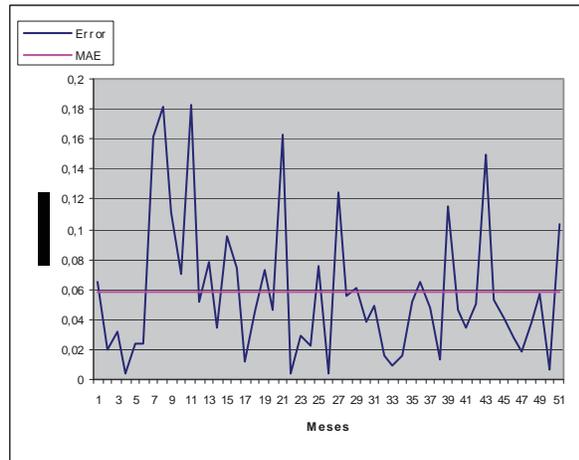


Figura 4.26: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 6, con un R^2 de 64,41%.



En las figuras 4.25 y 4.26 correspondientes al Random 6, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.27: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.8 obtenido en el Random 5, con un R^2 de 65,61%.

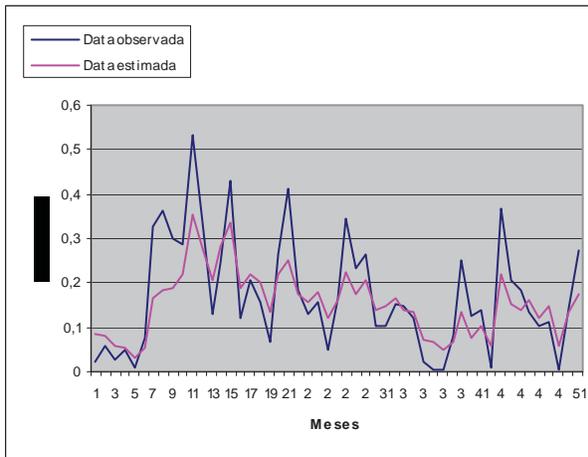
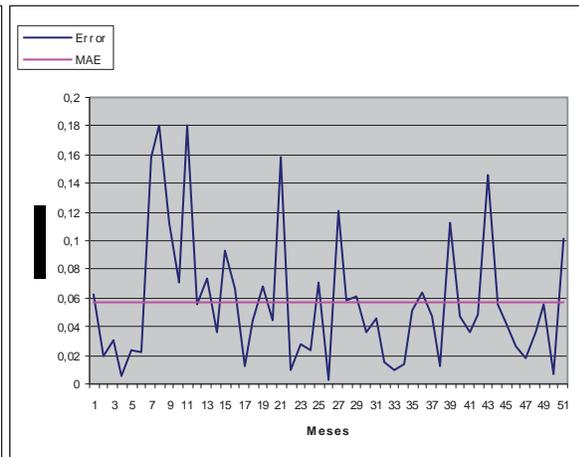


Figura 4.28: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 65,61%.



En las figuras 4.27 y 4.28 correspondientes al Random 5, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

	R^2	ECM	MAE
Random 1	0.65024378	0.00545054	0.08767642
Random 2	0.65221059	0.00541989	0.08775301
Random 3	0.65558822	0.00536725	0.08773169
Random 4	0.65010918	0.00545264	0.08761353
Random 5	0.65227552	0.00541888	0.08772572
Random 6	0.64733521	0.00549586	0.08768527
Random 7	0.64922739	0.00546638	0.08776241
Random 8	0.64912063	0.00546804	0.08767863
Random 9	0.64792595	0.00548666	0.08768948
Random 10	0.64812784	0.00548351	0.08770122

Tabla 4.9: Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 2 reglas y 10 partículas.

La tabla 4.9 corresponde a los resultados obtenidos de las pruebas con una configuración de 8 entradas, 2 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba no son buenos, ya que el mejor valor obtenido para la bondad de ajuste de R^2 fue de 65,55%.

Figura 4.29: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.9 obtenido en el Random 6, con R^2 de 64,73%.

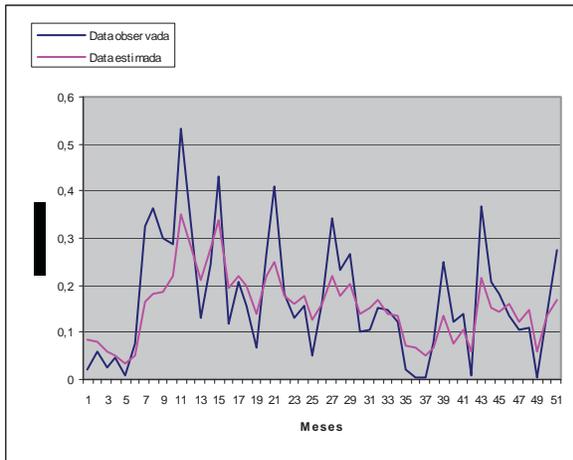
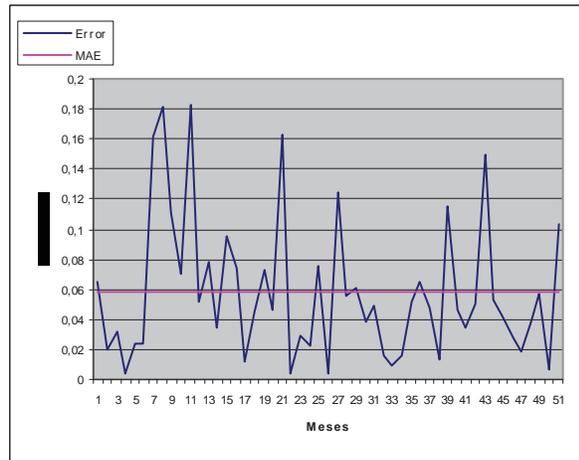


Figura 4.30: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 6, con un R^2 de 64,73%.



En las figuras 4.29 y 4.30 correspondientes al Random 6, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.31: Gráfico de la data estimada v/s la data observada del mejor resultado de la tabla 4.9 obtenido en el Random 3, con un R^2 de 65,55%.

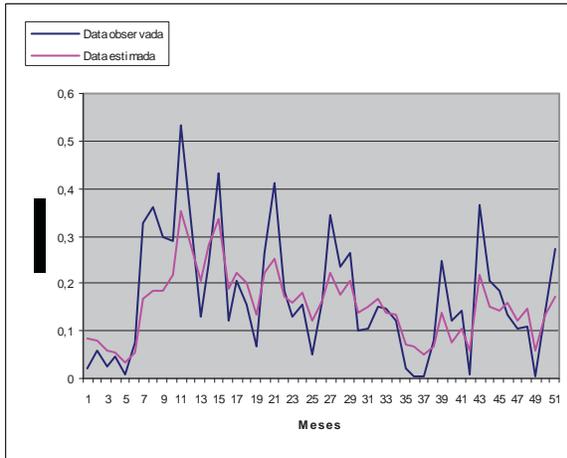
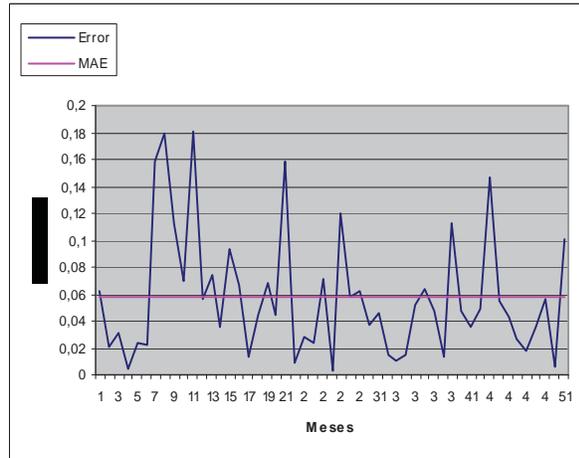


Figura 4.32: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 65,55%.



En las figuras 4.31 y 4.32 correspondientes al Random 3, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

	R^2	ECM	MAE
Random 1	0.64722138	0.00549764	0.05763982
Random 2	0.64514465	0.00553000	0.05735312
Random 3	0.65224075	0.00541942	0.05982782
Random 4	0.64708809	0.00549972	0.05770650
Random 5	0.64965983	0.00545964	0.05670834
Random 6	0.64699995	0.00550109	0.05655906
Random 7	0.56272667	0.00681439	0.06506654
Random 8	0.42797939	0.00891427	0.07342394
Random 9	0.49010747	0.00794608	0.06736959
Random 10	0.54141995	0.00714643	0.06650805

Tabla 4.10: Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 3 reglas y 10 partículas.

La tabla 4.10 corresponde a los resultados obtenidos de las pruebas con una configuración de 8 entradas, 3 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba fueron muy bajos, ya que el mejor valor obtenido para la bondad de ajuste de R^2 fue de 65,22%, pero su valor mas bajo fue de 42,79%

Figura 4.33: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.10 obtenido en el Random 8, con R^2 de 42,79%.

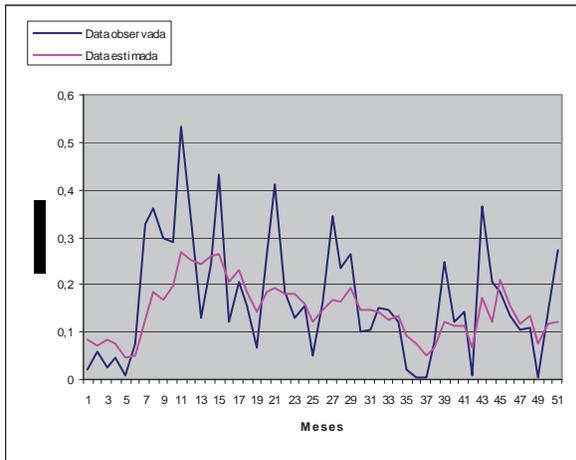
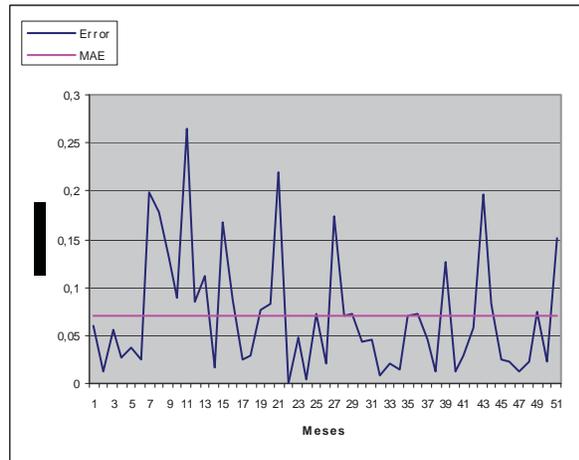


Figura 4.34: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 8, con un R^2 de 42,79%.



En las figuras 4.33 y 4.34 correspondientes al Random 8, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.35: Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 4.10 obtenido en el Random 3, con un R^2 de 65,22%.

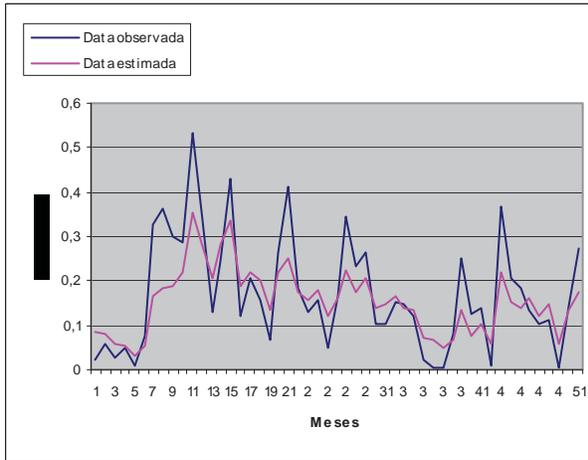
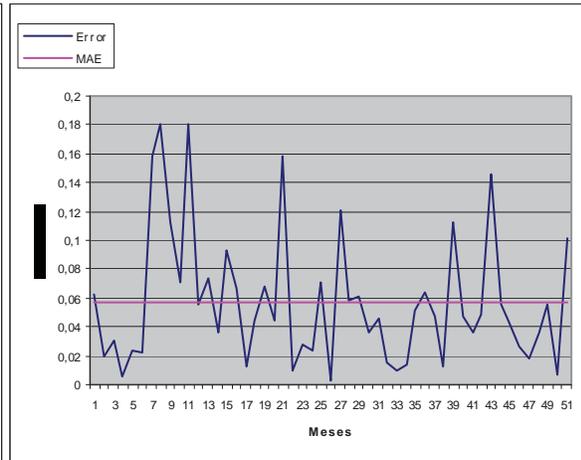


Figura 4.36: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 65,22%.



En las figuras 4.35 y 4.36 correspondientes al Random 3, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

4.6 Análisis de resultados con PSO tradicional

Como se puede apreciar en las pruebas que los mejores resultados obtenidos se obtuvieron con una configuración de 2 entradas y 3 reglas con un 84,26% en la bondad de ajuste de R^2 y que los peores resultados se obtuvieron con la configuración de 8 entradas y 3 reglas de un 42,79% en la bondad del ajuste de R^2 .

Los resultados indican que a menor carga de la partícula, el PSO entrega mejores resultados. A mayor cantidad de entradas el rendimiento del sistema es mas bajo, por lo que se concluye que una configuración de pocas entradas, la cual vendría siendo 2 entradas para este modelo, es la óptima.

Para ver en más detalle lo anterior, se muestran a continuación el peor resultado con su configuración y el mejor resultado con su respectiva configuración.

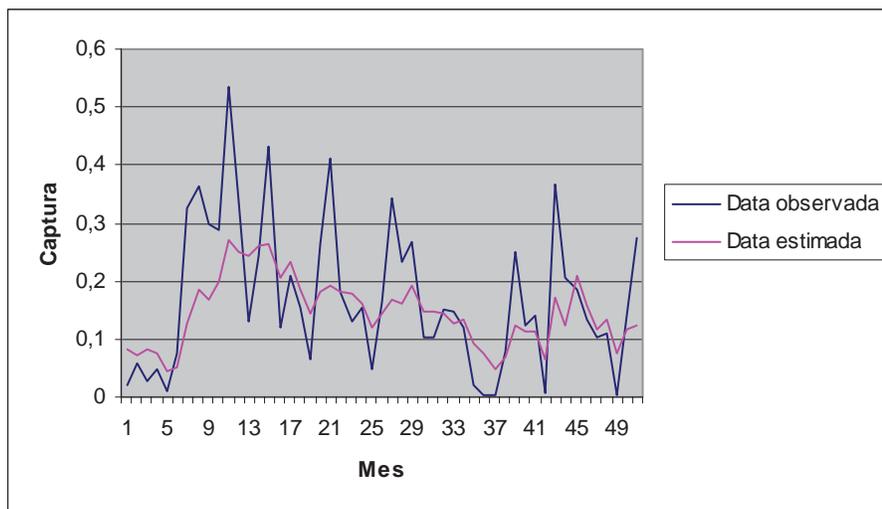


Figura 4.37: Gráfico del peor resultado estimado con el entrenamiento PSO tradicional para la data observada v/s la data estimada correspondiente a la data de testing desde el año 2002 hasta el año 2007.

La figura 4.37 corresponde al resultado mas bajo entregado por las distintas configuraciones del sistema, la cual se obtuvo con una configuración de 8 entradas y 3 reglas, aplicada a la data de testing que comprende del año 2002 hasta el año 2007 y cuya bondad de ajuste de R^2 obtenido fue de 42,79%. La coordenada x corresponde al los meses de la data de testing y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1]. El grafico demuestra que un número alto de entradas al sistema no entrega buenos resultados.

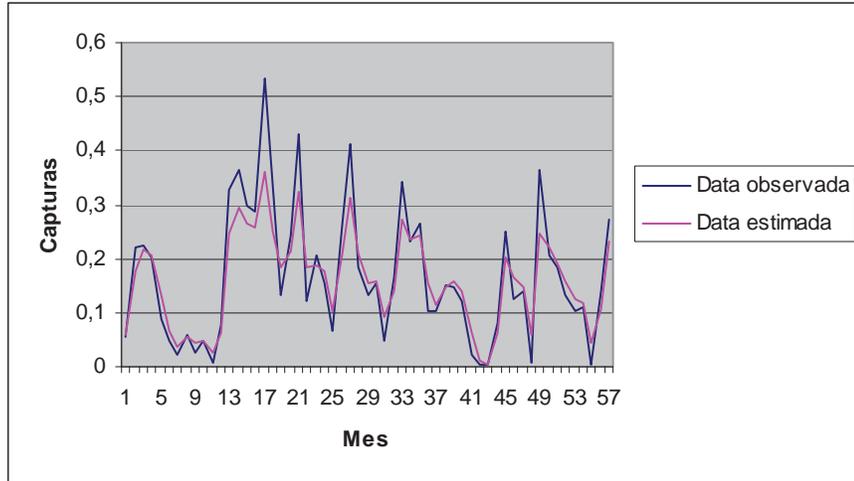


Figura 4.38: Gráfico del mejor resultado estimado con el entrenamiento PSO tradicional para la data observada v/s la data estimada correspondiente a la data de testing desde el año 2002 hasta el año 2007

La figura 4.38 corresponde al mejor resultado entregado por las distintas configuraciones del sistema, la cual se obtuvo con una configuración de 2 entradas y 3 reglas, aplicada a la data de testing que comprende del año 2002 hasta el año 2007 y cuya bondad de ajuste de R^2 obtenido fue de 84,26%. La coordenada x corresponde a los meses de la data de testing y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1]. El gráfico demuestra que un número bajo de entradas al sistema entrega mejores resultados.

Por lo anterior se concluye que la mejor configuración se obtiene con una cantidad baja de entradas, la cual es de un tamaño de 2 y con 3 reglas.

4.7 Pruebas con PSO Quantum

A continuación se mostrarán los resultados obtenidos con PSO Quantum para determinar si esta variante de PSO entrega mejores resultados que PSO tradicional.

La cantidad de pruebas para PSO Quantum son menos que para el modelo tradicional, debido a que todas las variables pertenecientes a la función de velocidad del modelo tradicional no existen en PSO Quantum.

El plan de pruebas es el siguiente:

Cantidad de entradas	Cantidad de reglas
2	3
4	3
8	3

Tabla 4.11: Plan de pruebas del sistema para determinar la efectividad de PSO Quantum.

4.7.1 Topología de 2 entradas

	R^2	ECM	MAE
Random 1	0.81573604	0.00270455	0.03647875
Random 2	0.79029566	0.00307795	0.03924349
Random 3	0.73082543	0.00395083	0.04514792
Random 4	0.83393522	0.00243743	0.03579925
Random 5	0.88904366	0.00162857	0.02708696
Random 6	0.83921602	0.00235992	0.03072854
Random 7	0.82654861	0.00254584	0.03625514
Random 8	0.76925994	0.00338670	0.04330791
Random 9	0.59842853	0.00589410	0.05410232
Random 10	0.84272203	0.00230846	0.03404936

Tabla 4.12: Evaluación de R^2 , ECM y MAE, para configuración de 2 entradas, 3 reglas y 10 partículas.

La tabla 4.12 corresponde a los resultados obtenidos de las pruebas con una configuración de 2 entradas, 3 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba demuestran que PSO Quantum entrega mejores resultados que PSO tradicional, considerando además que el resultado de R^2 obtenido fue muy bueno con un valor de 88,9% en la bondad de ajuste del modelo.

Figura 4.39: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.12 obtenido en el Random 9, con R^2 de 59,84%.

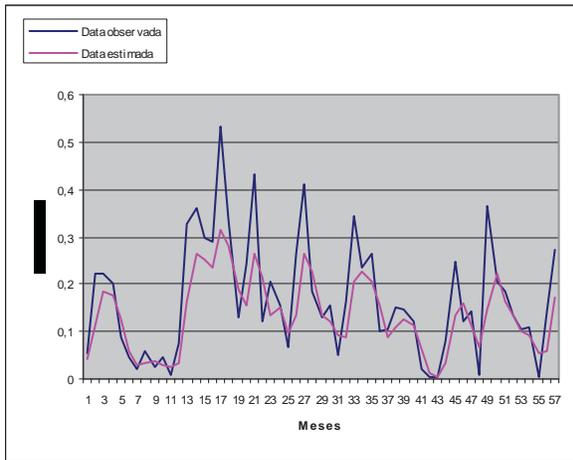
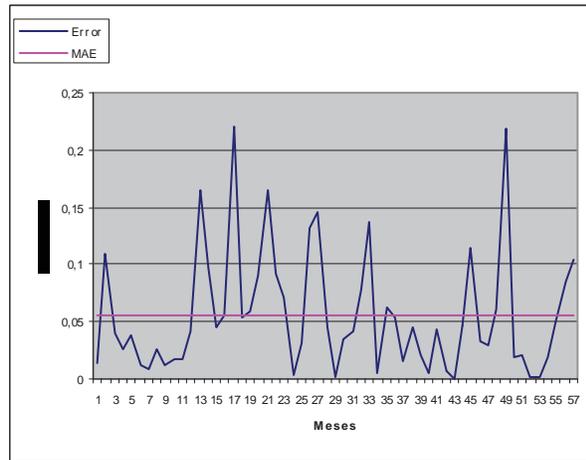


Figura 4.40: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 59,84%.



En las figuras 4.39 y 4.40 correspondientes al Random 9, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.41: Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 4.12 obtenido en el Random 5, con un R^2 de 88,9%.

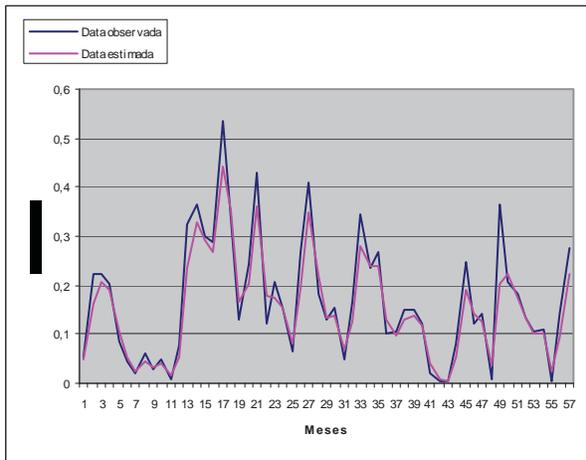
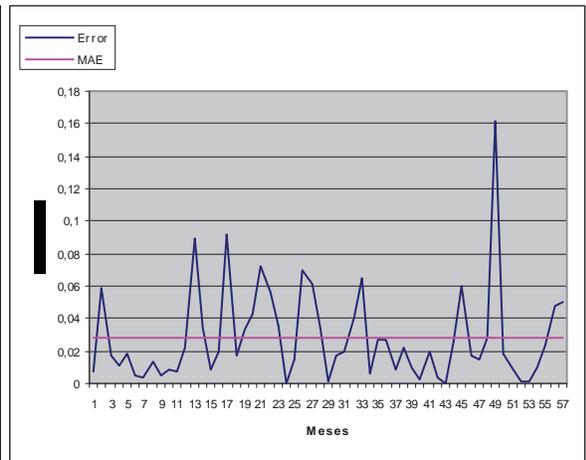


Figura 4.42: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 88,9%.



En las figuras 4.41 y 4.42 correspondientes al Random 5, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

4.7.2 Topología de 4 entradas

	R^2	ECM	MAE
Random 1	0.63032468	0.00551731	0.05477173
Random 2	0.72142176	0.00415771	0.04844609
Random 3	0.56313147	0.00652016	0.06091057
Random 4	0.61280708	0.00577876	0.05911553
Random 5	0.40516351	0.00887779	0.06996189
Random 6	0.62739391	0.00556105	0.05094643
Random 7	0.67853978	0.00479771	0.05243197
Random 8	0.30805907	0.01032705	0.07108375
Random 9	0.72924857	0.00404090	0.05013270
Random 10	0.26351682	0.01099184	0.08423046

Tabla 4.13: Evaluación de R^2 , ECM y MAE, para configuración de 4 entradas, 3 reglas y 10 partículas.

La tabla 4.13 corresponde a los resultados obtenidos de las pruebas con una configuración de 4 entradas, 3 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba demuestran que al igual que PSO tradicional, una configuración de 4 entradas no entrega buenos resultados.

Para apreciarlo con más detalle, se presentan a continuación los gráficos de los resultados obtenidos.

Figura 4.43: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.13 obtenido en el Random 10, con R^2 de 26,35%.

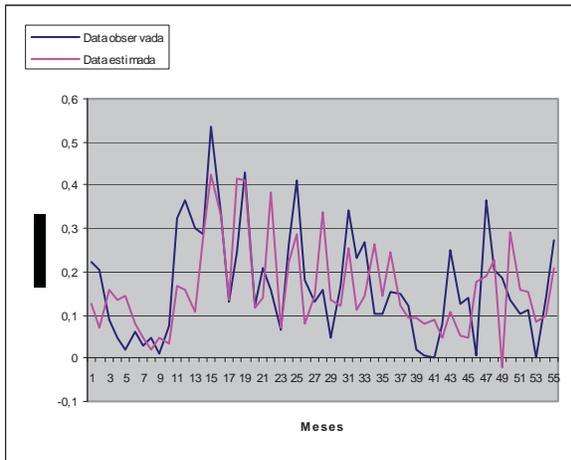
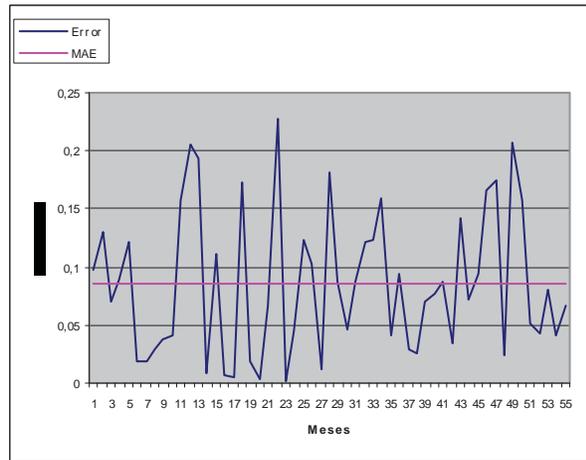


Figura 4.44: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 10, con un R^2 de 26,35%.



En las figuras 4.43 y 4.44 correspondientes al Random 10, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.45: Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 4.13 obtenido en el Random 9, con un R^2 de 72,92%.

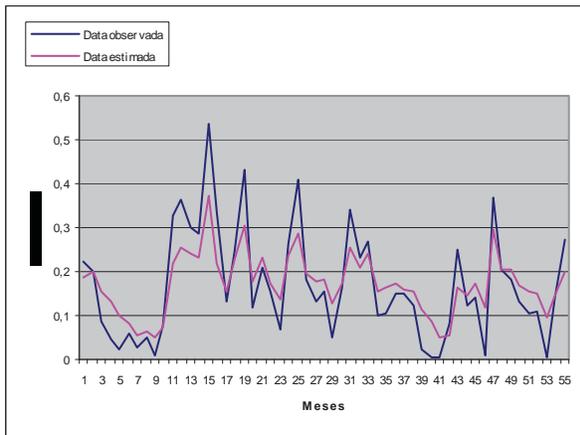
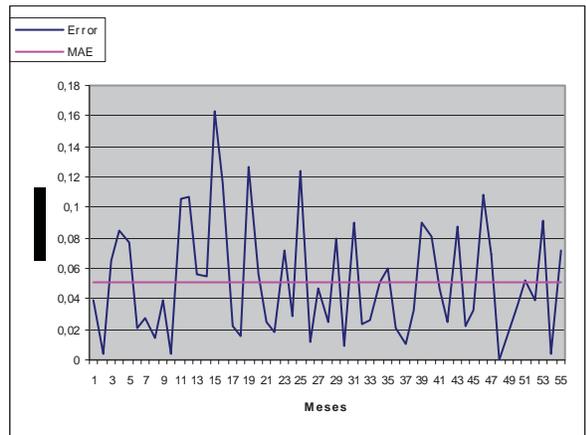


Figura 4.46: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 72,92%.



En las figuras 4.45 y 4.46 correspondientes al Random 9, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

4.7.3 Topología de 8 entradas

	R^2	ECM	MAE
Random 1	0.31460023	0.01068115	0.07202084
Random 2	-0.05597773	0.01645618	0.09596493
Random 3	0.41214429	0.00916104	0.07762396
Random 4	0.55487832	0.00693670	0.06354318
Random 5	-1.18856786	0.03410628	0.13793554
Random 6	-0.03779207	0.01617278	0.10571337
Random 7	-1.05747065	0.03206328	0.13733541
Random 8	0.30113074	0.01089106	0.08367570
Random 9	-0.15803748	0.01804666	0.10836959
Random 10	0.37596310	0.00972488	0.08374327

Tabla 4.14: Evaluación de R^2 , ECM y MAE, para configuración de 8 entradas, 3 reglas y 10 partículas.

La tabla 4.14 corresponde a los resultados obtenidos de las pruebas con una configuración de 8 entradas, 3 reglas y 10 partículas, aplicada a 10 mediciones sobre la data de testing.

Los resultados obtenidos en esta prueba son considerablemente malos, por lo que se concluye que esta configuración no representa una posible solución al sistema.

Figura 4.47: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 4.14 obtenido en el Random 5, con R^2 de -18,85%.

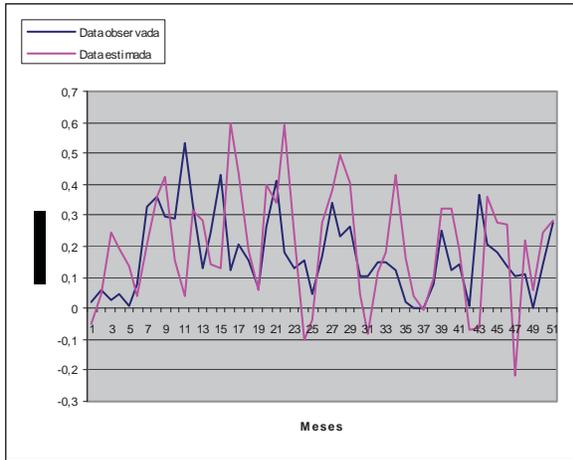
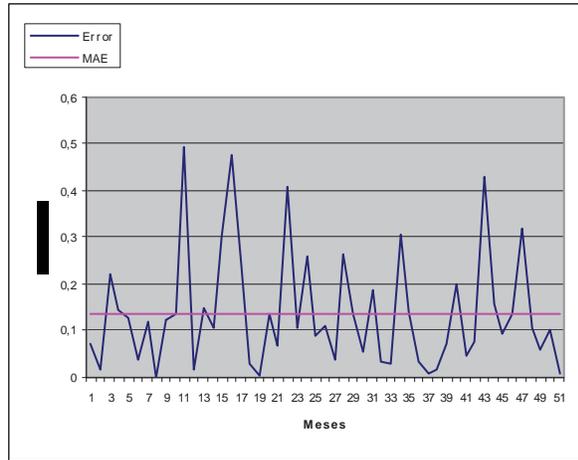


Figura 4.48: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de -18,85%.



En las figuras 4.47 y 4.48 correspondientes al Random 5, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 4.49: Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 4.14 obtenido en el Random 4, con un R^2 de 55,48%.

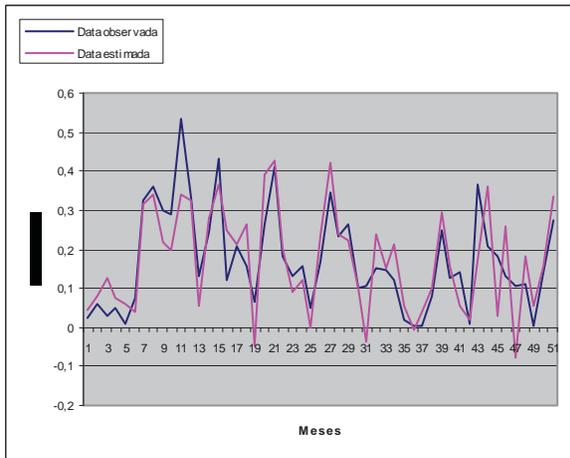
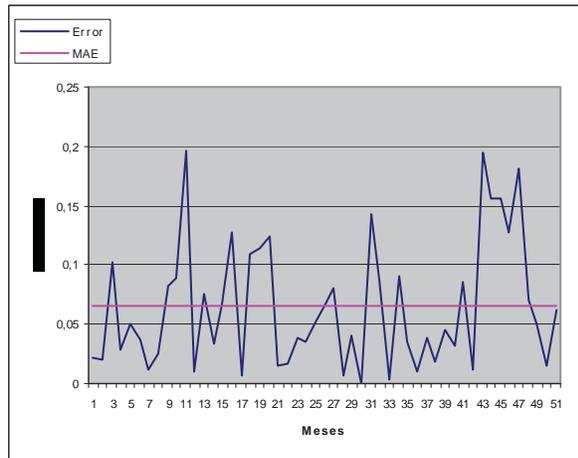


Figura 4.50: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 4, con un R^2 de 55,48%.



En las figuras 4.49 y 4.50 correspondientes al Random 4, la coordenada x corresponde al los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

4.8 Pruebas de las variables del sistema

A continuación se muestran los resultados de las pruebas para encontrar los valores óptimos para las variables de tamaño del enjambre, cantidad de épocas, peso de inercia w , C_1 y C_2 que corresponden a las variables de control de la influencia de las componentes cognitiva y social.

4.8.1 Pruebas para la cantidad óptima de épocas

	500	1000	2000
R^2	0.77051104	0.75964065	0.80743459
ECM	0.00336834	0.00352789	0.00282639

Tabla 4.15: Evaluación de R^2 y ECM con distinta cantidad de épocas del sistema para determinar la cantidad óptima de épocas.

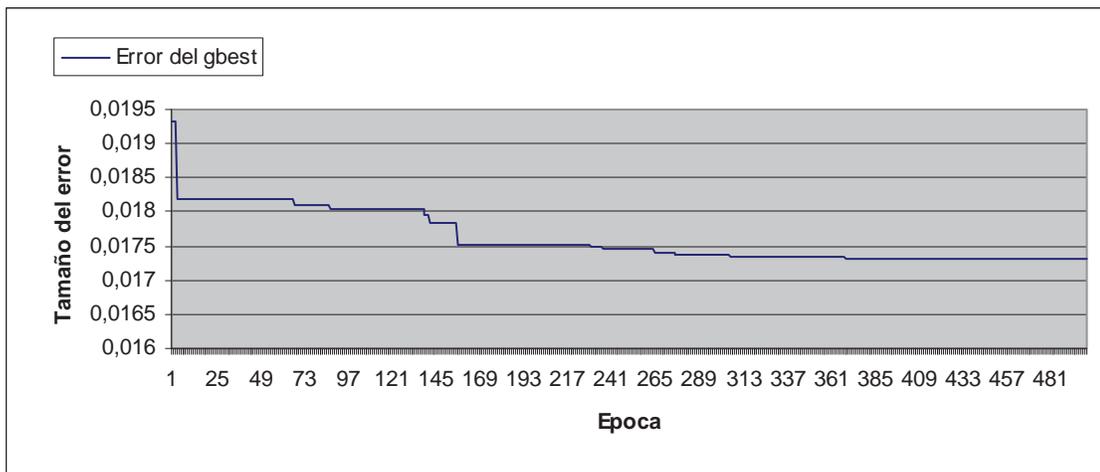


Figura 4.51: Gráfico del descenso del error de la mejor partícula del enjambre en cada una de las épocas del sistema, con un total de 500 épocas

Se puede apreciar en el gráfico de la figura 4.51, que aproximadamente en la época número 385, el valor del error de la mejor partícula del enjambre queda estático, lo que indica que después de esa época la solución del sistema no mejora y por lo tanto es innecesario continuar iterando el sistema.

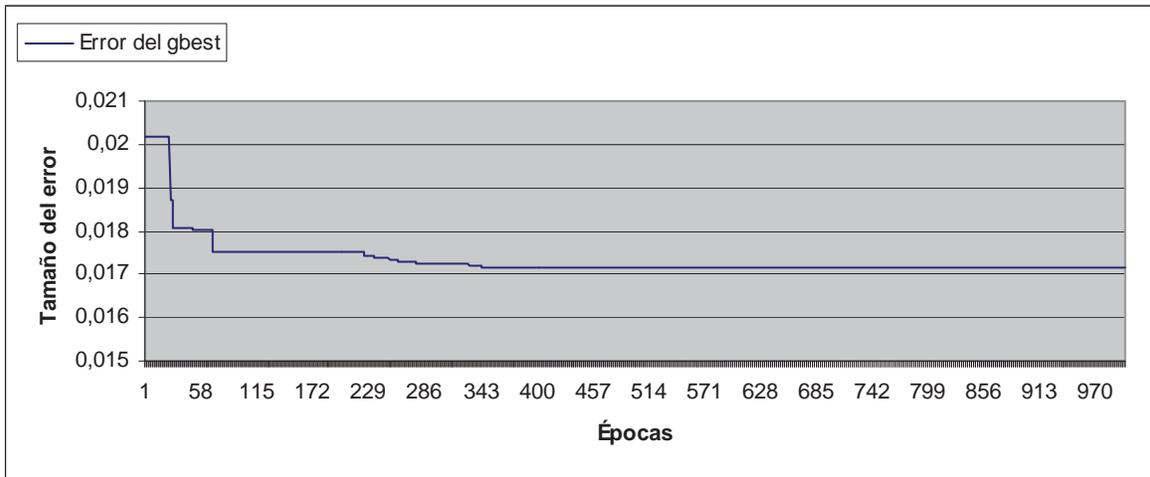


Figura 4.52: Gráfico del descenso del error de la mejor partícula del enjambre en cada una de las épocas del sistema, con un total de 1000 épocas.

Se puede apreciar en el gráfico de la figura 4.52, que aproximadamente en la época número 407, el valor del error de la mejor partícula del enjambre queda estático, lo que indica que después de esa época la solución del sistema no mejora y por lo tanto es innecesario continuar iterando el sistema.

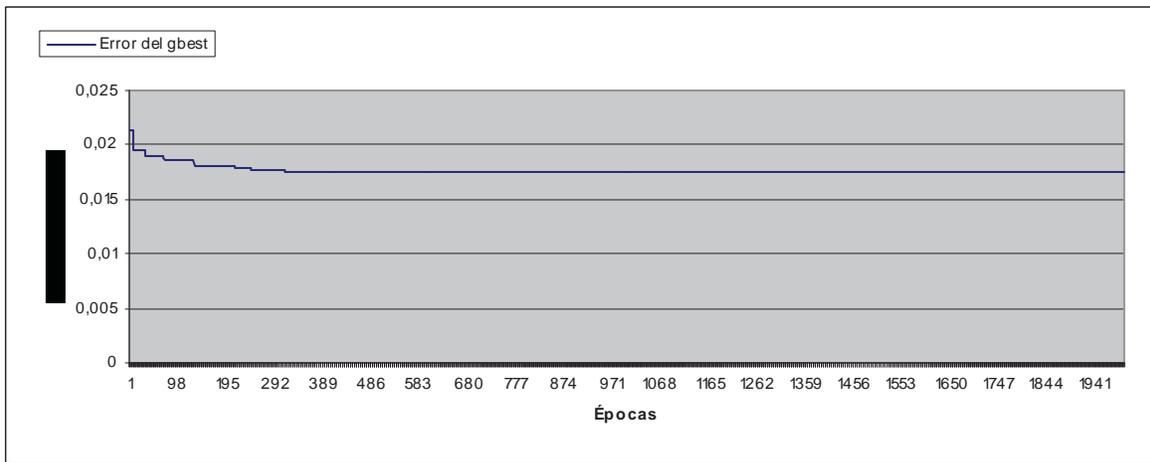


Figura 4.53: Gráfico del descenso del error de la mejor partícula del enjambre en cada una de las épocas del sistema, con un total de 2000 épocas

Se puede apreciar en el gráfico de la figura 4.53, que aproximadamente en la época número 389, el valor del error de la mejor partícula del enjambre queda estático, lo que indica que después de esa época la solución del sistema no mejora y por lo tanto es innecesario continuar iterando el sistema.

En conclusión, observando los resultados obtenidos y los gráficos, se estima que la mejor cantidad de épocas es igual a 500.

4.8.2 Pruebas para la cantidad óptima de partículas

Partículas	1	2	3	4	5	10	15	20	30
R^2	0.16111	0.24213	0.29799	0.30137	0.30294	0.32604	0.30294	0.33304	0.32332
ECM	0.02840	0.01854	0.01717	0.01709	0.01705	0.01648	0.01705	0.01631	0.01655

Tabla 4.16: Evaluación de R^2 y ECM con distinto tamaño de población del PSO

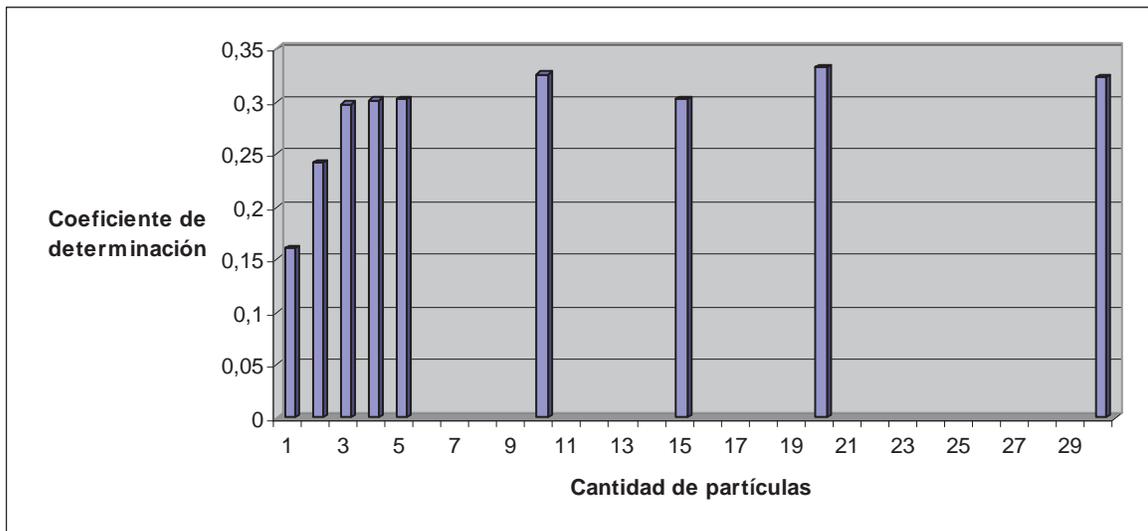


Figura 4.54: Gráfico de la cantidad de partículas y el resultado de su coeficiente de determinación

Se puede apreciar en la figura 4.53, que entre 1 y 5 partículas que el R^2 es relativamente bajo, pero mejora desde 10 partículas, sin embargo entre 10 y 30 partículas no se aprecia gran diferencia en el R^2 , por lo que se concluye que el tamaño del enjambre será de 10 partículas.

4.8.3 Pruebas para el valor de la variable de peso inercial

	0.1	0.3	0.5	0.7	0.9	descendente
R^2 Máx	0,8698090	0,8504260	0,8355192	0,8511820	0,9000056	0,8618921
R^2 Mín	0,5793502	0,7362283	0,6361758	0,7146186	0,6807302	0,6655605
R^2 Med	0,7100718	0,7586403	0,7247231	0,7839368	0,8168167	0,7831774

Tabla 4.17: Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo del peso inercial w .

Como se aprecia en la tabla 4.17, el mejor promedio de R^2 se obtuvo utilizando un w con valor 0,9, dando como resultado R^2 Med igual a 81,68%. Por lo tanto se concluye que el mejor valor para la variable de peso inercial w es de 0,9.

4.8.4 Pruebas para el valor de la variable de influencia cognitiva C_1

	0.2	0.5	0.8	1	1,2	1,49
R^2 Máx	0,9350111	0,9200213	0,8735007	0,9303921	0,8949601	0,9307076
R^2 Mín	0,7609422	0,5636701	0,7172278	0,7424727	0,7090526	0,7573961
R^2 Med	0,7840373	0,7836612	0,7911729	0,7962541	0,8022682	0,8209526

Tabla 4.18: Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo de la variable de influencia cognitiva C_1 .

Como se aprecia en la tabla 4.18, el mejor promedio de R^2 se obtuvo utilizando un C_1 con valor 1,49, dando como resultado R^2 Med igual a 82,09%. Por lo tanto se concluye que el mejor valor para la variable de influencia cognitiva C_1 es de 1,49.

4.8.5 Pruebas para el valor de la variable de influencia social C_2

	0.2	0.5	0.8	1	1,2	1,49
R^2 Máx	0,9542107	0,9598806	0,9383721	0,8725187	0,8885938	0,8601632
R^2 Mín	0,6171661	0,6442293	0,7200863	0,6809637	0,6347648	0,6598078
R^2 Med	0,7684278	0,7875296	0,8297618	0,7876661	0,7962398	0,7879871

Tabla 4.19: Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo de la variable de influencia social C_2 .

Como se aprecia en la tabla 4.19, el mejor promedio de R^2 se obtuvo utilizando un C_2 con valor 0,8, dando como resultado R^2 Med igual a 82,97%. Por lo tanto se concluye que el mejor valor para la variable de influencia social C_2 es de 0,8.

4.8.6 Pruebas para el valor de la variable de velocidad máxima

	0.6	0.06	0.006	0,0006	0,00006	0,000006
R^2 Máx	0,8233668	0,8212536	0,9048236	0,8594287	0,9281936	0,9025575
R^2 Mín	0,4806267	0,6317951	0,6534341	0,7101566	0,7483343	0,6992444
R^2 Med	0,7499268	0,7600523	0,7815852	0,7840732	0,8349318	0,7869566

Tabla 4.20: Evaluación de R^2 con sus valor máximo, mínimo y promedio de 10 muestras randoms realizadas para determinar el valor óptimo de la variable de velocidad v_{\max} .

Como se aprecia en la tabla 4.20, el mejor promedio de R^2 se obtuvo utilizando una velocidad máxima v con valor 0,00006, dando como resultado R^2 Med igual a 83,49%. Por lo tanto se concluye que el mejor valor para la variable v es de 0,00006.

Resultados

5.1 Introducción

En el capítulo anterior se realizaron las pruebas que permitieron determinar el mejor valor para cada una de las variables del modelo de inferencia difusa, con el objeto de encontrar su mejor configuración.

En este capítulo se mostrarán los resultados de la implementación de la mejor configuración del sistema y se analizarán los resultados obtenidos.

5.2 Configuración óptima del sistema

Con todas las pruebas concluidas, se ha determinado el valor que debe tener cada una de las variables del sistema, expresadas en las tablas 5.1 y 5.2 para Takagi Sugeno con PSO Tradicional y Takagi Sugeno con PSO Quantum respectivamente. Es importante recordar que la cantidad de variables del modelo con PSO Tradicional es mayor que la cantidad de variables del modelo con PSO Quantum debido a que este último no posee función de velocidad, tal como ya se explicó en el capítulo 2.

A continuación se entrega la mejor configuración para Takagi Sugeno con PSO Tradicional y Takagi Sugeno con PSO Quantum.

5.2.1 Implementación Takagi Sugeno con PSO Tradicional

Cantidad de partículas	10
Cantidad de épocas	500
Desfase mensual	2
Cantidad de reglas	3
Peso inercial w	0,9
Variable cognitiva C_1	1,49
Variable social C_2	0,8
Velocidad máxima	0,00006

Tabla 5.1: *Mejor configuración del sistema para Takagi Sugeno con PSO Tradicional.*

	R^2	ECM	MAE
Random 1	0.81164028	0.00276466	0.03543557
Random 2	0.74956466	0.00367578	0.04256206
Random 3	0.91548016	0.00124054	0.02037197
Random 4	0.83078198	0.00248371	0.03366763
Random 5	0.85147660	0.00217996	0.03736997
Random 6	0.90988872	0.00132261	0.02849593
Random 7	0.78028721	0.00322485	0.03932125
Random 8	0.83562052	0.00241269	0.03831477
Random 9	0.71661563	0.00415939	0.04331191
Random 10	0.87723414	0.00180190	0.02943375
Media	0.82785899	0.00252661	0.03482848

Tabla 5.2: Resultados de la mejor configuración para Takagi Sugeno con PSO Tradicional.

Como se puede apreciar en la tabla 5.2, los resultados obtenidos de la mejor configuración del sistema son considerablemente buenos ya que el mayor valor obtenido para R^2 es de 91,54%. Además el 60% de los resultados de los randoms obtuvo un R^2 superior al 80%.

Para ver mejor la efectividad del modelo, se entregan a continuación la gráfica de las curvas de la data observada y la data estimada.

Figura 5.1: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.2 obtenido en el Random 9, con R^2 de 71,66%.

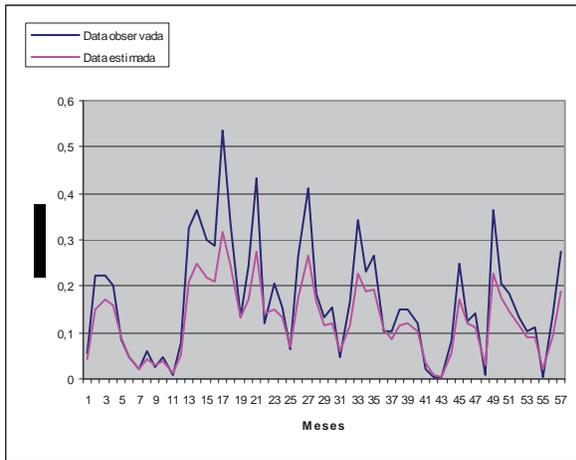
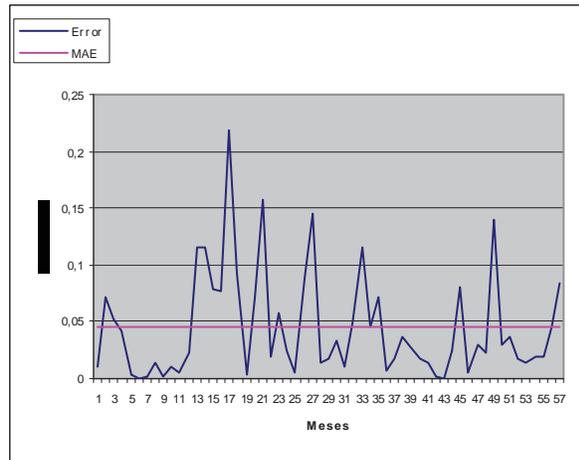


Figura 5.2: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 71,66%.



En las figuras 5.1 y 5.2 correspondientes al Random 9, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 5.3: Gráfico de la data estimada v/s la data observada de mejor resultado de la tabla 5.2 obtenido en el Random 3, con un R^2 de 91,54%.

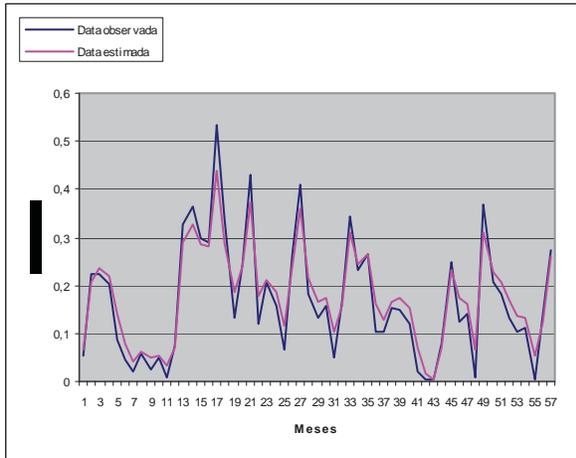
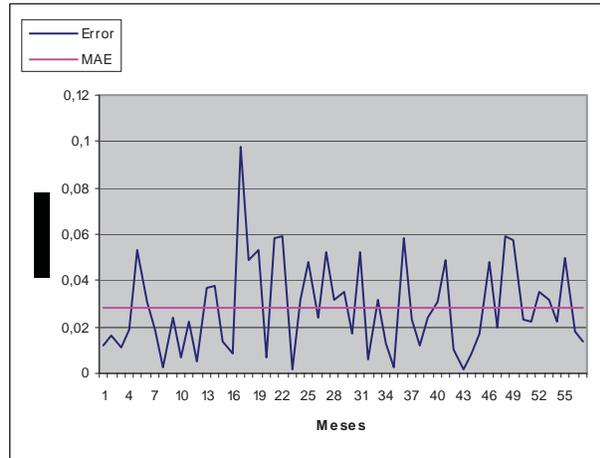


Figura 5.4: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 3, con un R^2 de 91,54%.



En las figuras 5.3 y 5.4 correspondientes al Random 3, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

5.2.2 Implementación Takagi Sugeno con PSO Quantum

Cantidad de partículas	10
Cantidad de épocas	500
Desfase mensual	2
Cantidad de reglas	3

Tabla 5.3: *Mejor configuración del sistema para PSO Quantum.*

	R^2	ECM	MAE
Random 1	0.81573604	0.00270455	0.03647875
Random 2	0.79029566	0.00307795	0.03924349
Random 3	0.73082543	0.00395083	0.04514792
Random 4	0.83393522	0.00243743	0.03579925
Random 5	0.88904366	0.00162857	0.02708696
Random 6	0.83921602	0.00235992	0.03072854
Random 7	0.82654861	0.00254584	0.03625514
Random 8	0.76925994	0.00338670	0.04330791
Random 9	0.59842853	0.00589410	0.05410232
Random 10	0.84272203	0.00230846	0.03404936

Tabla 5.4: *Resultados de la mejor configuración para Takagi Sugeno con PSO Quantum.*

La tabla 5.4 muestra los resultados de la mejor configuración para Takagi Sugeno con PSO Quantum, los cuales no superan a la configuración de Takagi Sugeno con PSO Tradicional.

A continuación se muestran los gráficos del peor resultado y del mejor resultado de la tabla 5.4.

Figura 5.5: Gráfico de la data estimada v/s la data observada del peor resultado de la tabla 5.4 obtenido en el Random 9, con R^2 de 59,84%.

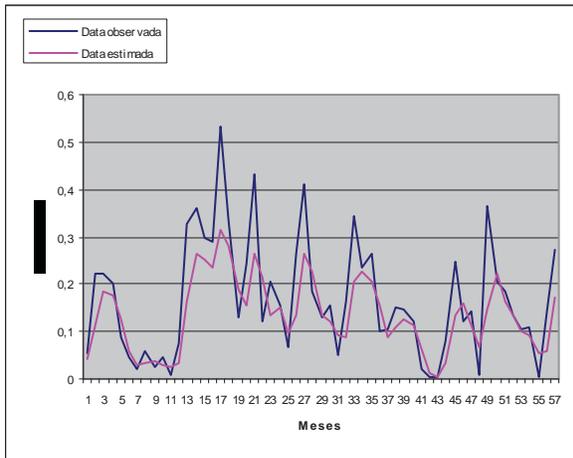
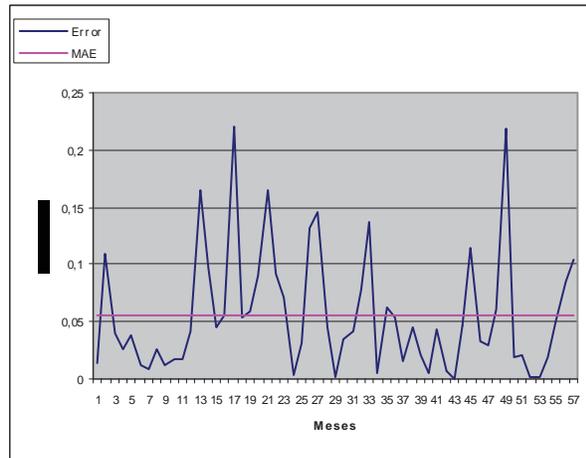


Figura 5.6: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 9, con un R^2 de 59,84%.



En las figuras 5.5 y 5.6 correspondientes al Random 9, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

Figura 5.7: Gráfico de la data estimada v/s la data observada del mejor resultado de tabla 5.4 obtenido en el Random 5, con un R^2 de 88,9%.

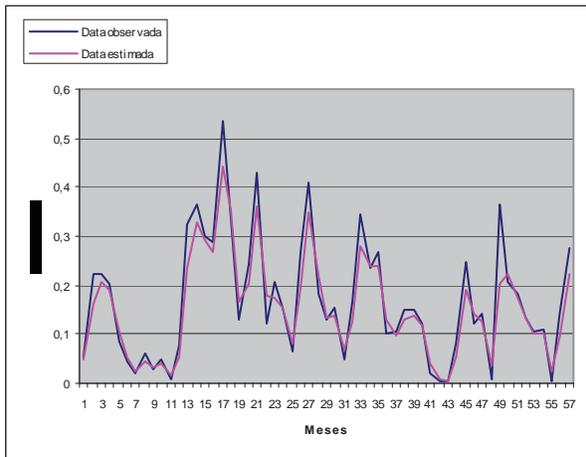
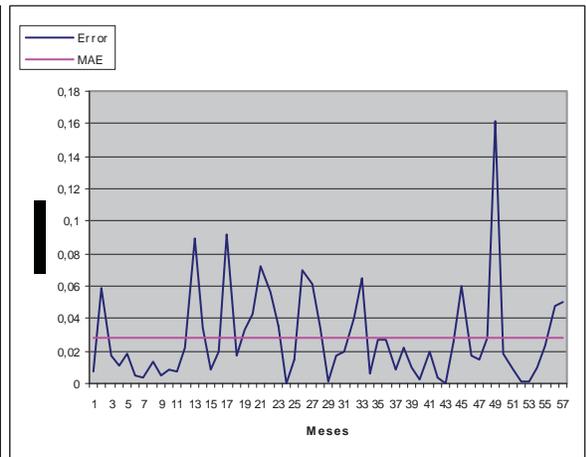


Figura 5.8: Gráfico del error de la data estimada con respecto a la data observada y su error medio (MAE) en el Random 5, con un R^2 de 88,9%.



En las figuras 5.7 y 5.8 correspondientes al Random 5, la coordenada x corresponde a los meses de estimación y la coordenada y corresponde a las toneladas de captura de anchovetas expresadas en valores normalizados entre [0;1] con la data de testing desde el año 2002 hasta el año 2007.

5.3 Análisis de resultados

Analizando los resultados obtenidos de la mejor configuración para Takagi Sugeno con PSO Tradicional y Takagi Sugeno con PSO Quantum, se entrega como conclusión que la configuración óptima del modelo de inferencia difusa se logra con los valores de la tabla 5.1 correspondiente al modelo Takagi Sugeno con PSO Tradicional.

Realizando un análisis de los resultados del capítulo 4, se podría decir que para resolver el problema propuesto en esta memoria de título, el mejor modelo se obtiene con PSO Quantum, ya que en las pruebas muestra mejores resultados que PSO Tradicional, sin embargo al aplicar la mejor configuración de ambos modelos se concluye que PSO Tradicional entrega los mejores resultados. Esto se debe a que en el capítulo 4, el PSO Tradicional no tenía configurada la función de velocidad y sus variables no eran las óptimas, esto hacía que PSO Quantum fuera superior, pero al configurar las variables del modelo con PSO Tradicional, este último presentó con mejores resultados.

Es importante señalar que los resultados obtenidos no demuestran que PSO Tradicional es superior a PSO Quantum, únicamente demuestran que PSO Tradicional entrega mejores resultados para el problema propuesto en esta memoria de título y que es posible realizar un buen ajuste de un modelo de inferencia difusa de tipo Takagi Sugeno con optimización por enjambre de partículas para este problema en específico.

El resultado final del sistema es un R^2 de 91,54%.

Capítulo 6

Conclusiones

En el presente proyecto de título se explicó el modelo de inferencia difusa Takagi Sugeno (TS) y se analizó detalladamente con el fin de poder utilizarlo para resolver el problema propuesto que es entregar un pronóstico para la cantidad de toneladas de capturas de anchovetas del mes siguiente en el norte de Chile. A modo de poder optimizar sus parámetros de inferencia, explicados en el capítulo 5, se explicó también el modelo de optimización por enjambre de partículas (PSO) y se elaboró una solución, la cual consistió en hacer que cada partícula del enjambre fuera una posible solución de optimización para los parámetros del TS.

Los resultados obtenidos en el capítulo 6 demuestran que el PSO es capaz de optimizar eficazmente los parámetros del modelo TS y que además el modelo TS es un buen estimador para entregar el pronóstico de la cantidad de capturas de anchovetas del mes siguiente. Dicha aseveración es demostrada por los resultados obtenidos de los estimadores usados en el capítulo 6 para determinar la eficacia del modelo, dentro de los cuales, el estimador coeficiente de determinación R^2 entregó una bondad de ajuste del modelo de 91,54% en la etapa de pruebas del modelo con los mejores valores encontrados para las variables del sistema.

La mejor configuración obtenida de la etapa de prueba para el PSO tradicional consta de un tamaño de enjambre de 10 partículas, 500 épocas, un valor de 0,9 para el peso inercial, un valor igual a 1,49 para la variable de influencia cognitiva C_1 , un valor igual a 0,8 para la variable de influencia social C_2 y la velocidad máxima que puede alcanzar una partícula es de 0,00006. Para el caso del modelo TS, la configuración consta de un desfase mensual de 2 meses y 3 reglas.

Es importante destacar que en la etapa de pruebas se determinó que la cantidad de parámetros necesarios para generar el pronóstico es igual a 12, lo cual significa que la carga del sistema es considerablemente baja.

Además, si el modelo obtenido en este proyecto se compara con el modelo de Red Neuronal Artificial propuesto por los autores de [10], se puede concluir que los resultados de este proyecto superan los resultados obtenidos por dicha red neuronal, tanto en carga de sistema como en la bondad de ajuste del modelo, ya que el modelo neuronal tiene entre 30 a 40 neuronas en su capa oculta, alcanzando un valor de 88,04% para R^2 y el modelo TS de este proyecto tiene solo 6 funciones de membresía, las cuales vienen siendo las homólogas de las neuronas de la capa oculta del modelo neuronal, además el modelo TS con PSO alcanzó un valor de 91,54% para R^2 .

Se realizó la comparación entre PSO tradicional y PSO Quantum (QPSO) para ver si esta variante de PSO podía entregar mejores resultados. Al principio QPSO entregó mejores resultados que PSO tradicional debido a que QPSO solo tiene una variable de desplazamiento, explicada en el capítulo 5, a diferencia de PSO tradicional que posee las

variables propias de su función de velocidad, las cuales al inicio de su implementación no estaban bien ajustadas para proporcionar buenos resultados. Sin embargo después que se realizaron las pruebas para ajustar cada una de estas variables de la función de velocidad de PSO tradicional, este entregó mejores resultados que QPSO.

El resultado final de las pruebas indica que para este problema específico, PSO tradicional entrega los mejores resultados, lo cual no significa que PSO tradicional sea mejor que QPSO ni que PSO tradicional sea la solución a cualquier problema de optimización. Sin embargo es importante señalar que este modelo de sistema sí sirve para resolver problemas no lineales con datos que no son estacionarios.

Como acotación final, se afirma que el sistema de inferencia difusa propuesto es capaz de resolver el problema propuesto en este proyecto de título.

Referencias

- [1] Mauricio Braun A., Hernán Reyes R., Vivian Valenzuela C., Elisa Pizarro R., Jorge Castillo P., Víctor Cataste B., Juan Saavedra N., José Luis Blanco G., “Monitoreo de las Condiciones Bio-Oceanográficas entre la I y II Regiones”, Proyecto FIP N°2003-01, Mayo 2005
- [2] <http://www.viarural.cl/alimentos/pescados-y-mariscos/anchoveta/anchoveta.htm>
- [3] <http://www.monografias.com/trabajos45/anchoveta-peruana/anchoveta-peruana2.shtml>
- [4] Olga Lucia Quintero Montoya, “Modelamiento e Identificación Takagi Sugeno en un Intercambiador de Calor”, paper, Junio 2003.
- [5] Eliana M. Toro Campo, Yov Steven Restrepo G., mauricio Granada E. “Adaptación de la técnica de particle swarm al problema de secuenciamiento de tareas”, paper, Diciembre 2006.
- [6] A. Abraham, H. Guo, H. Liu, “Swarm Intelligence: Foundations, Perspectives and Applications”, *Studies in Computational Intelligence (SCI)*, Springer-Verlag, vol. 26, pages 3-25, November 2006.
- [7] M. Sanaz^{1,2}, H. Wener, A. Wille, “Linear Multi-Objective Particle Swarm Optimization”, *Studies in Computational Intelligence (SCI)*, Springer-Verlag, vol. 31, pages 209-238, October 2006.
- [8] E. Ozcan, M. Yilmaz, “Particle Swarm for Multimodal Optimizacion”, *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 4431, pages 366-375, July 2007.
- [9] A. Khosla, S. Kumar, K.K. Aggarwal, J. Singh, “Particle Swarm for Fuzzy Models Identification”, *Studies in Computational Intelligence (SCI)*, Springer-Verlag, vol 26, pages 149-173, November 2006.
- [10] J. Kennedy, R.C. Eberhart, “Particle Swarm Optimization”, *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pages 1942-1948, December 1995.
- [11] J. Kennedy, R.C. Eberhart, Y. Shi, “Swarm Intelligence”, *The Morgan Kaufmann Series in Artificial Intelligence*, Morgan Kaufmann, San Francisco - California, pages 3-80, April 2001.
- [12] J.R. Pérez, “Contribución a los Métodos de Optimización Basados en Procesos Naturales y su Aplicación a la Medida de Antenas de Campo Próximo”, a Thesis presented to the University of Cantabria, October 2005.
- [13] L. Fang, P. Chen, S. Liu, “Particle Swarm Optimization with Simulated Annealing for TSP”, *Proceedings of the 6th Conference on 6th WSEAS*, vol. 6, pages 206-210, February 2007.
- [14] V. Méndez, F. García, “PSO vs. ACO, Data Grid Replication Service Performance Evaluation”, *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 4331, pages 833-843, November 2006.

- [15] A. Khosla, S. Kumar, K.K. Aggarwal, J. Singh, "Introducing Lifetime Parameter in Selection Based Particle Swarm Optimization for Improved Performance", *First Indian International Conference on Artificial Intelligence (IICAI-03)*, Hyderabad, India, pages 1175-1181, December 2003.
- [16] X. Xie, W. Zhang, Z. Yang, "A Dissipative Particle Swarm Optimization", *Proceedings of the 2002 Congress on Evolutionary Computation (CEC-02)*, vol. 2, pages 1456-1461, October 2002.
- [17] J. Gimmler, T. Stutzle, T.E. Exner "Hybrid Particle Swarm Optimization: An Examination of the Influence of Iterative Improvement Algorithms on Performance", *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 4150, pages 436-443, August 2006.
- [18] M. Iqbal, M.A. Montes de Oca, "An Estimation of Distribution Particle Swarm Optimization", *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 4150, pages 72-83, August 2006.
- [19] A.E. Eiben, Z. Michalewicz, M. Schoenauer, J.E. Smith, "Parameter Control in Evolutionary Algorithms", *Studies in Computational Intelligence (SCI)*, Springer-Verlag, vol. 54, pages 19-46, April 2007.
- [20] A.E. Eiben, E. Marchiori, V.A. Valkó, "Evolutionary Algorithms with On-the-Fly Population Size Adjustment", *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 3242, pages 41-50, December 2004.
- [21] J. Maturana, F. Saubion, "Automated Parameter Control for Evolutionary Algorithms", *1st Workshop on Autonomous Search*, September 2007.
- [22] V. Nannen, A.E. Eiben, "Relevance Estimation and Value Calibration using Meta-Algorithms", *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 975-980, January 2007.
- [23] E. Monfroy, C. Castro, B. Crawford, "Adaptive Enumeration Strategies and Metabacktracks for Constraint Solving", *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 4243, pages 354-363, October 2006.
- [24] X. Pu, Z. Fang, Y. Liu, "Multilayer Perceptron Networks Training Using Particle Swarm Optimization with Minimum Velocity Constraints", *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, vol. 4493, pages 237-245, July 2007.
- [25] Juan Carlos Gutiérrez Estrada, Claudio Silva, Eleuterio Yáñez, Nibaldo Rodríguez, Inmaculada Pulido Calvo, "Monthly catch forecasting of anchovy *Engraulis ringens* in the north area of Chile: Non-linear univariate approach", Junio 2007.
- [26] H. Shono, "Is Model Selection Using Akaike's Information Criterion Appropriate for Catch per Unit Effort Standardization in Large Samples?", *Fisheries Science*, Blackwell, vol. 71, no. 5, pages 978-986, October 2005.
- [27] C.J. Lopez-Toribio, R.J. Patton, S. Daley. "Takagi-Sugeno Fuzzy Fault-Tolerant Control of an Induction Motor", paper, 2000.
- [28] http://en.wikipedia.org/wiki/Lotfi_Asker_Zadeh
- [29] José Manuel García Nieto, "Algoritmos basados en Cúmulos de partículas para la Resolución de Problemas Complejos", Memoria de Título, página 25, Septiembre 2006

- [30] P. C. Nayak, K. P. Sudheer, “Fuzzy model identification based on cluster estimation for reservoir inflow forecasting”, Publicación en Wiley InterScience, Junio 2007
- [31] Jesús R. Pérez, José Basterrechea, “Optimización con enjambre de partículas aplicada a la reconstrucción del diagrama de radiación de antenas”. Publicación, ETSIT. Dpto. Ingeniería de comunicaciones, Universidad de Cantabria, 2005

Anexo A: Código Fuente

Main Class

```
/*
 * Sistema.java
 *
 * Created on 10 de junio de 2009, 03:33 PM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */

package tpsso;

/**
 *
 * @author Roderich Henriquez
 */
public class Sistema {

    /** Creates a new instance of Sistema */
    public Sistema() {
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Interfaz ventana = new Interfaz();
        ventana.setVisible(true);
    }
}
```

Takagi

```
/*
 * Enjambre.java
 *
 * Created on 17 de marzo de 2009, 10:32 AM
 *
```

```
* To change this template, choose Tools | Template Manager
* and open the template in the editor.
*/
```

```
package tspso;
```

```
import java.util.Random;
```

```
/**
```

```
 *
```

```
 * @author Roderich Henriquez
```

```
 */
```

```
public class Takagi{
```

```
    Datas base;
```

```
    private double x[]; // vector de variables de entrada
```

```
    private double centros[][]; // valores de los centros de las funciones de membresia
```

```
    private double radios[][]; // valores de los radios de las funciones de membresia
```

```
    private double membresia[][]; // vector de resultados de las regla
```

```
    private double product[]; // vector de resultados de las productorias
```

```
    private double w[]; // vector de valores normalizados
```

```
    private int reglas; // cantidad de reglas del modelo
```

```
    private int ents ; // tamaño del vector de entrada
```

```
    private int dimensiones;
```

```
    private double norma; // valor del divisor de la normalizacion
```

```
    private double salida;
```

```
    private double particula[]; // corresponde a la mejor particula entregada por el PSO
```

```
(gbest)
```

```
    private double polinomios[][];
```

```
    private double y[]; // valores del polinomio de optimizacion
```

```
    private double error[]; // guarda el error de cada muestra en un forward de TS
```

```
    private double coef_1r;
```

```
    public Takagi(){
```

```
    }
```

```
    public Takagi(int a,int b, int c,int vecs) {
```

```
        ents = a;
```

```
        reglas = b;
```

```
        dimensiones = c;
```

```
        x = new double[ents]; // x[i][k] donde i es el orden de los datos, k es la iteracion del
```

```
vector de entrada
```

```
        membresia = new double[reglas][ents];
```

```
        product = new double[reglas];
```

```
        w = new double[reglas];
```

```
        y = new double[reglas];
```

```
    particula = new double[dimensiones]; // matriz de informacion entregada por la
particula gbest
    centros = new double[reglas][ents];
    radios = new double[reglas][ents];
    error = new double[vecs];
    polinomios = new double[reglas][ents];
}

public void inicio_ts() // inicia los valores de las variables de TS
{
    int i,k,j;
    k = 0;

    for(i=0;i<reglas;i++) // numero de la regla
    {
        for(j=0;j<ents;j++) // numero de la entrada
        {
            centros[i][j] = particula[k]; // inicia los centros de todas las reglas
            k++;
        }
    }
    for(i=0;i<reglas;i++) // numero de la regla
    {
        for(j=0;j<ents;j++) // numero de la entrada
        {
            radios[i][j] = particula[k]; // inicia los centros de todas las reglas
            k++;
        }
    }
    for(i=0;i<reglas;i++) // numero de la regla
    {
        for(j=0;j<ents;j++) // numero de la entrada
        {
            polinomios[i][j] = particula[k]; // inicia los centros de todas las reglas
            k++;
        }
    }
}

public void calcular_reglas() // resultados de las funciones de membresia
{
    int i,k,j;
    double res;

    for(i=0;i<reglas;i++) // k es el numero de la regla
    {
        for(j=0;j<ents;j++) // i es el numero de la funcion de membresia
```

```
        {
            res = ( x[j]-centros[i][j] ) / ( radios[i][j]*radios[i][j]);
            membresia[i][j] = Math.exp(res);
        }
    }
}
public void productoria() // calcula todos los valores de las productorias
{
    double aux;
    int i,j;

    aux = 1;
    for(i=0;i<reglas;i++)
    {
        product[i] = 1;
    }
    for(i=0;i<reglas;i++) // id de regla
    {
        for(j=0;j<ents;j++) // id entrada
        {
            product[i] = product[i] * membresia[i][j];
            //System.out.print("\n resultado productoria =" + product[j]);
        }
    }
}
public void normalizar()
{
    int i;

    for(i=0;i<reglas;i++)
    {
        w[i] = product[i]/norma;
    }
}
//***** setear valores de variables / arreglos *****
public void set_x(int i, double valor)
{
    x[i] = valor;
}
public void set_vect_products()
{
    int i;

    for(i=0;i<reglas;i++)
    {
        product[i]=1;
    }
}
```

```
}
public void set_norma()
{
    int i;
    norma = 0;
    for(i=0;i<reglas;i++)
    {
        norma = norma + product[i];
    }
}
public void set_salida()
{
    int i,j,k;
    salida = 0;

    for(i=0;i<reglas;i++)
    {
        y[i] = 0;
    }
    k = 0;
    for(i=0;i<reglas;i++)
    {
        for(j=0;j<ents;j++)
        {
            // dim - ents*reg es desde donde comienzan los terminos en la
particula
            y[i] = y[i] + polinomios[i][j] * x[j];
        }
        salida = salida + (y[i] * w[i]); // calcula el valor de salida Y del TS
    }
}
public void set_particula(int i,double val)
{
    particula[i] = val;
}
//***** obtener valores de variables / arreglos *****

public double get_x(int i)
{
    return(x[i]);
}
public double get_salida()
{
    return(salida);
}
public double acotar(double valor)
{
    if(valor < 0)
```

```
        {
            valor = valor -1;
        }
    if(valor > 1)
        {
            valor = valor /10;
        }
    return(valor);
}
public void set_error(double esperado, int i)
{
    error[i] = esperado - salida;
}
public double get_ecm(int cant_vects)
{
    int i;
    double resultado = 0;
    for(i=0;i<cant_vects;i++)
        {
            resultado = resultado + Math.pow(error[i],2);
        }
    resultado = resultado / cant_vects;
    return(resultado);
}
}
```

Enjambre

```
/*
 * Enjambre.java
 *
 * Created on 31 de marzo de 2009, 09:45 PM
 *
 */

package tspso;

import java.lang.Math;
import java.util.Random;

public class Enjambre {

    private double vector_entrada[][]; // la matriz se llama vector_entrada porque representa
    el vector de entrada del
        // modelo, pero contiene la coleccion de valores de la base de datos
```

```
private double particula_pos[][]; // matriz de posiciones de n particulas con d
dimensiones
private double particula_vel[][]; // matriz de velocidades de las particulas
private double pbestk[][];
private double w; // factor inercial
private double medias[];
private double c1,c2;
private double r1,r2;
private double fitness[];
private double bestfitness;
private double part_gbest[];
private double besta,bestb;
private double desv; // valor de la desviacion estandar
private double media; // valor de la media
private double Vmax = 0.00006;
private int gbest; // guarda la posicion de la mejor particula
private int iteracion;
private int n; // tamaño del vector de entrada
private int dim; // cantidad de dimensiones de la particula
private int cp; // cantidad de particulas del modelo
private int gen; // representa la cantidad vectores de entrada que se emplearán
private int num_gen; // id del vector de entrada actual del sistema
private int muestras = 20; // cantidad de muestras
private Random rand;
private Variables vals;
private Datas dats;

// NOTA: las dimensiones de la particula contienen primero todos los centros, luego
// todos los radios y finalmente los terminos de ajuste del TS

public Enjambre() {
}
public Enjambre(int dims,int parts, int ents,int rgl) {

    dim = dims;
    cp = parts;
    //declaracion de variables
    // vals = new Variables(ents,rgl,parts);
    dats = new Datas();
    gen = cant_vectores();
    double mejor_fitness = 100000; // valor grande para encontrar el fitness mas pequeño
    double error = 1000000;
    double aux4 = 0,aux5 = 0;
    double fitnessaux,fitnessaux2;
    double aux1,aux2,aux3,aux8; // variables auxiliares
    int i,j,k,s;
    int nada =0;
```

```
int iteracion = 0;
particula_pos = new double[cp][dim];
particula_vel = new double[cp][dim];
medias = new double[dim];
//gbest = 5;
pbestk = new double[cp][dim];
vector_entrada = new double [n][gen];
fitness = new double[cp]; // vector que almacena el fitness de cada particula
part_gbest = new double[dim];
rand = new Random();

//inicializacion de variables
desv = 0;
media = 0;
c1 = 1.49;
c2 = 0.8;
w = 0.9;

}
// setear e inicializar variables, arreglos, etc
public void inicializar_enjambre()
{
    Random rnd;
    int i,j;
    rnd = new Random();
    inicializar_fitness();
//***** inicializacion del enjambre
    for(i=0;i<cp;i++)
    {
        for(j=0;j<dim;j++)
        {
            particula_pos[i][j] = acotar_enj(rnd.nextDouble());// valores random a las
posiciones
            pbestk[i][j] = particula_pos[i][j]; // mejores posiciones = a las posiciones
iniciales
            particula_vel[i][j] = acotar_enj(rnd.nextDouble()); // valores random a las
velocidades
        }
    }
    set_gbest();
}
public void actualizar_particula(int i,boolean act) // i representa el identificador de la
particula
{
    // act es el tipo de actualizacion: max-min(true) o
quantum(false)
    int j,k,d;
    double aux1,aux2,acum;
```

```
double u,fi,alfa,p;
Random rnd;

rnd = new Random();

// actualización por PSO Quantum
if(act == false)
{
    for(d=0;d<dim;d++)
    {

        u = rnd.nextDouble()*(double)0.25;
        fi = (double)0.3;
        alfa = rnd.nextDouble() * (double)1.4 + (double)-0.7;
        p = fi * pbestk[i][d] + (1-fi) * part_gbest[d];
        partícula_pos[i][d] = p + alfa*Math.abs(medias[d] - partícula_pos[i][d]) *
(double)Math.log(1/u);
    }
}

// actualización por PSO max/min
if(act == true)
{
    for(j=0;j<dim;j++)
    {
        r1 = acotar_pos(rnd.nextDouble());
        r2 = acotar_pos(rnd.nextDouble());
        if(r1<0){r1 = r1*-1;}
        if(r2<0){r1 = r1*-1;}
        aux1 = c1 * r1 * (pbestk[i][j] - partícula_pos[i][j]);
        aux2 = c2 * r2 * (part_gbest[j] - partícula_pos[i][j]);
        partícula_vel[i][j] = w * partícula_vel[i][j] + aux1 + aux2;
        if((partícula_vel[i][j]<(-Vmax)) || (partícula_vel[i][j]>Vmax))
        {
            partícula_vel[i][j] = signum(partícula_vel[i][j]) * media_vel();
        }
        partícula_pos[i][j] = partícula_pos[i][j] + partícula_vel[i][j];
    }
}
}

public void evaluar_fitness(int i,double error)
{
    int j,a,b;
    double fitnessaux = 0;

    if(error < fitness[i]) // evalua mejor personal
    {
```

```
    fitness[i] = error;
    for(a=0;a<dim;a++)
    {
        pbestk[i][a] = particula_pos[i][a]; // actualiza mejor personal
    }
}
if(error < bestfitness) // evalua mejor global
{
    bestfitness = error;
    for(b=0;b<dim;b++)
    {
        part_gbest[b] = particula_pos[i][b];
    }
}
}
// obtencion de valores
public double get_rand()
{
    double valor;
    valor = rand.nextDouble();
    return(valor);
}
public double signum(double nro)
{
    if(nro > 0)
    {
        nro = 1;
    }
    if(nro < 0)
    {
        nro = -1;
    }
    return(nro);
}
public double media_vel() // calculo de la media de las velocidades del enjambre
{
    // donde i representa el id de la particula y j la dimension
    int i,j;
    double acum = 0;

    for(i=0;i<cp;i++)
    {
        for(j=0;j<dim;j++)
        {
            if(particula_vel[i][j] < 0)
            {
                acum = acum + (particula_vel[i][j] * -1);
            }
        }
    }
}
```

```
        else
        {
            acum = acum + particula_vel[i][j];
        }
    }
}
acum = acum / (dim * cp);
return(acum);
}
public int cant_vectores()
{
    int num;
    num = (muestras + 1) - n; // n es el tamaño del vector de entrada
    return(num);
}

public void print_gbest()
{
    int i;
    for(i=0;i<dim;i++)
    {
        System.out.print(" " + part_gbest[i]);
    }
}

public double acotar_enj(double valor)
{
    if(valor > 1)
    {
        do{
            valor = valor / 10;
        }while(valor > 1);
    }
    if(valor < -0.1)
    {
        do{
            valor = valor /10;
        }while(valor < -1);
    }
    return(valor);
}

public double acotar_pos(double valor)
{
    if(valor < 0)
    {
        valor = valor * -1;
    }
}
```

```
do{
    valor = valor /10;
}while(valor > 1);

return(valor);
}
public double get_gbest(int i)
{
    return(part_gbest[i]);
}
public double get_bestfitness()
{
    return(bestfitness);
}
public double get_particula_pos(int i,int d) // i es el numero de la particula
// d es el numero de la dimencion
{
    return(particula_pos[i][d]);
}
public double get_particula_vel(int i,int d) // i es el numero de la particula
// d es el numero de la dimencion
{
    return(particula_vel[i][d]);
}
public void peso_inercial_res()
{
    w = w - 0.00166;
}
public void peso_inercial_add()
{
    w = w + 0.00166;
}
public double get_inercia()
{
    return(w);
}
public void set_medias()
{
    int d,p;
    double acum;
    for(d=0;d<dim;d++) // calcula la media de las mejores posiciones del enjambre por
dimension
    {
        acum = 0;
        for(p=0;p<cp;p++)
        {
            acum = acum + pbestk[p][d];
        }
        medias[d] = acum / cp;
    }
}
```

```
    }  
  }  
  public void set_pos(int i,int j, double val)  
  {  
    particula_pos[i][j] = val;  
    pbestk[i][j] = val;  
  }  
  public void set_vel(int i,int j, double val)  
  {  
    particula_vel[i][j] = val;  
  }  
  public void set_gbest()  
  {  
    Random ran;  
    int i;  
    ran = new Random();  
    for(i=0;i<dim;i++)  
    {  
      part_gbest[i] = acotar_enj(ran.nextDouble()); // valores random a las posiciones  
    }  
  }  
  public void inicializar_fitness()  
  {  
    int i;  
    for(i=0;i<cp;i++)  
    {  
      fitness[i] = 1000000; // inicializar con un valor grande para encontrar el menor  
    }  
    bestfitness = 1000000;  
  }  
  public void set_gbest2(int dim,double val)  
  {  
    part_gbest[dim] = val;  
  }  
}
```