



**UNIVERSIDAD AUTÓNOMA CHAPINGO**

**Posgrado en Ingeniería Agrícola y Uso Integral del Agua**



**TÉCNICAS DE INTELIGENCIA ARTIFICIAL EN MODELADO Y CONTROL DEL CLIMA EN INVERNADEROS**

**TESIS**

QUE COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:  
DOCTOR EN INGENIERÍA AGRÍCOLA Y USO INTEGRAL DEL AGUA



DIRECCION GENERAL ACADEMICA  
DEPTO. DE SERVICIOS ESCOLARES  
OFICINA DE EXAMENES PROFESIONALES

PRESENTA:

**M en C ELMER CÉSAR TREJO ZÚÑIGA**

BAJO LA DIRECCIÓN DE:

**DR. IRINEO L. LÓPEZ CRUZ**

**Enero del 2014**

**CHAPINGO, ESTADO DE MÉXICO**

TÉCNICAS DE INTELIGENCIA ARTIFICIAL EN MODELADO Y CONTROL DEL CLIMA EN INVERNADEROS

Tesis realizada por Elmer César Trejo Zúñiga bajo la dirección del Comité Asesor indicado, aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

DOCTOR EN INGENIERÍA AGRÍCOLA  
Y USO INTEGRAL DEL AGUA

DIRECTOR:

  
\_\_\_\_\_  
DR. IRINEO D. LÓPEZ CRUZ

ASESOR:

  
\_\_\_\_\_  
DR. J. ARMANDO RAMÍREZ ARIAS

ASESOR:

  
\_\_\_\_\_  
DR. EFRÉN FÍJZ RODRÍGUEZ

LECTOR EXTERNO:

  
\_\_\_\_\_  
DR. JOHN CH. GODDARD CLOSE

## **Datos Biográficos**

El autor del presente trabajo, Elmer César Trejo Zúñiga, realizó sus estudios de ingeniería en la Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME) del Instituto Politécnico Nacional (IPN) obteniendo el grado de Ingeniero en Comunicaciones y Electrónica con especialidad en control automático en el año 2005.

Posteriormente, realizó sus estudios de Maestría en el Centro de Investigaciones y Desarrollo de Tecnología Digital (CITEDI) del Instituto Politécnico Nacional (IPN) obteniendo el grado de Maestro en Ciencias en Sistemas Digitales con especialidad en Control en el año 2008.

## Agradecimientos

*Al Padre Azul (Ab – b̄) ...*

*Un agradecimiento especial al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico brindado durante mis estudios doctorales, sin ello hubiera sido imposible culminarlos.*

*A la Universidad Autónoma Chapingo (UACH) por la oportunidad que me dió de formarme como profesional en sus aulas y adquirir conocimientos sólidos transmitidos por su completo grupo de especialistas en las diferentes materias que se imparten en el posgrado en Ingeniería Agrícola y Uso Integral del Agua (Especialidad en Biosistemas). Gracias a todos mis profesores por compartir su experiencia y conocimiento sin limitaciones.*

*Agradezco infinitamente al Dr. Irineo L. López Cruz, quien dirigió satisfactoriamente esta tesis, por su apoyo moral, sabios consejos y por compartir su experiencia profesional lograda a través de años de esfuerzo y dedicación. Además, por su calidad humana y por ser una persona que sabe actuar con paciencia, cautela y comprensión, así como conocer sabiamente los límites y potenciales de cada uno de sus alumnos a los cuales dirige. Es todo un profesional envidiable.*

*Al M en I. Agustín Ruíz García por su apoyo incondicional en momentos críticos, él siempre estuvo disponible para realizar sugerencias y facilitarme su vasto conocimiento en programación. Además por su gran calidad humana.*

*A todas aquellas personas que directa o indirectamente hicieron posible la culminación de mis estudios.*

# Contenido

<b>Lista de Cuadros</b>	<b>vii</b>
<b>Lista de Figuras</b>	<b>viii</b>
<b>1. Introducción General</b>	<b>1</b>
1.1. Procedimiento de modelación en sistemas agrícolas . . . . .	2
1.1.1. Modelado de cultivos en campo abierto . . . . .	6
1.1.2. Modelado del sistema invernadero . . . . .	8
1.2. Control y optimización de sistemas agrícolas . . . . .	9
1.3. Estimación de parámetros . . . . .	13
1.4. Inteligencia Artificial y computacional . . . . .	15
1.5. Objetivos de la tesis . . . . .	16
1.6. Esbozo de la tesis . . . . .	17
<b>2. Técnicas de inteligencia artificial en modelado y control del ambiente de inver-</b>	
<b>naderos: Estado del Arte</b>	<b>20</b>
2.1. Introducción . . . . .	21
2.2. Técnicas de inteligencia artificial . . . . .	23
2.2.1. Lógica difusa o sistemas difusos . . . . .	23
2.2.2. Redes Neuronales Artificiales . . . . .	27
2.2.3. Algoritmos evolutivos . . . . .	31
2.2.4. Algoritmos Bio-inspirados . . . . .	42
2.2.5. Algoritmos Híbridos . . . . .	50
2.2.6. Conclusiones . . . . .	51
<b>3. Estimación de parámetros para un modelo de crecimiento de cultivos usando</b>	
<b>algoritmos evolutivos y bio-inspirados. Parte I</b>	<b>53</b>
3.1. Introducción . . . . .	54
3.2. Materiales y métodos . . . . .	55
3.2.1. Algoritmo de evolución diferencial . . . . .	55
3.2.2. Algoritmo Cuco clásico (CS) y Modificado (MCS) . . . . .	57
3.2.3. Modelo para crecimiento potencial de cultivos SUCROS . . . . .	59
3.2.4. Función a optimizar y características de los algoritmos . . . . .	60
3.3. Resultados y discusión . . . . .	61
3.4. Conclusiones . . . . .	67

<b>4. Estimación de parámetros para un modelo de crecimiento de cultivos usando algoritmos evolutivos y bio-inspirados. Parte II</b>	<b>68</b>
4.1. Introducción . . . . .	69
4.2. Materiales y métodos . . . . .	71
4.2.1. El modelo SUCROS para crecimiento potencial . . . . .	71
4.2.2. Descripción del sitio experimental . . . . .	72
4.2.3. Descripción del problema a optimizar . . . . .	74
4.2.4. Ajustes e implementación de los algoritmos . . . . .	75
4.3. Resultados y discusión . . . . .	77
4.4. Conclusión . . . . .	88
<b>5. Discusión general</b>	<b>89</b>
<b>6. Conclusiones</b>	<b>94</b>
<b>Referencias bibliográficas</b>	<b>95</b>

# Lista de Cuadros

1.	Aplicaciones de la Lógica Difusa al modelado y control del clima de invernaderos. . . . .	28
2.	Aplicaciones de las ANN's al modelado y control del clima de invernaderos.	32
3.	Algoritmos de evolución diferencial (DE) implementados en Matlab. . . . .	37
4.	Aplicaciones de los Algoritmos Evolutivos al modelado y control del clima de invernaderos. . . . .	43
5.	Aplicaciones de los algoritmos híbridos al modelado y control del clima de invernaderos. . . . .	51
1.	Parámetros del modelo SUCROS para crecimiento potencial de cultivos. . .	62
2.	Valor mínimo de la función objetivo ( $f(p)$ ) y número de veces que fue evaluada la función objetivo (NEF) (el número entre paréntesis es la desviación estandar). . . . .	63
3.	Valores promedio estimados de los parámetros del modelo SUCROS mediante los tres algoritmos de optimización global y resultados del ANDEVA (*Diferencia significativa $\alpha = 0,05$ ). . . . .	64
4.	Resultados estadísticos de las cinco variables de salida del modelo SUCROS calibrado con tres métodos de optimización global. . . . .	66
1.	Ecuaciones principales del modelo para crecimiento potencial de cultivos SUCROS. . . . .	72
2.	Parámetros del modelo SUCROS para crecimiento potencial de cultivos. . .	73
3.	Valor mínimo promedio ( $f(p)$ ) y número de veces que fue evaluada la función objetivo (NEF) (Los números entre paréntesis son la desviación estandar). . . . .	78
4.	Valores promedio de los parámetros estimados del modelo SUCROS (Números entre paréntesis son la desviación estandar). . . . .	79
5.	Resultados estadísticos después de la calibración usando AE's y BI's. . . .	81
6.	ANDEVA de los cinco algoritmos evaluados sobre el modelo SUCROS (*Diferencias significativas $\alpha = 0,05$ ). . . . .	82

# Lista de Figuras

1.	Proceso de modelado de sistemas (adaptado de van Straten, 2012) . . . . .	5
2.	Diagrama relacional que muestra los principales componentes del modelo SUCROS. . . . .	7
3.	Principales variables en el modelado del clima de un invernadero. . . . .	9
4.	Estructura del sistema de control (adaptado de Cunha y Oliveira, 2003) . . .	11
5.	Esquema de control óptimo del clima de invernaderos . . . . .	13
6.	Esquema del procedimiento de estimación de parámetros usando un método de búsqueda (van Straten, 2012). . . . .	14
1.	(a) Función característica de un conjunto duro y (b) función de membresía de un conjunto difuso ( $b > d > a > c > 0$ ) (adaptado de Huang, 2010). . .	24
2.	Principales componentes de un sistema difuso (adaptado de Schmid, 2005). . .	26
3.	Estructura de un controlador difuso . . . . .	27
4.	Modelo de una red neuronal artificial. . . . .	29
5.	Modelo no lineal de una neurona. . . . .	29
6.	Funciones de activación: a) Escalón, b) Lineal a trozos, c) Sigmoidea. . . .	30
7.	Diferentes arquitecturas de ANNs (Notación abreviada). (a) Una sola capa, (b) Multicapa, (c) Recurrente (Adaptado de Hagan, <i>et al</i> , 1996). . . . .	31
8.	Proceso básico de un GA (adaptado de Huang <i>et al.</i> , 2010). . . . .	34
9.	Ejemplo de programación genética básica . . . . .	41
1.	Pseudocódigo para el algoritmo DE con cruce binomial . . . . .	56
2.	Pseudocódigo del algoritmo Búsqueda Cuco clásico . . . . .	59
3.	Pseudocódigo del algoritmo Búsqueda Cuco Modificado . . . . .	61
4.	Comparación de estimaciones del modelo SUCROS calibrado con tres métodos de optimización global para las cinco variables de salida consideradas. . .	65
1.	Variables de entrada del modelo SUCROS. . . . .	74
2.	Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca total del tomate de cáscara. . .	83
3.	Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca de tallos del tomate de cáscara. . .	84
4.	Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca de frutos del tomate de cáscara. . .	85
5.	Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca de hojas del tomate de cáscara. . .	86

6.	Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para el índice de área foliar del cultivo tomate de cáscara. . . . .	87
1.	Clasificación de los algoritmos bio-inspirados (adaptado de Fister <i>et al.</i> , 2013) . . . . .	92

# TÉCNICAS DE INTELIGENCIA ARTIFICIAL EN MODELADO Y CONTROL DEL CLIMA EN INVERNADEROS

Elmer César Trejo Zúñiga<sup>a</sup>  
Irineo L. López Cruz<sup>b</sup>

En esta tesis, se analizan algunas Técnicas de Inteligencia Artificial (TIA) aplicadas principalmente al modelado y control del ambiente de invernaderos. El modelo para crecimiento potencial de cultivos SUCROS (a Simple and Universal CROP growth Simulator) fue calibrado usando algoritmos evolutivos y bio-inspirados. Las TIA, tales como: Lógica difusa (FL), Redes Neuronales Artificiales (ANN), Algoritmos Evolutivos (EA), Algoritmos Bio-inspirados y Algoritmos Híbridos, han cobrado una importancia relevante dentro de la investigación científica, debido a su capacidad para dar solución a problemas complejos y no-lineales que los métodos convencionales no pueden resolver. En la última década se ha dado énfasis a la aplicación de estas técnicas para el desarrollo de algoritmos inteligentes que permitan identificar, pronosticar, optimizar y controlar el clima para cultivos en ambientes controlados (invernaderos) y generar condiciones climáticas cercanas a las óptimas para el crecimiento y desarrollo de los cultivos y obtener así un mayor rendimiento y calidad de productos. De acuerdo a lo antes dicho, se presenta un resumen de algunas aplicaciones de la inteligencia artificial en modelado y control del ambiente en invernaderos poniendo énfasis en los resultados obtenidos por los investigadores. La calibración de modelos dinámicos para crecimiento y desarrollo de cultivos con muchos parámetros, genera estimaciones imprecisas de estos y predicciones erróneas del modelo cuando se usa estimación por mínimos cuadrados o máxima verosimilitud. Por lo tanto, es necesario el uso de métodos globales de optimización para resolver este problema. Para mostrar lo anterior, se realizó una comparación del desempeño entre dos algoritmos evolutivos: Evolución diferencial (DE) y Estrategias evolutivas con matriz de covarianza adaptable (CMA-ES) y cuatro bio-inspirados: Búsqueda Cuco (CS), Búsqueda Cuco Modificado (MCS), Optimización por enjambre de partículas (PSO) y Colonia artificial de abejas (ABC). El problema prueba consistió en estimar los 25 parámetros del modelo para crecimiento potencial de cultivos SUCROS (a Simple and Universal CROP growth Simulator). Se usaron datos obtenidos de un experimento de crecimiento de un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. ex Horm.) llevado a cabo en Chapingo, México. El objetivo fue determinar qué algoritmo genera valores para los parámetros del modelo que permitan lograr las predicciones más precisas. Se realizó un análisis de varianza (ANDEVA) para evaluar estadísticamente la eficiencia y efectividad de los algoritmos propuestos. Las variables de entrada del modelo fueron la temperatura mínima diaria, temperatura máxima diaria y la radiación global. Siendo las variables de salida (estimadas) el peso seco total, peso seco de tallos, peso seco de frutos, peso seco de hojas e índice de área foliar. Los resultados mostraron un mejor desempeño del algoritmo DE estándar (DE/rand/1/bin) en términos de eficiencia y eficacia para converger a una solución óptima. Los algoritmos bio-inspirados mostraron un buen desempeño, debido a esto, son confiables y pueden ser aplicados en el proceso de estimación de parámetros de modelos de cultivos.

**Palabras clave:** Invernaderos, computación evolutiva, algoritmos bio-inspirados, estimación de parámetros.

---

<sup>a</sup>Tesista

<sup>b</sup>Director

# ARTIFICIAL INTELLIGENCE TECHNIQUES IN MODELLING AND CONTROL OF GREENHOUSE CLIMATE

Elmer César Trejo Zúñiga<sup>a</sup>  
Irineo L. López Cruz<sup>b</sup>

In this thesis, some Artificial Intelligence techniques (AIT) mainly applied to modeling and control of greenhouse environment are analyzed. The potential crop growth SUCROS model (a Simple and Universal Crop Growth Simulator) was calibrated using evolutionary and bio-inspired algorithms. The AIT, such as Fuzzy Logic (FL), Artificial Neural Networks (ANN), Evolutionary Algorithms (EA), Bio-inspired Algorithms and Hybrid Algorithms, have gained relevance in scientific research, due to their ability to solve complex and non-linear problems that the conventional methods cannot solve. The research undertaken during the last decade has emphasized the application of these techniques for the development of intelligent algorithms applied to identify, to predict, to optimize and to control the climate for crops in controlled environments (greenhouses) and to generate near optimal climatic conditions in indoor bio-systems for crop growth and development, and to obtain higher yields and product quality. According to the above, a summary of some applications of artificial intelligence in modeling and control of greenhouse environment with emphasis on the results obtained by the researchers is presented. The parameter estimation (calibration) of dynamic models for crop growth and development with parameter ranges commonly generates imprecise estimations and erroneous predictions of the model when estimation is used by the least squares or maximum likelihood procedures. Therefore, the use of global optimization methods for solving this problem is investigated. To show this, a comparison was carried out of the performance between two evolutionary algorithms: Differential Evolution (DE) and Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES) and four bio-inspired algorithms: Cuckoo Search (CS), Modified Cuckoo Search (MCS), Particle Swarm Optimization (PSO) and artificial bee colony (ABC). The test problem consisted of the estimation of the 25 parameters of the model for potential crop growth SUCROS (a Simple and Universal Crop growth Simulator). The data used was obtained from an experiment of growth of a husk tomato crop (*Physalis ixocarpa* Brot. Ex Horm.) carried out at Chapingo, Mexico. The aim was to determine which algorithm generates values for the parameters of the model that makes possible to obtain the most accurate predictions. An analysis of variance (ANOVA) was performed to evaluate statistically the efficiency and effectiveness of the proposed algorithms. The input variables of the model were the minimum daily temperature, maximum daily temperature and global radiation. Being the output variables (estimated) total dry weight, stem dry weight, fruit dry weight, leaf dry weight and leaf area index. Results showed a better performance of the standard DE algorithm (DE/rand/1/bin) in terms of efficiency and effectiveness to converge to an optimum solution. The bio-inspired algorithms showed good performance; therefore, they are reliable and can be applied in the estimation of parameters of crop models.

**Keywords:** Greenhouses, evolutionary computation, Bio-inspired algorithms, Parameter estimation.

---

<sup>a</sup>Thesis author

<sup>b</sup>Thesis director

---

# CAPÍTULO 1

---

## INTRODUCCIÓN GENERAL

En las últimas décadas, la comunidad científica ha desarrollado y usado modelos matemáticos de crecimiento y desarrollo de cultivos que constituyen herramientas fundamentales para la investigación. La modelación matemática consiste en representar, mediante objetos matemáticos, las diferentes entidades que intervienen en cierto proceso físico, por tanto, describen una parte de la realidad y lo más importante para la ciencia es el estudio de sus propiedades. De acuerdo con Thornley y Johnson (2000), los modelos matemáticos pueden contribuir a ampliar el conocimiento y ayudar en aplicaciones prácticas. Los modelos explicativos simulan una característica (i.e. crecimiento) en términos de procesos fisiológicos. Los modelos descriptivos muestran la existencia de relaciones entre los componentes del sistema. Los modelos pueden ser usados para probar hipótesis y pueden ser aplicados en condiciones diferentes. También son útiles para formular nuevas hipótesis que requieren pruebas experimentales. En general, se reconocen los enfoques teórico y empírico como fundamentales para generar modelos de simulación de crecimiento de cultivos (Poluekov y Topaj, 2001, Thornley y Johnson, 2000). Los modelos empíricos son descripciones directas de datos y proporcionan relaciones observables entre las variables de un sistema sin mediar alguna explicación de los mecanismos subyacentes, por lo tanto son necesarios para describir y resumir datos. Los modelos empíricos más conocidos son los modelos de regresión simple y múltiple o modelos no lineales conocidos como ajuste de curvas. Otros modelos empíricos son los modelos autoregresivos con extradas externas (ARX), las redes neuronales artificiales, los modelos basados en Lógica difusa y los modelos Neuro-difusos. Los modelos teóricos o mecanicistas son modelos explicativos de un sistema. Generalmente, se define un conjunto de ecuaciones diferenciales ordinarias para describir el comportamiento de las variables de estado del sistema, las variables que representan las propiedades o atributos del sistema considerado. Los modelos mecanicistas son más apropiados para expresar hipótesis matemáticamente y proporcionan así una descripción cuantitativa y una expli-

cación de los procesos más importantes que ocurren en un sistema de cultivo (Thornley y Johnson, 2000).

## **1.1. Procedimiento de modelación en sistemas agrícolas**

De acuerdo con Rabbinge y de Wit (1989), para generar un modelo matemático de un sistema de cultivo es necesario aplicar el proceso que se describe a continuación:

1. Formular los objetivos del modelo.
2. Definir los límites del sistema.
3. Especificar conceptualmente el sistema (variables de estado, parámetros, variables de entrada, variables de salida, tasas de cambio).
4. Determinar las relaciones entre las variables de estado, tasas de cambio, variables de entrada mediante experimentos, conocimiento previo o estimación.
5. Derivar la estructura del modelo.
6. Verificar el modelo.
7. Evaluación del desempeño del modelo, partes o como un todo usando experimentos independientes.
8. Realizar un análisis de sensibilidad del modelo.
9. Simplificar el modelo de ser posible.
10. Formular reglas de decisión o modelos de predicción para aplicaciones prácticas.

Las etapas 1-3 se denominan fase preliminar o conceptual del desarrollo de un modelo, las etapas 4-6 se llaman fase de comprensión del modelo y las etapas 7-10 se llaman fase de resumen del modelo.

De acuerdo con la teoría de sistemas dinámicos el procedimiento general de modelación de un sistema (Rabbinge y de Wit, 1989; Ljung y Glad, 1994; van Straten, 2012) empieza por la definición de un problema y su análisis (Figura 1). Se entiende un sistema como una porción de la realidad con elementos interconectados y un modelo matemático como

una representación simplificada de un sistema. Posteriormente se elabora un modelo conceptual del sistema usando conocimiento teórico existente, así como tomando en cuenta los objetivos del proceso de modelación. Una representación conceptual de un modelo, encontrada frecuentemente en la literatura son los diagramas relacionales que permiten la generación de un modelo cualitativo del sistema. Para la generación del modelo cuantitativo de un sistema existen dos caminos: los modelos de caja negra y los modelos transparentes o mecanicistas. En el primer enfoque se diseñan y se llevan a cabo experimentos reales para poder generar una modelación empírica basada en mediciones de variables denominadas entradas y salidas del sistema. Esta modelación se denomina identificación del sistema. En el caso de los modelos mecanicistas, tomando como base el modelo conceptual, los objetivos del proceso de modelación, teorías y datos existentes en la literatura, así como las restricciones existentes (datos, tiempo, etc.) se postula el conjunto de ecuaciones dinámicas también llamada la estructura del modelo (Thornley y France, 2007). Generalmente, para esto se consideran principios primarios y conocimiento existente de los procesos más importantes que ocurren en el sistema. Una vez que se ha generado el conjunto de ecuaciones diferenciales ordinarias no lineales, se procede a obtener simulaciones o sus soluciones numéricas. Posteriormente se puede llevar a cabo un análisis de sensibilidad del modelo con la finalidad de estudiar cómo sus condiciones iniciales, las variables de entrada y los parámetros afectan el comportamiento de las variables de estado y variables de salida. En general existen dos métodos de análisis de sensibilidad de modelos matemáticos: métodos locales y métodos globales. Cuando este es local, se necesita plantear y resolver numéricamente ecuaciones diferenciales denominadas ecuaciones de sensibilidad. Otro enfoque consiste en utilizar simulación Monte Carlo para calcular índices de sensibilidad utilizando funciones de densidad de probabilidades para las variables analizadas. El análisis de identificabilidad permite conocer si el vector de parámetros del modelo dinámico puede ser determinado en forma única a partir de las variables de entrada y variables de salida. Existen varios métodos para determinar la identificabilidad de un modelo, pero varios se basan en los resultados de análisis de sensibilidad llevado a cabo en la fase previa. La siguiente etapa es la estimación de parámetros o calibración del modelo usando tanto información existente en la literatura como datos provenientes de experimentos, así como también los resultados obtenidos del análisis de sensibilidad. Esta etapa consiste en aproximar lo más posible las predicciones de las variables de estado y variables de salida del modelo a mediciones obtenidas del sistema real. Tanto en los modelos mecanicistas como

en los empíricos se pueden usar varios conjuntos de datos para arribar a un modelo calibrado en forma satisfactoria. Generalmente, la estimación de parámetros se lleva a cabo planteando y resolviendo un problema de optimización. Lo anterior implica usar algoritmos de búsqueda local como mínimos cuadrados o máxima verosimilitud o algoritmos globales tales como procedimientos estocásticos, algoritmos evolutivos y bio-inspirados. En caso de que el ajuste logrado entre las predicciones y las mediciones no sea aceptable, de acuerdo a criterios estadísticos, se puede retornar a alguna etapa anterior del procedimiento de modelación, de lo contrario se continua con la etapa de evaluación (validación) del modelo. La evaluación del modelo consiste en usar conjuntos de datos independientes para estudiar el comportamiento del modelo, usando los valores de sus parámetros obtenidos durante la fase de calibración. Nuevamente, para poder decidir si el modelo es satisfactorio se usan normalmente varios criterios estadísticos (Wallach, 2006) tales como el sesgo, el cuadrado medio del error, o su raíz cuadrada, el error absoluto medio, etc. Si es necesario, se puede regresar a etapas anteriores del proceso de modelación. Una vez que se obtiene un modelo validado se puede usar este para alguna aplicación o antes llevar a cabo su análisis de incertidumbre. El análisis de incertidumbre permite evaluar cuantitativamente la variabilidad de los parámetros y variables de entrada del modelo, mediante la deducción de distribuciones de incertidumbre para cada variable que el modelo predice.

Un modelo matemático es usado para representar aspectos esenciales de un sistema o proceso de tal manera que las interacciones entre las variables involucradas (entrada-salida) puedan ser aproximadas de manera eficiente. Existen dos tipos de modelos: el estático (obtenido a través de un balance térmico) y el dinámico (obtenido a través de un balance energético) (Lafont y Balmat, 2004). Un método práctico utilizado en modelado es la implementación de un proceso de identificación no lineal en base a datos experimentales postulando primeramente un modelo parametrizado (estructura del modelo), el mejor modelo puede ser encontrado optimizando los parámetros de acuerdo a ciertos criterios estipulados (conjunto de observaciones entrada-salida). La estructura del modelo es seleccionado de un conjunto de estructuras matemáticas generadas en el proceso (ejemplo, el modelo ARX, Auto-regresive with external inputs; ARMAX, Auto-regresive moving average with external inputs) (López-Cruz *et al.*, 2007; Frausto *et al.*, 2003). Este tipo de procesos donde no se incluye conocimiento a priori del funcionamiento interno del sistema es llamado modelado de caja negra (black-box). El objetivo del modelado de caja negra es desarrollar una aproximación de las funciones del sistema desconocido usando un conjunto de datos exper-

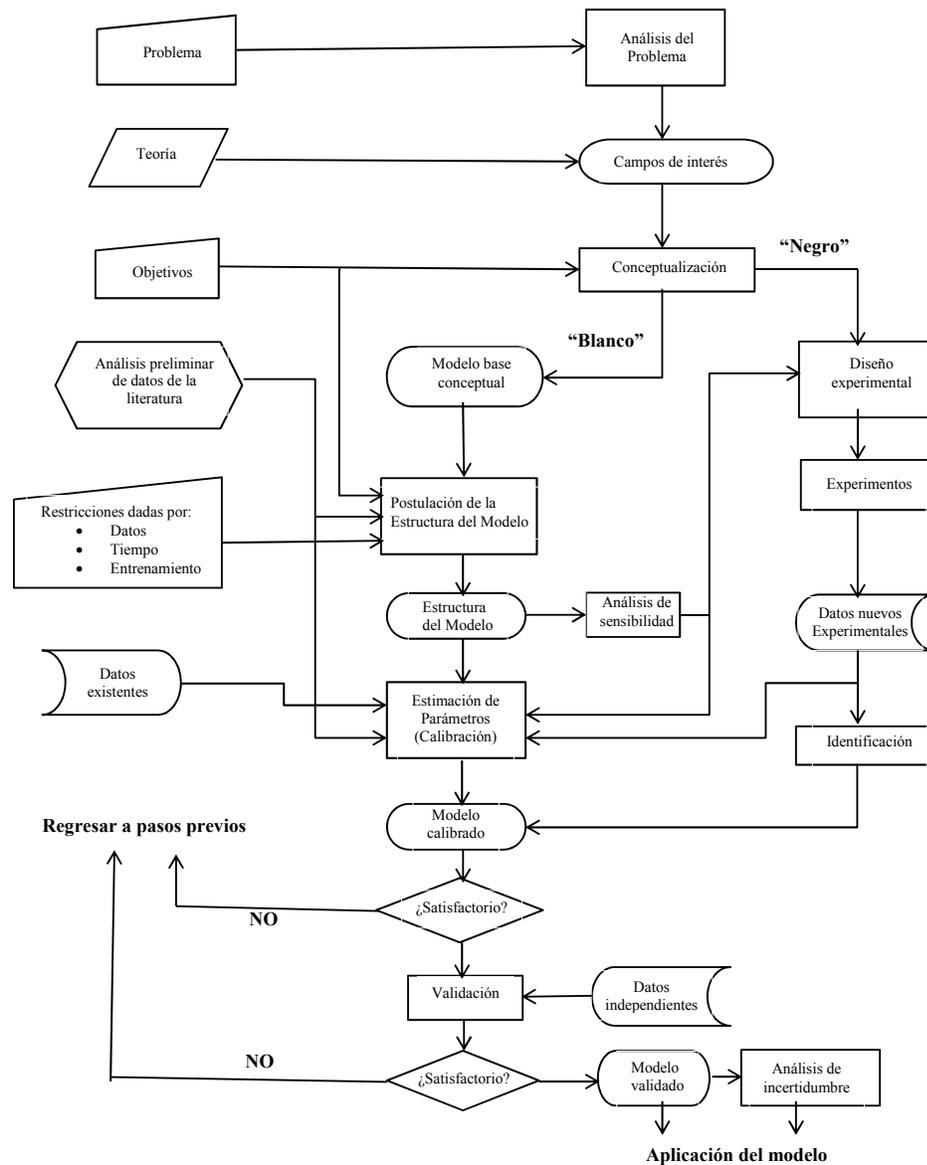


Figura 1: Proceso de modelado de sistemas (adaptado de van Straten, 2012)

imentales. Si existiera la posibilidad de anexas información a priori del sistema, entonces es llamado modelado de caja gris (grey-box). El modelado con Redes neuronales artificiales (ANNs) y Lógica difusa (FL) son ejemplos típicos de modelado de caja negra y gris respectivamente (Tien, 1997; Tien y Van Straten, 1998).

### 1.1.1. Modelado de cultivos en campo abierto

Los modelos matemáticos en la agricultura son herramientas poderosas para describir y comprender sistemas complejos. Estos modelos se han utilizado en la producción de plantas para simular los efectos de los cambios en las características morfológicas y fisiológicas de los cultivos que ayudan en la identificación de ideotipos para diferentes entornos. El modelo de crecimiento y desarrollo de un cultivo en campo abierto puede definirse mediante la ecuación diferencial ordinaria de primer orden no-lineal de la forma:

$$\frac{dx}{dt} = f(x, u, p), \quad x(t_0) = \beta \quad (1)$$

donde  $x \in R^n$ ,  $u \in R^m$ ,  $p \in R^q$  y  $\beta \in R^n$ . El vector de estados ( $x$ ) contiene las variables que caracterizan el cultivo como peso seco, número de frutos, área foliar, etc. El vector de entradas ( $u$ ) contiene variables tales como temperatura,  $CO_2$  y radiación fotosintéticamente activa (PAR). El vector ( $p$ ) representa los parámetros o constantes fisiológicas y físicas del modelo. Dado que en general  $f$  es un vector de funciones no-lineales, el modelo de crecimiento no siempre tiene solución analítica y en esos casos debe ser resuelta mediante integración numérica. Un modelo mecanicista explica el crecimiento de un cultivo sobre la base de procesos subyacentes, tales como la asimilación de  $CO_2$  y la respiración, afectadas por las condiciones ambientales (Goudriaan y van Laar, 1994). A partir de las ecuaciones de estado (1) se pueden calcular otras variables llamadas salidas del sistema ( $y$ ). La ecuación algebraica general no-lineal es entonces:

$$y = g(x, u, p) \quad (2)$$

Algunas variables de salida en cultivos son: Peso seco de tallos, peso seco de hojas, el índice de cosecha o la biomasa total de la planta (incluyendo raíces), etc.

Se han desarrollado varios modelos para cultivos en campo abierto con diferentes objetivos, tanto científicos como prácticos. Pero en general, los modelos de crecimiento han sido desarrollados para formular y probar hipótesis sobre conceptos fisiológicos y agronómicos de los cultivos, para ayudar la enseñanza y para apoyar la toma de decisiones sobre el manejo del sistema agrícola (van Ittersum *et al.*, 2003, Bouman *et al.*, 1996). Se han generado modelos o familias de modelos para diversos cultivos entre los cuales están **CERES** (Ritchie y Otter, 1985; Jones y Kiniry, 1986), **EPIC** (Williams *et al.*, 1989; **SUCROS** (van Keulen *et al.*, 1982; Spitters *et al.*, 1989).

Uno de los modelos teóricos más conocido es **SUCROS** (a Simple and Universal **CRO**p growth **Simulator**), el cual es un modelo de crecimiento y desarrollo potencial para cultivos en campo abierto desarrollado durante las últimas tres décadas (Figura 2). **SUCROS** ha sido usado para estudiar el crecimiento potencial de cultivos tales como trigo, papa, remolacha, maíz, caña de azúcar y girasol. También ha sido la base para el desarrollo de modelos de crecimiento limitado por nutrientes y agua, así como para representar el crecimiento real de diversas especies anuales (arroz, trigo, soya, caña de azúcar, papa) y algunas especies perennes (pastos, pinos, manzana, cacao) (van Ittersum *et al.*, 2003).

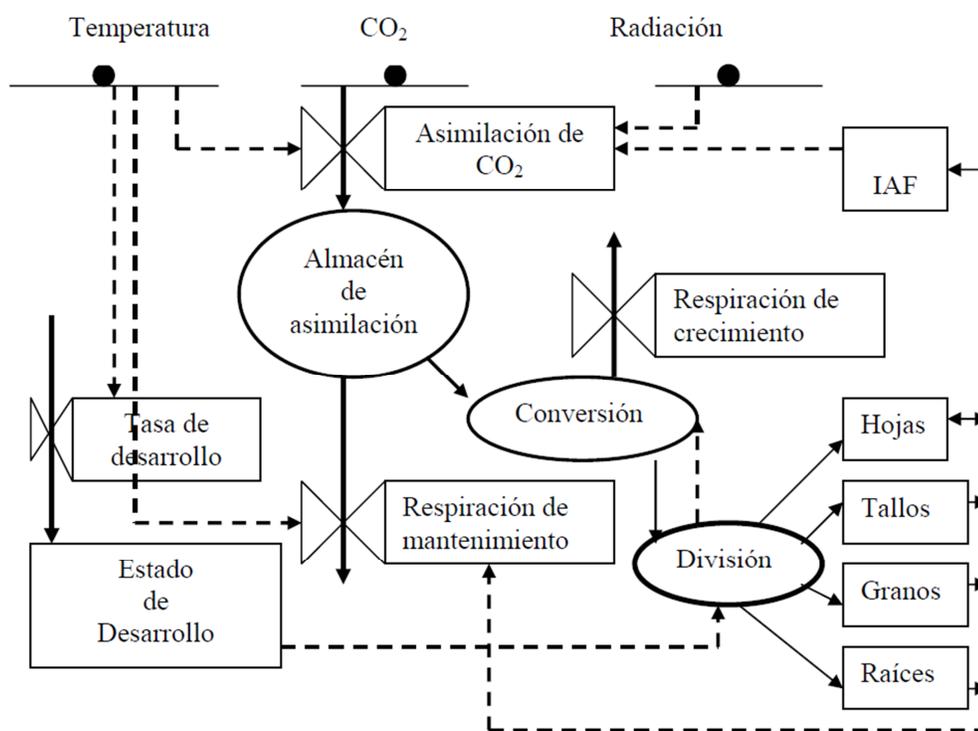


Figura 2: Diagrama relacional que muestra los principales componentes del modelo SUCROS.

El *crecimiento potencial* se define como aquel limitado por la concentración atmosférica de  $CO_2$ , la cantidad de radiación solar, temperatura, duración del día y factores genéticos del cultivo (Penning de Vries *et al.*, 1989, Bouman *et al.*, 1996). En este nivel se consideran procesos básicos de crecimiento del cultivo como fotosíntesis, respiración, crecimiento del tejido y desarrollo. Se utiliza el concepto de crecimiento potencial ya que se supone el

cultivo tiene un suministro adecuado de agua y fertilizantes y además se ignoran los daños por enfermedades y plagas. Un nivel superior considera la *producción alcanzable* por lo que los modelos incluyen las limitantes de agua y fertilizantes. Finalmente, en la *producción real* se intenta tomar en cuenta todos los factores que influyen en el crecimiento del cultivo incluyendo los daños por plagas y enfermedades, así como el manejo.

### 1.1.2. Modelado del sistema invernadero

Existen en la literatura muchos modelos dinámicos para el clima de invernaderos, todos ellos de naturaleza no lineal. Esto se debe a que un invernadero es un sistema dinámico de parámetros distribuidos variantes en el tiempo no-lineales. Por lo tanto, el modelado involucra un amplio conocimiento de factores bióticos y abióticos cuya representación matemática está dada por ecuaciones diferenciales de solución compleja (Bernardo *et al.*, 2006).

El clima dentro de un invernadero se describe por un modelo dinámico representado por la ecuación diferencial

$$\dot{x} = f(x, u, p, t), \quad x(t_b) = \beta \quad (3)$$

donde  $x \in R^n$  son las variables de estado,  $u \in R^m$  son las entradas de control,  $p \in R^q$  son los parámetros invariantes en el tiempo,  $\beta$  es el vector de condiciones iniciales y  $t$  denota el tiempo (van Henten, 1994).

Las principales variables de estado involucradas son: temperatura del aire interior ( $T_i$ ), humedad relativa interior ( $HR_i$ ) y concentración de bióxido de carbono en el interior ( $CO_2$ ). Las perturbaciones consideradas dentro del modelo son radiación solar global ( $R_g$ ), temperatura y humedad del aire exterior ( $T_e$  y  $HR_e$ ), velocidad y dirección del viento ( $V_v$ ). Algunas interacciones entre plantas y suelo también son incluidas ( $T_o$ ). Las variables de control consideradas son apertura de ventilas ( $A_{vent}$ ), inyección de  $CO_2$  ( $CO_{2iny}$ ) y calefacción ( $H$ ) (López-Cruz, 2002; Fourati y Chtourou, 2007; Blasco *et al.*, 2007) (Figura 3).

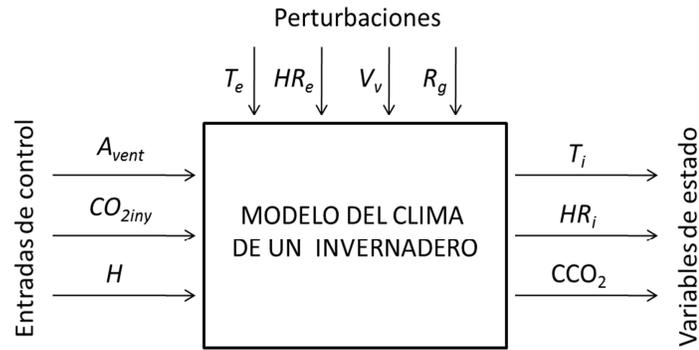


Figura 3: Principales variables en el modelado del clima de un invernadero.

Cada una de las variables está asociada a una ecuación diferencial de la forma:

$$C_g \frac{dT_i}{dt} = K_v(T_o - T_i) + H + K_r(T_o - T_i) + K_s(T_s - T_i) + \eta G - \lambda E + \frac{\lambda}{\varepsilon + 1} M_c \quad (4)$$

$$C_s \frac{dT_s}{dt} = -K_s(T_s - T_i) + K_d(T_d - T_s) \quad (5)$$

$$\frac{V_g}{A_g} \frac{dCC_i}{dt} = Q_v(CC_o - CC_i) + \phi_{inj} + R - \mu P \quad (6)$$

$$\frac{dV_i}{dt} = \frac{A_g}{V_g} [E - Q_v(V_i - V_o) - M_c] \quad (7)$$

donde  $T_i$  es la tasa de cambio de la temperatura del aire interior,  $T_s$  la temperatura del suelo,  $CC_i$  es la concentración de  $CO_2$  y  $V_i$  la humedad absoluta en el interior del invernadero (Tap, 2000).

El modelado del subsistema cultivo dentro de un invernadero es realizado de la misma manera como el modelado de cultivos en campo abierto de acuerdo a la descripción planteada en la sección anterior.

## 1.2. Control y optimización de sistemas agrícolas

Un invernadero es un biosistema complejo con características especiales (componentes biológicos y perturbaciones climáticas) las cuales lo hacen un reto matemático. Un requisito para usar la teoría de control óptimo en el manejo de un invernadero es tener modelos precisos de los componentes mas relevantes del sistema. Sin embargo, los modelos dinámicos

que describen el comportamiento de un cultivo y las variables climáticas en un invernadero generalmente son no-lineales y no-convexas (Chalabi y Zhou, 1996), su comportamiento dinámico es rígido (stiff) ya que existen procesos como la fotosíntesis la cual ocurre muy rápidamente pero el crecimiento (aumento de biomasa) de la planta es un proceso muy lento. Las constantes tiempo de las variables climáticas son pequeñas comparadas con aquellas del cultivo. Además, existe una considerable incertidumbre relacionada con los parámetros de los modelos dinámicos, las condiciones iniciales y el comportamiento de las perturbaciones o el clima externo que afecta al cultivo dentro del invernadero. También el sistema invernadero es afectado por las condiciones del mercado si se desea hacer un manejo óptimo-económico (López- Cruz, 2004; van Henten y Bontsema, 1996).

En general, un invernadero puede ser considerado como un sistema dinámico, por lo tanto uno de los enfoques para estudiar estos sistemas es el de *espacio de estados*. La representación interna de un sistema se lleva a cabo mediante un conjunto de  $n$  medidas, llamadas variables de estado, cuyo cambio en el tiempo se expresa mediante un conjunto de  $n$  ecuaciones diferenciales simultáneas de primer orden, en el dominio del tiempo, denominadas ecuaciones dinámicas o ecuaciones de movimiento del sistema. Las variables de estado representan la cantidad mínima de información que sintetiza el pasado del sistema y es lo que se requiere conocer para predecir su evolución futura, frente a cualquier señal de entrada que se aplique al sistema. Por lo tanto, toda la teoría completa del análisis de sistemas, asociada generalmente con la teoría del control, puede ser aplicada (López-Cruz *et al.*, 2004).

Los principales factores que caracterizan el clima del invernadero son: La radiación, la temperatura, la humedad relativa y la concentración de bióxido de carbono, estos inciden sobre la planta determinando la tasa de fotosíntesis, respiración y transpiración que a su vez modifican el clima del invernadero (Roblero-Jara y Gaytán-Hernández, 2005). Debido al impacto en los principales procesos fisiológicos, el control del clima del invernadero, el control del agua (riego) y el contenido de nutrientes en las raíces se consideran partes fundamentales para manipular el desarrollo y crecimiento del cultivo (van Henten, 2004). Básicamente, un sistema de control para el clima de un invernadero tiene los componentes mostrados en la Figura 4.

El modelo dinámico del biosistema invernadero se puede expresar de la siguiente man-

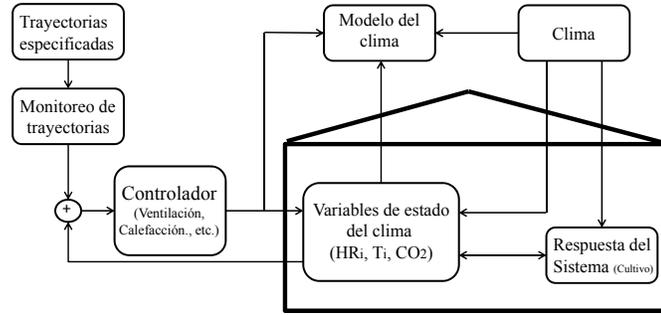


Figura 4: Estructura del sistema de control (adaptado de Cunha y Oliveira, 2003)

era:

$$\dot{x} = f(x, u, v, p, t), \quad x(t_0) = x_0 \quad (8)$$

donde  $x(t) \in R^n$ , es el vector de  $n$  variables de estado,  $x = [T_g \ T_s \ C_i \ V_i \ W_n \ W_s]^T$  de la temperatura del aire ( $T_g$ ), temperatura del suelo ( $T_s$ ), concentración de  $CO_2$  ( $C_i$ ), humedad absoluta ( $V_i$ ), peso seco no estructural ( $W_n$ ) y peso seco estructural ( $W_s$ ) del cultivo dentro del invernadero. El vector de las variables de entrada (variables de control)  $u(t) \in R^m$  de  $m$  variables de control  $u = [H \ r_w \ \phi_i]^T$  de la entrada de calor ( $H$ ), apertura de ventanas ( $r_w$ ) y el flujo de inyección de  $CO_2$  ( $\phi_i$ ). El vector de las perturbaciones  $v(t) \in R^q$  de  $q$  variables  $v = [T_0 \ G \ w \ C_0 \ R_0]^T$  de la temperatura del aire ( $T_0$ ), radiación solar global ( $G$ ), velocidad del viento ( $w$ ), concentración de  $CO_2$  ( $C_0$ ) y humedad ( $R_0$ ) fuera del invernadero (van Henten, 1994, Tap, 2000). El vector de los parámetros  $p \in R^s$  de  $s$  constantes físicas y coeficientes fisiológicos. Las variables de control tienen restricciones en cuanto a su magnitud ya que por ejemplo la cantidad de energía del sistema de calefacción está limitada por la capacidad de un calentador y esta no puede ser negativa ya que su propósito es calentar el invernadero. Las restricciones físicas pueden ser representadas por la ecuación:

$$u_{i,min}(t) \leq u_i(t) \leq u_{i,max}, \quad i = 1, \dots, m \quad (9)$$

donde  $u_{i,min}(t)$  y  $u_{i,max}(t)$  son los límites inferior y superior de las entradas de control. Los variables climáticas dentro del invernadero tienen también restricciones ya que en general se conocen en forma aproximada los valores críticos para cada cultivo. Entonces, es necesario definir restricciones para las variables de estado:

$$x_{i,min}(t) \leq x_i(t) \leq x_{i,max}, \quad i \in I_{xc} \quad (10)$$

donde  $x_{i,min}(t)$  y  $x_{i,max}$  son los límites inferior y superior de las variables de estado y  $I_{xc}$  es el índice asignado a las restricciones de las variables de estado.

En la optimización de sistemas agrícolas es necesario contar con un modelo matemático que represente los principales componentes del sistema (Ecuación 8), restricciones físicas (Ecuación 9) y definir una medida de desempeño (Ecuación 11). Optimizar el clima de un invernadero implica manipular las variables de control de tal manera que el beneficio económico del agricultor sea maximizado. Para resolver este problema, es necesario el empleo de estrategias de control óptimo cuya solución puede ser planteada en forma analítica o numérica (Kirk, 1998). En un invernadero los principales componentes son el cultivo y las variables climáticas internas y el clima externo. Así se pueden identificar varias variables de estado tales como la temperatura del aire, la humedad del aire y la concentración de  $CO_2$ , también la biomasa total y biomasa de frutos o tallos del cultivo. El comportamiento del cultivo y del clima dentro del invernadero es afectado tanto por las variables climáticas externas como también por las acciones de control. Las perturbaciones que más afectan el ambiente invernadero son la radiación solar, la temperatura, velocidad del viento, concentración de  $CO_2$  y humedad fuera del invernadero. Las variables de control, son variables manipulables que permiten modificar el ambiente del invernadero tales como: la apertura de las ventanas, el sistema de calefacción, nebulización, ventilación, refrigeración o el flujo de  $CO_2$  aplicado.

El criterio de desempeño se puede formular como el ingreso económico neto del proceso en el intervalo  $[t_0, t_f]$ , que es el periodo de crecimiento del cultivo en el invernadero, mediante la funcional

$$J(u) = \Phi[x(t_f), t_f] - \int_{t_0}^{t_f} L(x, u, v, p, t) dt \quad (11)$$

donde  $\Phi[x(t_f), t_f]$  representa los beneficios económicos obtenidos de la venta del producto en el momento de la cosecha ( $t_f$ ). El tiempo de cosecha puede definirse como fijo o variable y sujeto a optimización. La función  $L(x, u, v, p, t)$  representa los costos relacionados con los sistemas de control, calefacción, ventilación y suministro de  $CO_2$ , durante el periodo de cultivo.

En resumen, el problema de optimización consiste en encontrar las estrategias de control óptimas en lazo abierto  $u^*(t)$ , que maximizan el criterio de desempeño (Ecuación 11), sujeto a las ecuaciones de estado (Ecuación 8) y a las restricciones (Ecuación 9) y (10), dadas las predicciones de las perturbaciones ( $v(t)$ ) durante todo el periodo de cultivo  $[t_0, t_f]$

(van Henten *et al.*, 1997). Un diagrama de control óptimo del clima de un invernadero se presenta en la Figura 5.

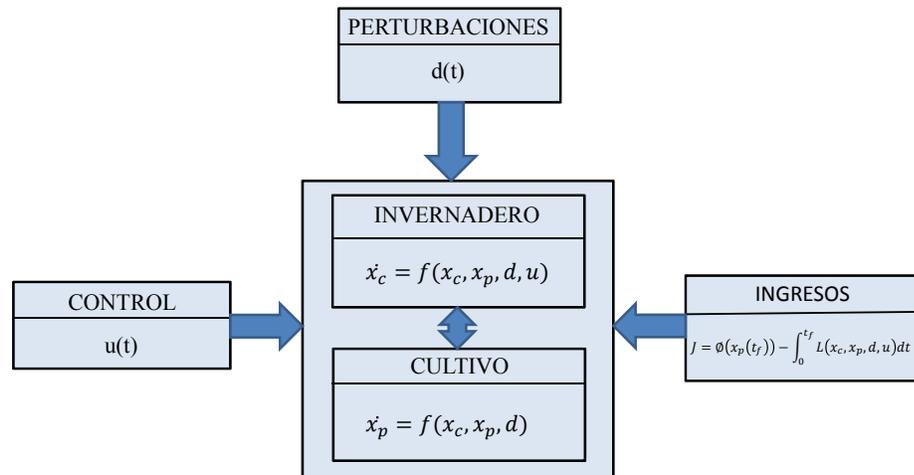


Figura 5: Esquema de control óptimo del clima de invernaderos

### 1.3. Estimación de parámetros

El procedimiento general de estimación de parámetros o calibración de modelos dinámicos es ilustrado en la Figura 6. Como se puede observar, el problema de estimación de parámetros (Ioslovich *et al.*, 2002; van Straten, 2012) consiste en encontrar un conjunto de valores para los parámetros ( $p$ ) del modelo dinámico, de tal forma que minimice un criterio ( $V_p$ ), normalmente el cuadrado medio del error entre mediciones obtenidas del sistema real ( $z_k$ ) y las predicciones del modelo ( $y(k | p)$ ) usando un algoritmo de optimización o método de búsqueda. La estimación de parámetros en un modelo implica la realización de experimentos para poder disponer de mediciones del sistema real. En particular se requiere primero del diseño de un experimento. Su ejecución y la obtención de mediciones de las variables de salida (Ecuación 2) del modelo. Y posteriormente el planteamiento y solución de un problema de optimización. En general el problema de estimación de parámetros en modelos dinámicos puede abordarse mediante dos enfoques: el frecuentista y el Bayesiano (Makowski *et al.*, 2006). Hasta ahora el primer enfoque es más común tanto en agricultura

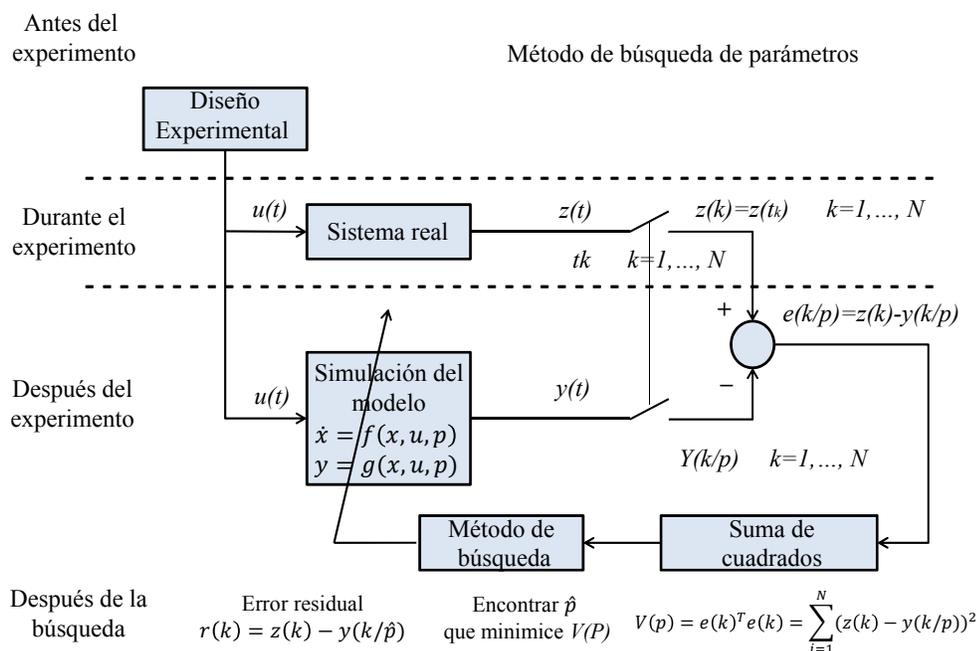


Figura 6: Esquema del procedimiento de estimación de parámetros usando un método de búsqueda (van Straten, 2012).

en campo abierto como en ambientes de agricultura controlada.

Los algoritmos locales de búsqueda empleados son: mínimos cuadrados, máxima verosimilitud, programación cuadrática secuencial, etc. Estos funcionan en forma adecuada cuando los modelos tienen pocos parámetros (menos de 10), pero generan estimaciones imprecisas cuando el modelo tiene muchos parámetros. Además, estos algoritmos pueden converger a soluciones sub-óptimas o mínimos locales. Por estas razones, recientemente se han empezado a utilizar métodos globales de búsqueda. Los algoritmos globales de búsqueda son: algoritmos evolutivos (algoritmos genéticos, estrategias evolutivas, programación evolutiva, evolución diferencial), algoritmos bio-inspirados (Optimización con enjambres de partículas, optimización con colonias de hormigas, optimización con colonias artificiales de abejas, búsqueda Cuco) y métodos estocásticos (búsqueda aleatoria controlada). La estimación de parámetros, en modelos para cultivos en invernadero, mediante métodos locales es más común en la literatura (van Straten *et al.*, 1999; Ioslovich *et al.*, 2002; Linker y Seginer, 2004). Especialmente para la calibración de modelos para lechugas cultivadas en invernadero. Recientemente se usaron algoritmos genéticos, programación evolutiva, y estrategias evolutivas para calibrar un modelo del clima del invernadero (Guzmán-Cruz *et al.*, 2009) y se comparó su desempeño contra el de mínimos cuadrados y programación

cuadrática secuencial. También se han usado algoritmos genéticos para calibrar el modelo reducido TOMGRO (Vázquez-Cruz *et al.*, 2013). En otro estudio se usaron algoritmos genéticos para calibrar un modelo de crecimiento para un cultivo de pepinos cultivados en invernadero (Dai *et al.*, 2009). De acuerdo con Makowski *et al.* (2006) el problema de estimación de parámetros en modelos dinámicos de cultivos y en general de sistemas de agricultura controlada es todavía un campo abierto para la investigación.

## 1.4. Inteligencia Artificial y computacional

La Inteligencia Artificial (IA) es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de problemas. Por lo tanto, se puede definir como “la construcción de mecanismos computacionales para realizar actividades en las cuales se considera que se requiere un grado de inteligencia similar al ser humano” (Ponce-Cruz, 2010). El hombre dentro de su entorno es capaz de combinar la información adquirida por sus sentidos para tomar diferentes acciones y decisiones, además con las diferentes tareas que realiza tiene la capacidad de aprender de sus experiencias. En cambio para las máquinas, esto representa acciones complejas en las cuales deben existir los elementos adecuados para percibir el medio, compararlo con información preestablecida y realizar acciones según las indicaciones programadas. Para lograr esto, se requiere una gran cantidad de procesos, los cuales deben ser implementados adecuadamente con la finalidad de lograr el objetivo establecido. Lo que se intenta en la actualidad es reproducir, de alguna manera, estas interacciones entre percepción del medio (a través del uso de sensores), procesamiento de la información y toma de decisiones (grado de autonomía). Los algoritmos computacionales actuales (evolutivos e inspirados en la naturaleza, por mencionar algunos) son capaces de agregar a un sistema un “grado” de inteligencia debido a sus características de intercambio de información entre individuos (agentes).

En resumen, un algoritmo de Inteligencia Artificial debe ser capaz de almacenar conocimiento, aplicar el conocimiento almacenado para resolver problemas y adquirir nuevo conocimiento a través de la experiencia. Esto es lo que caracteriza a los algoritmos evolutivos y bio-inspirados (Trejo-Zúñiga y López-Cruz, 2011).

## 1.5. Objetivos de la tesis

El objetivo de este estudio es el de aplicar algoritmos heurísticos de búsqueda en la solución de problemas en ingeniería agrícola. Este objetivo se alcanza a través de:

- Hacer un resumen de la literatura existente para conocer el estado actual de las aplicaciones de la inteligencia artificial en ingeniería agrícola, poniendo énfasis en los resultados obtenidos por los investigadores.
- Procesar datos obtenidos del ambiente y del cultivo para calibrar un modelo de crecimiento potencial de cultivos en campo abierto.
- Aplicar técnicas de inteligencia artificial buscando establecer otras alternativas para resolver el problema planteado.

La revisión de literatura realizada arrojó indicios de una casi nula aplicación de los algoritmos bio-inspirados tanto en el modelado como en el control del clima en invernaderos, generándose así un amplio campo de investigación para el estudio, análisis y aplicación de una amplia diversidad de algoritmos inspirados en la naturaleza dentro de la ingeniería agrícola.

Se usaron varios algoritmos evolutivos y bio-inspirados en un proceso de estimación de 25 parámetros que constituye un modelo de crecimiento potencial de cultivos (SUCROS) para detectar cuál de ellos era el más eficiente y eficaz para obtener el menor valor de la función objetivo propuesta. Se realizaron veinte corridas de prueba de cada algoritmo estudiado con mil iteraciones como condición de paro y se obtuvieron valores promedio de los parámetros estimados para ser usados en la calibración del modelo. Posteriormente, se aplicó el modelo calibrado al cultivo del tomate de cáscara (*Physalis ixocarpa* Brot. ex Horm.) usando datos obtenidos de un proceso experimental realizado en Chapingo, Estado de México. Los parámetros asociados a cada heurística fueron ajustados de acuerdo a la literatura y en más de una ocasión fueron ajustados por acierto y error y fijados en un valor donde el algoritmo tuvo el mejor desempeño en el proceso de optimización en cuestión.

Las variables de entrada del modelo fueron la temperatura mínima diaria ( $^{\circ}C$ ), temperatura máxima diaria ( $^{\circ}C$ ) y la radiación global ( $J\ m^{-2}d^{-1}$ ). Siendo las variables de salida el peso seco total ( $g\ m^{-2}$ ), peso seco de tallos ( $g\ m^{-2}$ ), peso seco de frutos ( $g\ m^{-2}$ ), peso seco de hojas ( $g\ m^{-2}$ ) e índice de área foliar ( $m^2\ m^{-2}$ ). Se realizó un análisis estadístico para focalizar las diferencias entre los algoritmos estudiados, resultando el algoritmo de evolución

diferencial el que mejor desempeño tuvo en esta tarea. Un análisis de varianza (ANDEVA) realizado mostró diferencias significativas en casi todos los parámetros estimados por los algoritmos bajo estudio.

En este trabajo no fue considerado un análisis de sensibilidad para detectar los parámetros que más afectan el comportamiento del modelos, tampoco se realizó la validación ni se aplicó un proceso de identificabilidad del modelo.

En general, los algoritmos estudiados en este trabajo presentaron resultados aceptables en el proceso de calibración del modelo SUCROS.

## 1.6. Esbozo de la tesis

El objetivo de este estudio se aborda en tres capítulos. Una descripción de los principales algoritmos basados en inteligencia artificial y sus aplicaciones potenciales al modelado y control del ambiente en invernaderos es presentado en el Capítulo 2. En él, se presentan los conceptos básicos asociados a cada algoritmo, su estructura y modo de operación. Los algoritmos estudiados en esta parte son: Lógica difusa (Zadeh, 1965; Zadeh, 1988; Lee, 2005; Huang *et al.*, 2010), Redes neuronales (Russel y Norvig, 1995; Rojas, 1996; Haykin, 1998), algoritmos evolutivos (Beyer, 2001) y algoritmos bio-inspirados (Yang, 2010). Dentro de los algoritmos evolutivos considerados se pueden mencionar a los algoritmos genéticos (Goldberg, 1989; Holland, 1992; Michalewicz, 1996), programación evolutiva (Fogel, 1962, 1963; Fogel, 2006), evolución diferencial (Storn y Price, 1997; López-Cruz, 2002; López-Cruz *et al.*, 2003a; Price *et al.*, 2005; Chakraborty, 2008), estrategias evolutivas (Beyer, 2001) y programación genética (Willis *et al.*, 1997; Koza, 2010). Los algoritmos bio-inspirados considerados son: Optimización por enjambre de partículas (Kennedy y Eberhart, 1995; Kennedy *et al.*, 2001; Chen *et al.*, 2008), optimización por colonia de abejas (Pham *et al.*, 2006; Karaboga y Basturk, 2007; Karaboga y Akay, 2009; Sundar y Singh, 2010; Karaboga y Ozturk, 2011) y optimización por colonia de hormigas (Dorigo *et al.*, 1996; Dorigo y Stützle, 2004; Jackson y Ratnieks, 2006; Niknam, 2008). También se presenta algunas aplicaciones de los algoritmos híbridos (fusión de dos o más algoritmos inteligentes), siendo considerados en esta caso a los algoritmos Neuro-difusos tales como NUFZY (Tien, 1997; Tien y Van Straten, 1998) y ANFIS (Jang y Chuen-Tsai, 1995). Un resumen de las principales aplicaciones de los algoritmos antes descritos en el modelado y control del clima en invernaderos es presentado en cuadros comparativos

donde se especifica el autor, fecha de publicación y la aplicación específica del algoritmo. La temperatura, humedad relativa y concentración de  $CO_2$  en el interior del biosistema, son las principales variables consideradas para el control en la mayoría de los trabajos.

La estimación de parámetros para un modelo de crecimiento potencial de cultivos en campo abierto conocido como SUCROS (Goudriaan y Van Laar, 1994; van Laar *et al.*, 1997) es el tema principal del Capítulo 3. Debido a que la estimación de parámetros de cualquier modelo dinámico es considerada como un problema de optimización, en esta primera etapa fueron usados los algoritmos de Evolución diferencial (Storn y Price, 1997; López-Cruz, 2002; López-Cruz *et al.*, 2003a; Price *et al.*, 2005; Chakraborty, 2008), Búsqueda cuco clásico (Yang y Deb, 2009) y Búsqueda cuco modificado (Walton *et al.*, 2011) para resolver el problema de optimización en cuestión (minimizar el error de la suma de cuadrados entre las mediciones y predicciones). Un total de 25 parámetros, que constituyen el modelo, fueron estimados. Los parámetros inherentes a cada algoritmo fueron ajustados de acuerdo a datos proporcionados en la literatura. Se presenta un análisis estadístico como herramienta comparativa para demostrar el desempeño de cada uno de los algoritmos estudiados. El modelo calibrado fue aplicado a un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. ex Horm.) usando datos experimentales.

El modelo del Capítulo 3 (SUCROS), fue usado nuevamente en el Capítulo 4. En este Capítulo se desarrolló la calibración del modelo. Se implementaron cuatro algoritmos de optimización global, a saber: Evolución diferencial (DE)(Storn y Price, 1997; López-Cruz, 2002; López-Cruz *et al.*, 2003a; Price *et al.*, 2005; Chakraborty, 2008), Estrategias evolutivas con matriz de covarianza adaptable (CMA-ES) (Hansen y Ostermeier, 1997, 2001; Hansen *et al.*, 2003; Hansen, 2006), Optimización por enjambre de partículas (PSO) (Kennedy y Eberhart, 1995; Kennedy *et al.*, 2001; Chen *et al.*, 2008) y Optimización por colonia de abejas (Pham *et al.*, 2006; Karaboga y Basturk, 2007; Karaboga y Akay, 2009; Sundar y Singh, 2010; Karaboga y Ozturk, 2011), además de un algoritmo de optimización local llamado algoritmo de mínimos cuadrados (LSE) (Gill *et al.*, 1981). Generalmente, LSE usa el gradiente de la función objetivo. Sin embargo, debido a la no-linealidad alta de los modelos de crecimiento y desarrollo de cultivos y la dependencia posible entre los parámetros (epístasis), el problema de optimización puede resultar no convexo, multimodal o sobre-parametrizado. Además, cuando se estiman más de 10 parámetros mediante algoritmos de mínimos cuadrados, se generan varianzas muy altas en los parámetros estimados e imprecisiones en las predicciones del modelo (Makowski *et al.*, 2006) como se puede

apreciar en los resultados obtenidos. Finalmente, se presenta un análisis estadístico como herramienta comparativa para mostrar el desempeño de cada uno de los algoritmos estudiados. El modelo calibrado fue aplicado a un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. ex Horm.) usando datos experimentales.

En la Discusión general (Capítulo 5) se reconsideran los resultados obtenidos en el proceso de estimación de parámetros para el modelo de crecimiento potencial de cultivos SUCROS descritos en los Capítulos 3 y 4. Además, se dan recomendaciones sobre algunos algoritmos bio-inspirados de reciente desarrollo para ser aplicados al proceso de optimización.

---

## CAPÍTULO 2

---

### TÉCNICAS DE INTELIGENCIA ARTIFICIAL EN MODELADO Y CONTROL DEL AMBIENTE DE INVERNADEROS: ESTADO DEL ARTE

Elmer César Trejo Zúñiga<sup>1</sup>, Irineo L. López Cruz<sup>1</sup>

#### Resumen

Las técnicas de Inteligencia Artificial o Computación Suave, tales como: Lógica difusa (FL), Redes Neuronales Artificiales (ANN), Algoritmos Evolutivos (EA), Algoritmos Bio-inspirados y Algoritmos Híbridos (combinación entre ellos) han cobrado una importancia relevante dentro de la investigación científica, debido a su capacidad para dar solución a problemas complejos y no-lineales que los métodos convencionales no pueden resolver. En la última década se ha dado énfasis a la aplicación de estas técnicas para el desarrollo de algoritmos inteligentes que permitan identificar, pronosticar, optimizar y controlar el clima para cultivos en ambientes controlados (invernaderos) y generar condiciones climáticas dentro de los biosistemas, cercanas a las óptimas para el crecimiento y desarrollo de los cultivos y obtener así un mayor rendimiento y calidad de productos. En el presente trabajo se revisan las técnicas de Inteligencia Artificial aplicadas en ingeniería agrícola, principalmente al modelado y control del ambiente de invernaderos.

**Palabras clave adicionales:** Lógica difusa, Redes neuronales, Algoritmos evolutivos, Algoritmos bio-inspirados, Algoritmos híbridos, Invernadero, modelado, control.

---

<sup>1</sup>Postgrado en Ingeniería Agrícola y Uso Integral del Agua, Universidad Autónoma Chapingo, Km. 38.5 Carr. México-Texcoco. C.P. 56230. Chapingo, Estado de México.

\* Capítulo ampliado basado en Trejo-Zúñiga y López-Cruz (2011a,b).

## 2.1. Introducción

Un invernadero es una estructura diseñada especialmente para modificar las condiciones ambientales de un área específica y generar un ambiente adecuado para el cultivo establecido en su interior. El cultivo bajo invernadero es una técnica moderna que ofrece ventajas con respecto al cultivo a campo abierto ya que genera una barrera que protege a las plantas de condiciones externas adversas como heladas, granizo, lluvia, plagas, enfermedades, etc. La cubierta del invernadero permite la implementación de estrategias de control adecuadas que permiten la manipulación de las variables climáticas involucradas (temperatura y humedad del aire, concentración de  $CO_2$ ) para obtener altos rendimientos en la producción, calidad en el fruto y larga vida de anaquel de los productos (Seginer *et al.*, 1994; Latimer y Scoggins, 2003; Cook y Calvin, 2006; Castañeda *et al.*, 2007; van Straten *et al.*, 2011).

Debido a que es necesario un conocimiento completo de la dinámica interna del biosistema y la interacción entre las variables físicas, es de vital importancia la obtención de un modelo matemático que represente apropiadamente su comportamiento. La mayoría de los estudios sobre análisis, modelado y control del clima de invernaderos están basados en el concepto de balance de energía y masa resultando en un conjunto de ecuaciones diferenciales por lo general no lineales. Esos conceptos son básicos para aclarar nociones sobre el control ambiental y detalles de las estrategias a implementar para llegar gradualmente a la optimización económica que es el objetivo final del control del clima de los invernaderos (Hanan, 1997; Pasgianos *et al.*, 2003; López-Cruz *et al.*, 2003a; van Straten *et al.*, 2011).

El advenimiento de nuevos algoritmos inspirados en el comportamiento humano tanto en percepción, aprendizaje y razonamiento (Redes Neuronales), así como en la evolución genética (Algoritmos Evolutivos) y aquellos inspirados en el comportamiento natural de algunas especies animales (Bio-inspirados), ha generado un gran avance en el modelado y control del clima de invernaderos debido a que permiten robustez, docilidad y soluciones a bajo costo de problemas complejos y altamente no lineales con un margen de tolerancia a la imprecisión e incertidumbre (Huang *et al.*, 2010). Dentro de estas estrategias, conocidas en la literatura como Técnicas de Inteligencia Artificial (AI, Artificial Intelligence) o Computación Suave (SC, Soft Computing) (Chen *et al.*, 2008) destacan la Lógica Difusa (FL, Fuzzy Logic), Redes Neuronales Artificiales (ANN, Artificial Neural Networks), Cómputo Evolutivo (EC, Evolutionary Computation), Algoritmos Bio-inspirados (NI, Na-

ture Inspired) y Algoritmos Híbridos.

La FL son sistemas de inferencia que simulan el comportamiento de las decisiones humanas. Con posibilidad de introducir la experiencia de un experto en la elaboración de reglas de decisión, en un esquema si-entonces. Estos sistemas se pueden clasificar en dos categorías: la primera, incluye modelos lingüísticos basados en colección de reglas si-entonces, cuyos antecedentes y consecuentes usan valores difusos (modelo Mamdani) (Mamdani, 1974; Mamdani y Assilian, 1975); y la segunda categoría, está basada en sistemas tipo Takagi-Sugeno y usa una estructura de reglas que tienen antecedentes difusos y consecuentes funcionales (Takagi y Sugeno, 1985; Sugeno, 1985; Takagi y Hayashi, 1991). Las ANN son un sistema computacional inspirado en las características de aprendizaje y estructura de la red neuronal biológica (Huang et al., 2010). Las ANN tienen características como: generalización y habilidad para aprender de la experiencia sin requerir un modelo matemático a priori de las características de la señal, adaptación a los cambios de condiciones ambientales y habilidad para procesar datos degradados o incompletos. Sin embargo, las ANN requieren de un gran número de muestras y mucho tiempo de entrenamiento para un aprendizaje adecuado. La relevancia de las redes neuronales en control radica en que se adaptan a los cambios del sistema y de esta manera tienen un comportamiento satisfactorio en procesos variables no lineales (Huang *et al.*, 2010).

Los Algoritmos Genéticos (GAs, Genetic Algorithms) son una de las más conocidas y originales técnicas de resolución de problemas dentro de lo que se ha definido como “Computación Evolutiva” (“Algoritmos Evolutivos”), término que también agrupa a las Estrategias Evolutivas (ES, Evolution Strategies), la Programación Evolutiva (EP, Evolutionary Programming), Evolución Diferencial (DE, Differential Evolution) y Programación Genética (GP, Genetic Programming). En realidad todas estas técnicas son muy parecidas y comparten muchos aspectos ya que un Algoritmo Evolutivo es una técnica de búsqueda de soluciones inspirada en la evolución de los seres vivos, en el que se define una población de soluciones potenciales para un problema. Nuevas soluciones son generadas mediante tres operadores evolutivos llamados mutación, cruzamiento y selección. En cada ciclo o iteración (generación) se seleccionan las soluciones que más se aproximan al objetivo buscado, eliminando soluciones pobres (Beyer, 2001).

La Computación Bio-Inspirada (Bio-inspired Algorithms, Natural computing, Biologically Inspired Computing) se basa en el uso de analogías con sistemas naturales o sociales para la resolución de problemas. Los Algoritmos Bio-inspirados simulan el comportamiento de

sistemas naturales para el diseño de métodos heurísticos y estocásticos de búsqueda, aprendizaje y comportamiento. Esta técnica involucra a la Optimización por Colonia de Hormigas (ACO, Ant Colony Optimization), Optimización por Enjambre de Abejas (BCO, Bee Colony Optimization) y, Optimización por Enjambre de Partículas (PSO, Particle Swarm Optimization).

Finalmente, en los sistemas híbridos se fusionan las ventajas inherentes de cada una de las técnicas involucradas en el proceso de diseño. Como ejemplo podemos citar los sistemas Neuro-Difusos que son la combinación de ANNs y FL lo cual permite una relación simbiótica, que aprovecha el conocimiento de un experto y la capacidad de aprendizaje y eficiencia computacional de la red neuronal, logrando un sistema de decisiones más inteligente (Jang, 1993, 1995; Jang y Chuen-Tsai, 1995; Gray y Kilgour, 2007).

El presente trabajo muestra cómo las estrategias de computación suave o Técnicas de Inteligencia Artificial han sido aplicadas a la ingeniería agrícola, en general para la modelación y en particular en el control del ambiente de los invernaderos.

## 2.2. Técnicas de inteligencia artificial

Un algoritmo de Inteligencia Artificial debe ser capaz de almacenar, aplicar y adquirir nuevo conocimiento para resolver problemas a través de la experiencia. Esto es lo que caracteriza a las Redes neuronales, Lógica Difusa, Computación Evolutiva y a los Algoritmos Bio-inspirados.

### 2.2.1. Lógica difusa o sistemas difusos

La Lógica difusa (FL) es una forma de lógica multi-valuada derivada de la teoría de conjuntos difusos para tratar con razonamiento que es aproximado más que preciso, en contraste con la lógica binaria, si/no, 0/1 (conjuntos duros) (Huang *et al.*, 2010). En general, los conjuntos duros son definidos por funciones características y los conjuntos difusos son caracterizados por funciones de membresía (Zadeh, 1965; Zadeh, 1988; Jang y Sun, 1995; Tanaka, 1997; Lee, 2005).

*Definición 1.-* Sea  $A$  un conjunto duro en el universo  $X$ . Su función característica  $X_A$  puede ser definida por un mapeo  $X_A : X \rightarrow 0, 1$  como:

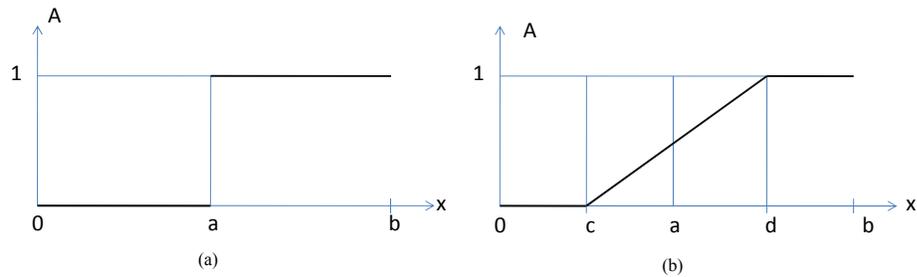


Figura 1: (a) Función característica de un conjunto duro y (b) función de membresía de un conjunto difuso ( $b > d > a > c > 0$ ) (adaptado de Huang, 2010).

$$X_A(x) = \begin{cases} 1 & \text{si } x \in X \\ 0 & \text{si otro caso} \end{cases} \quad (1)$$

Las funciones características son muy poco usadas en aplicaciones de conjuntos duros. Sin embargo, cuando se extiende esta idea a conjuntos difusos, el papel de la función característica se vuelve significativo. Por lo tanto, las funciones de membresía son una extensión de las funciones características (Tanaka, 1997; Lee, 2005).

*Definición 2.-* Un conjunto (subconjunto)  $A$  del universo  $X$  es un conjunto definido por una función de membresía  $\mu_A$  representado por un mapeo  $\mu_A : X \rightarrow \{0, 1\}$ .

Aquí el valor de  $\mu_A(x)$  para el conjunto difuso  $A$  es llamado el valor de membresía o el grado de pertenencia de  $x$  en  $X$ . El valor de membresía representa el grado de  $x$  perteneciente al conjunto difuso  $A$  (Zadeh, 1965; Zadeh, 1988; Jang, 1993; Jang y Sun, 1995). El valor de las funciones características para conjuntos duros son 1 ó 0, pero el valor de membresía de un conjunto difuso puede ser un valor real arbitrario entre 0 y 1 (Figura 1).

Los conjuntos difusos pueden ser definidos de la siguiente manera (Jang y Sun, 1995; Tanaka, 1997; Lee, 2005):

- Expresión discreta (universo finito): Sea el universo  $X = \{x_1, x_2, \dots, x_n\}$ . Entonces un conjunto difuso  $A$  en  $X$  puede ser representado de la siguiente manera:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n = \sum_{i=1}^n \mu_A(x_i)/x_i \quad (2)$$

- *Expresión continua* (universo infinito): Cuando el universo  $X$  es un conjunto infinito, un conjunto difuso  $A$  en  $X$  puede ser representado de la siguiente manera:

$$A = \int_x \mu_A(x_i)/x_i \quad (3)$$

En procesos de modelado y control, los sistemas que no están bien definidos y con incertidumbre pueden ser modelados mediante sistemas de inferencia difusa empleando reglas si-entonces para cuantificar el proceso de conocimiento y razonamiento humano sin emplear análisis cuantitativo preciso. El sistema de inferencia difuso debe incluir las siguientes etapas (Schmid, 2005):

- *Fusificación*: La cual comprende el proceso de transformación de los valores duros en grados de pertenencia para términos lingüísticos de conjuntos difusos. La función de membresía es usada para asociar un grado de pertenencia a cada término lingüístico.
- *Máquina de inferencia difusa*: La sección central de un sistema difuso es la máquina de inferencia difusa ya que combina los datos proporcionados por la etapa de fusificación con la base de reglas y lleva a cabo el proceso de razonamiento difuso.
- Como resultado de la aplicación de los pasos anteriores, se obtiene un conjunto difuso  $\mu_{B'}(u)$ . Es decir, para cada valor posible  $u$  se obtiene un grado de pertenencia que describe en qué medida es razonable utilizar este valor  $u$ . Cuando se usa un sistema difuso como controlador, se requiere transformar esta información difusa en un valor único  $u'$  que realmente se aplique. Esta transformación de un conjunto difuso a un número duro se llama defusificación.

Estos componentes caracterizan a un modelo tipo Mamdani. El modelo difuso tipo Takagi-Sugeno se describe mediante reglas difusas sí-entonces que representan relaciones entradas-salidas locales de un sistema no-lineal, su principal característica es expresar la dinámica local de cada regla difusa por un modelo de sistema lineal (Takagi y Sugeno, 1985). Algunas técnicas de defusificación incluyen el método del máximo, el método del centro de gravedad, método del centroide, método del centro singleton y propiedades de margen del método del centroide. En la Figura 2 se puede apreciar los principales componentes de un sistema difuso.

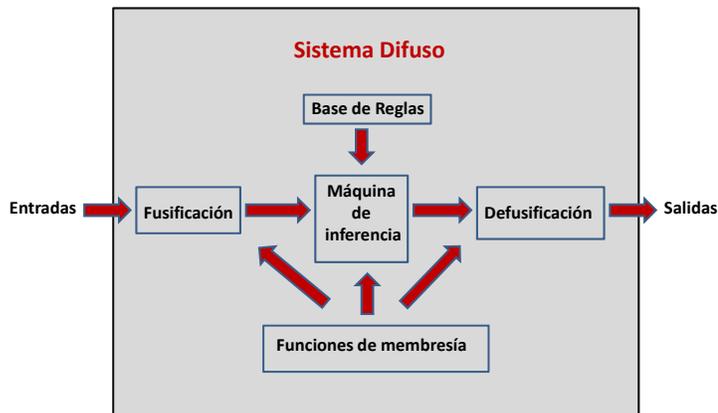


Figura 2: Principales componentes de un sistema difuso (adaptado de Schmid, 2005).

### Estructura de un controlador difuso

En un sistema controlado por lógica difusa, el valor de entrada proveniente de los sensores, es convertido a un valor difuso mediante un proceso llamado *fusificación*. A continuación, ese valor es introducido a un proceso también difuso llamado *mecanismo de Inferencia*, en donde continuamente es evaluado mediante un conjunto de reglas, provenientes generalmente del conocimiento de un experto o elaboradas de manera empírica a través de datos; este proceso produce un valor de salida también difuso, el cual es transformado a una salida real, a través de un proceso llamado *defusificación*. El valor de la salida del proceso de defusificación es utilizado como señal de control o como entrada a un actuador que es en realidad, quien se encarga de ajustar la salida del sistema. Los cambios en el sistema son medidos por los sensores y el proceso se repite nuevamente (Figura 3).

### Técnica de agrupamiento (Clustering)

La técnica de agrupamiento (Clustering) tiene como objetivo encontrar patrones o grupos interesantes en un conjunto de datos dado, de tal manera que tales patrones, estructuras o grupos encontrados sirvan para clasificación, diseño de estrategias, soporte de decisiones, organización de la información, entre otras. Las técnicas de lógica difusa permiten manejar datos en los cuales existe una transición suave entre categorías distintas, por lo que algunos datos pueden tener propiedades de clases diferentes, estando parcialmente en más de un grupo con un grado específico de pertenencia.

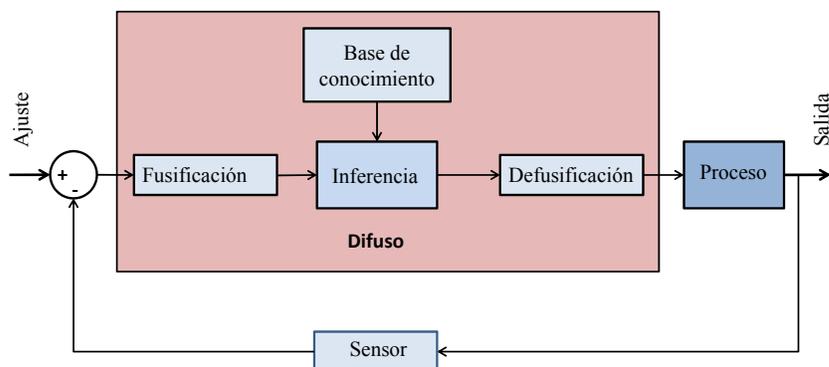


Figura 3: Estructura de un controlador difuso

Una de las tareas de la minería de datos (Data mining) es la identificación de grupos o clusters naturales en los conjuntos de datos. Una técnica difusa bastante conocida y que ha cobrado importancia en la tarea de clustering o agrupamiento es el algoritmo Fuzzy C-Means (FCM) el cuál es una extensión difusa del conocido C-Means. El algoritmo FCM asigna a cada dato un valor de pertenencia dentro de cada cluster y por consiguiente un dato específico puede pertenecer parcialmente a más de un cluster.

### Aplicaciones al modelado y control del clima de invernaderos

En el Cuadro 1 se presenta un resumen de algunas aplicaciones de la Lógica Difusa al modelado y control del clima de invernaderos.

#### 2.2.2. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (ANN), emplean una imagen simplificada de la forma en que el cerebro humano procesa la información. Una ANN tiene muchas unidades de procesamiento trabajando en paralelo (neuronas o nodos). Las neuronas artificiales están altamente interconectadas por ligas (sinapsis) con pesos. Las redes tienen una capa de entrada, una capa de salida y cualquier número de capas ocultas. Cada neurona está ligada a todas las neuronas de la próxima capa, como se muestra en la Figura 4 (Haykin, 1998).

Una neurona es una unidad de procesamiento de información que es fundamental para la operación de una red neuronal. La Figura 5 muestra el modelo de una neurona la cual

**Cuadro 1:** Aplicaciones de la Lógica Difusa al modelado y control del clima de invernaderos.

Algoritmo	Aplicación	Referencia
Controlador difuso	Control de temperatura para evitar agrietamiento del tomate.	Hahn (2011).
Sistema tipo Takagi-Sugeno con agrupamiento difuso.	Identificación de la temperatura y la humedad relativa.	Errahmani <i>et al.</i> (2009)
Controlador difuso tipo Mamdani en Matlab comparado con el controlador todo-nada con histéresis y un simple todo-nada.	Estimar la cantidad de agua absorbida por las plantas.	Trabelsi <i>et al.</i> (2007). Kia <i>et al.</i> (2009)
Modelos Takagi-Sugeno compactos y técnica de agrupamiento difuso.	Control de temperatura y humedad relativa.	Errahmani <i>et al.</i> (2009).
Sistema de control automático difuso.	Control de temperatura y humedad relativa.	Carrillo-Reveles <i>et al.</i> (2008).
Controlador P-I y un controlador difuso.	Comparación de eficiencia en el control de temperatura y humedad relativa.	Dumitrascu (2007).
Controlador PI-Difuso con arquitectura descentralizada.	Control de temperatura y humedad relativa en cámaras de crecimiento de cultivos.	Riadi <i>et al.</i> (2007).
Sistema difuso combinado con Proporcional y P-D.	Control de temperatura, humedad relativa y $CO_2$ .	Saridakis <i>et al.</i> (2006)
Sistema con funciones de membresía triangulares y método del centroide.	Estimar temperatura y humedad.	Bernardo <i>et al.</i> (2006)
Controlador difuso tipo Takagi-Sugeno con compensación distribuida paralela.	Controlar la temperatura y humedad relativa.	Nachidi <i>et al.</i> (2006).
Controlador difuso aplicando (FPGA's).	Control de variables climáticas y funcionamiento de actuadores.	Castañeda-Miranda <i>et al.</i> (2006).
Control difuso asociado con control $H_\infty$ .	Control de temperatura y humedad.	Souissi <i>et al.</i> (2005).
Estructura Jerárquica Colaborativa que organiza automáticamente los conjuntos difusos de reglas sí-entonces.	Modelación de la temperatura y humedad.	Salgado y Cunha (2005).
Controlador difuso tipo Mamdani.	Control de humidificación, sombreado, suministro de $CO_2$ , y ventilación.	Sriraman y Mayorga (2004).
Toolbox de FL de Matlab.	Simulación de la temperatura y humedad.	Costa <i>et al.</i> (2004).
Sistema tipo Takagi-Sugeno, optimización de árboles de regresión difusos y modelos difusos iterativos.	Identificación y modelado de la temperatura y humedad relativa.	Lafont y Balmat (2004).
Sistema de control difuso.	Manipular las variables climáticas temperatura y humedad relativa.	Lafont y Balmat (2002).
Sistema difuso adaptativo combinado con un modelo físico.	Modelado de temperatura del aire.	Lanfang <i>et al.</i> (2000).

†FL: Lógica difusa, PI: Proporcional-Integral, P-D: Proporcional-Derivativo, FPGA: Redes de compuertas lógicas programables.

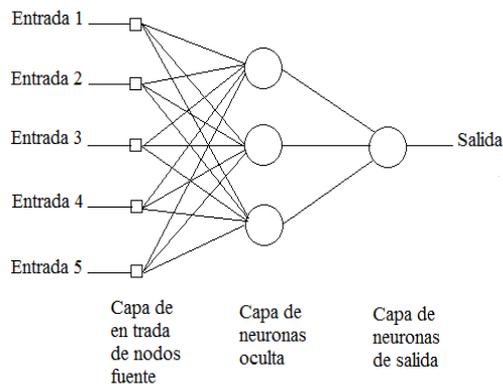


Figura 4: Modelo de una red neuronal artificial.

forma la base para el diseño de ANNs, definida por la siguiente ecuación:

$$y(x) = \varphi \left( \sum_{i=0}^m w_i x_i \right) \quad (4)$$

donde  $(w_0, \dots, w_m)$  son los pesos sinápticos,  $x_i$  es el vector de entradas,  $y(x)$  es la salida,  $\varphi$  es la función de activación, usualmente un escalón (umbral), tangente sigmoidea y lineal a trozos (Ross, 1999).

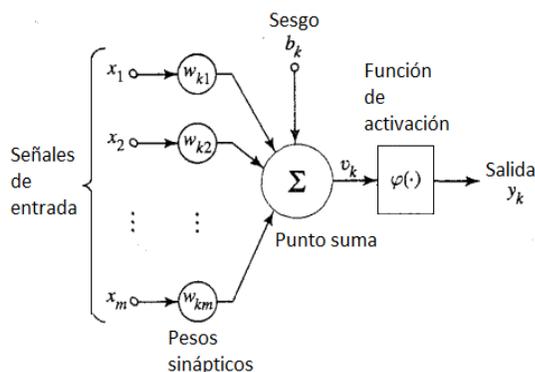


Figura 5: Modelo no lineal de una neurona.

Las funciones de activación denotadas por  $\varphi(v)$  pueden ser definidas de la siguiente manera (Figura 6):

- Función escalón (umbral)

$$\varphi(v) = \begin{cases} 1 & \text{si } v \geq 0 \\ 0 & \text{si } v < 0 \end{cases} \quad (5)$$

- Función lineal a trozos

$$\varphi(v) = \begin{cases} 1 & \text{si } v \geq +\frac{1}{2} \\ v & \text{si } +\frac{1}{2} > v > -\frac{1}{2} \\ 0 & \text{si } v \leq -\frac{1}{2} \end{cases} \quad (6)$$

- Función sigmoidea

$$\varphi(v) = \frac{1}{1 + e^{-av}}, \text{ con } a > 0 \quad (7)$$

Las ANNs más comunes pueden ser con alimentación hacia adelante cuya información es unidireccional (sin ciclos) y realimentadas cuya información fluye en ambas direcciones (con ciclos) (Russell y Norvig, 1995). En general, se pueden distinguir tres clases diferentes de arquitecturas de ANNs (Figura 7):

- Redes con alimentación hacia adelante de una sola capa.
- Redes con alimentación hacia adelante multicapa.
- Redes recurrentes.

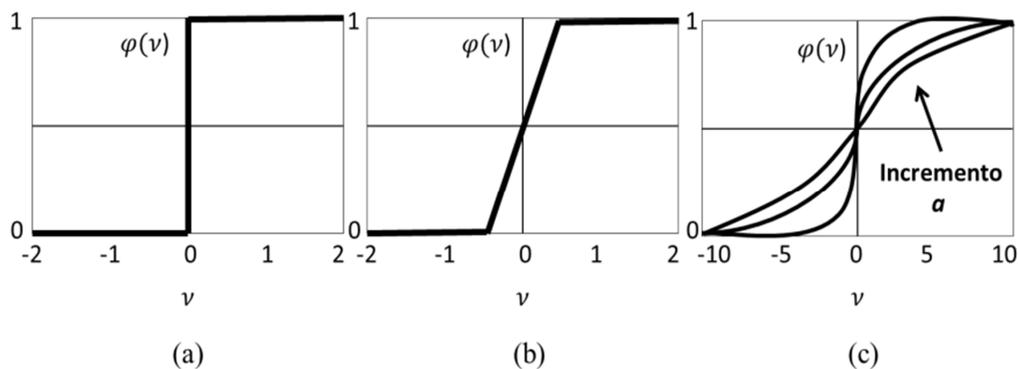


Figura 6: Funciones de activación: a) Escalón, b) Lineal a trozos, c) Sigmoidea.

Un aspecto importante respecto al diseño de ANNs es el aprendizaje, lo cual significa conocer cómo se modifican los valores de los pesos, es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información. Hay dos métodos de aprendizaje importantes que pueden distinguirse (Rojas, 1996): Aprendizaje supervisado y no supervisado.

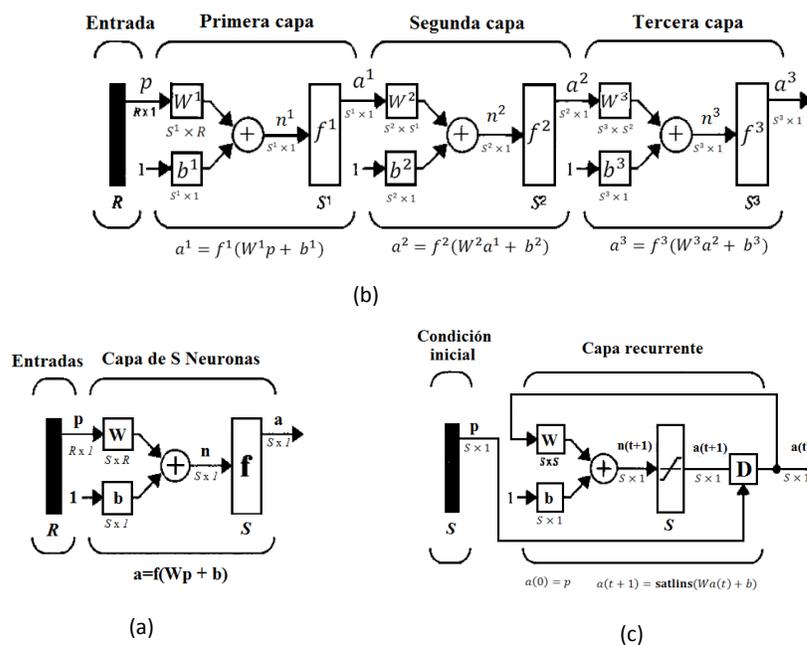


Figura 7: Diferentes arquitecturas de ANNs (Notación abreviada). (a) Una sola capa, (b) Multicapa, (c) Recurrente (Adaptado de Hagan, *et al*, 1996).

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual o si el aprendizaje supone la desconexión de la red, es decir, su inhabilitación hasta que el proceso termine. En el primer caso, se trataría de un aprendizaje en línea (on-line), mientras que el segundo es lo que se conoce como fuera de línea (off-line) (Rojas, 1996).

### Aplicaciones al modelado y control del clima de invernaderos

En el Cuadro 2 se presenta un resumen de algunas aplicaciones de las ANN's al modelado y control del clima de invernaderos.

#### 2.2.3. Algoritmos evolutivos

Un Algoritmo Evolutivo es una técnica de búsqueda de soluciones inspirada en la evolución de los seres vivos, en él se define una población de soluciones potenciales para un problema. Nuevas soluciones son generadas mediante tres operadores evolutivos llamados mutación, cruzamiento y selección. En cada ciclo o iteración (generación) se seleccionan

Cuadro 2: Aplicaciones de las ANN's al modelado y control del clima de invernaderos.

Algoritmo	Aplicación	Referencia
ANN de retro-propagación.	Modelar la humedad del aire.	He y Ma (2010).
Multicapa (MLP) entrenada con el algoritmo de retro-propagación Levenberg-Marquardt.	Para predecir el consumo de energía.	Trejo-Perea <i>et al.</i> (2009).
ANN y FL.	Predicción del rendimiento y caracterización del modo de crecimiento del cultivo de jitomate.	Fitz-Rodríguez y Giacomelli (2009).
Redes neuronales de simple capa y multicapa en tiempo discreto.	Identificación del crecimiento de un cultivo de lechuga.	Córdova y Yu (2009).
ANNs tipo perceptrón multicapa.	Predicción de temperatura, $CO_2$ y fotosíntesis.	Salazar-Moreno <i>et al.</i> (2009).
ANN con el método de análisis de componentes principales.	Identificación de las variables temperatura y humedad.	Pessel y Balmat (2008).
ANN usando RBF parametrizada con el criterio de Levenberg-Marquardt.	Simulación del control del ambiente de un invernadero.	Ferreira y Ruano (2008).
Programación Neuro-dinámica.	Control de temperatura y humedad relativa.	Patño <i>et al.</i> (2008).
ANN Elman.	Control de temperatura y humedad relativa.	Fourati y Chtourou (2007).
Controlador PI+FF y una estructura de ANN basada en RBF.	Identificación de la temperatura de la tubería de un sistema de calefacción.	Chao-gang <i>et al.</i> (2005).
Modelos FIR no lineal y ANN's.	Simular la temperatura interna.	Arahal <i>et al.</i> (2005).
ANN Sigmoidea de caja negra (BBNN) y ANN (RBF) combinadas con un modelo físico.	Predicción de temperatura.	Linker y Seginer (2004).
ANN, modelos auto-regresivos con entradas exógenas (ARX) y modelos físicos.	Identificar la temperatura del aire, humedad relativa y concentración de $CO_2$ .	Cunha (2003).
ANN con un algoritmo de retro-propagación cuasi-Newton para entrenamiento.	Modelado de la conductividad eléctrica y pH de un sistema hidropónico con cultivo de lechuga.	Ferentinos y Albright (2002).
ANN usando RBF parametrizada con el criterio de Levenberg-Marquardt.	Identificación, predicción y modelado de temperatura y humedad relativa.	Ferreira <i>et al.</i> (2000a, 2000b, 2000c, 2002), Ferreira y Ruano (2001a, 2001b, 2002).
ANN combinada con un modelo físico del sistema.	Detección e identificación de fallas en sensores y actuadores que controlan variables climáticas.	Linker <i>et al.</i> (2000).
ANN combinada con un modelo físico del sistema.	Optimización del clima de invernaderos.	Linker y Seginer (1999).
Modelado con ANN	Control óptimo de $CO_2$ en invernaderos.	Linker <i>et al.</i> (1998).
Modelado con ANN	Modelado del clima de invernaderos	Seginer <i>et al.</i> (1994).

†ANN: Red neuronal artificial, PI+FF: Proporcional-Integral más feed-forward, RBF: Función de base radial.

las soluciones que más se aproximan al objetivo buscado, eliminando soluciones pobres (Beyer, 2001). El término Algoritmo Evolutivo también agrupa a las Estrategias Evolutivas (ES, Evolution Strategies), la Programación Evolutiva (EP, Evolutionary Programming), Evolución Diferencial (DE, Differential Evolution) y Programación Genética (GP, Genetic Programming).

### **Algoritmos Genéticos**

Los Algoritmos Genéticos (GAs) reproducen el proceso evolutivo biológico y determinan un valor óptimo conjuntamente con un procedimiento de búsqueda multipunto basado en selección, cruce y mutación genética (Goldberg, 1989; Holland, 1992; Michalewicz, 1996). Inicialmente se genera un conjunto de soluciones aleatorias (cromosomas) para el problema; posteriormente se generan iteraciones (nuevas generaciones), cada una de ellas consiste de la selección de buenas soluciones, cruce y reproducción de las mismas (Beasley, 1993). A través de estas etapas, el algoritmo desarrollará incrementalmente mejores soluciones en cada iteración (generación) como en el caso de la evolución natural. Ocasionalmente se pueden tener mutaciones en ciertas soluciones. El algoritmo básico está constituido de los siguientes pasos (Beasley, 1993; Munakata, 2008) (Figura 8):

- Inicialización de la población (generar un conjunto de soluciones aleatoriamente, individuos).
- Reproducción.
- Recombinación (Cruza, soluciones de la próxima generación).
- Mutación aleatoria (o simplemente mutación).

El cuarto punto es tomado del concepto de mutación natural. Por lo tanto, se espera crear una nueva especie la cual no podría ser creada en el proceso de reproducción y recombinación normales. En GAs, después de un cierto número de iteraciones (generaciones) la mayoría de las soluciones en la población se vuelven semejantes, de tal manera que no se presenta ningún cambio significativo, pero todavía están lejos de la solución óptima (Munakata, 2008). Una selección cuidadosa de la estructura y de los parámetros del GAs puede asegurar la obtención de la solución óptima global después de una cantidad razonable de iteraciones. Por lo tanto, los GAs son herramientas alternativas poderosas para optimización tradicional (Huang *et al.*, 2010).

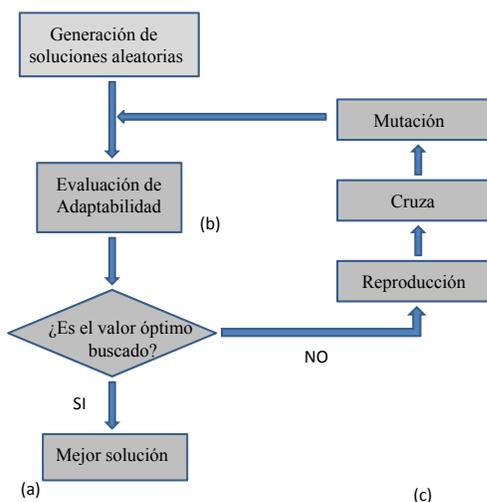


Figura 8: Proceso básico de un GA (adaptado de Huang *et al.*, 2010).

### Evolución Diferencial

El algoritmo de Evolución Diferencial (DE, Differential Evolution) es un método de optimización perteneciente a la categoría de computación evolutiva, aplicado a la resolución de problemas complejos. Al igual que otros algoritmos evolutivos, la DE mantiene una población de soluciones candidatas, las cuales se recombinan y mutan para producir nuevos individuos los cuales serán elegidos de acuerdo al valor de su función de desempeño (López-Cruz *et al.*, 2003a; López-Cruz, 2002). Lo que caracteriza a la DE es el uso de vectores de prueba, los cuales compiten con los individuos de la población actual a fin de sobrevivir (Storn y Price, 1997; Price, 1999; Price *et al.*, 2005; Chakraborty, 2008). El algoritmo asume que las variables del problema a optimizar están codificadas como un vector de números reales. La longitud de estos vectores ( $n$ ) es igual al número de variables del problema, y la población está compuesta de  $NP$  (Número de padres) vectores. Se define un vector  $x_p^g$ , en donde  $p$  es el índice del individuo en la población ( $p = 1 \dots NP$ ) y  $g$  es la generación correspondiente. Cada vector está a su vez compuesto de las variables del problema  $x_{p,m}^g$ , en donde  $m$  es el índice de la variable en el individuo ( $m = 1, \dots, n$ ). Normalmente el dominio de las variables del problema está restringido entre valores mínimos y máximos  $x_m^{min}$  y  $x_m^{max}$ , respectivamente. DE se compone básicamente de cuatro pasos (Storn y Price, 1997; Chakraborty, 2008; Price *et al.*, 2005):

- Inicialización de la población de soluciones

- Mutación
- Recombinación
- Selección

La inicialización se realiza al principio de la ejecución de la búsqueda, los pasos de mutación, recombinación y selección se realizan repetidas veces, hasta que una condición de término sea satisfecha (número de generaciones, tiempo transcurrido, o calidad de solución alcanzada, entre otras).

### *Inicialización*

La población es inicializada (primera generación) aleatoriamente, considerando los valores mínimos y máximos de cada variable:

$$x_{p,m}^1 = x_m^{min} + rand(0, 1) * (x_m^{max} - x_m^{min}), \quad (8)$$

para  $p = 1, \dots, NP$ ,  $m = 1, \dots, n$ , y  $rand(0, 1)$  genera un número aleatorio uniformemente distribuido en el rango  $[0,1]$ .

### *Mutación*

La mutación consiste en la construcción de  $NP$  vectores aleatorios ruidosos, los cuales son creados a partir de tres individuos elegidos al azar, llamados vectores objetivo,  $x_a, x_b, x_c$ . Los vectores aleatorios ruidosos ( $n_p^t$ ) son obtenidos de la siguiente manera:

$$n_p^g = x_c + F(x_a - x_b) \quad (9)$$

con  $p, a, b$  y  $c$  distintos entre sí, y  $p = 1, \dots, NP$ .  $F$  es un parámetro que controla la tasa de mutación, y se encuentra en el rango  $[0,2]$ .

### *Recombinación*

Una vez obtenidos los  $NP$  vectores aleatorios ruidosos, la recombinación se efectúa de manera aleatoria, comparándolos con los vectores originales ( $x_p^g$ ), obteniendo los vectores de prueba ( $t_m^g$ ) de la siguiente manera:

$$t_{p,m}^g = \begin{cases} n_{p,m}^g & \text{si } rand(0, 1) < CR \\ x_{p,m}^g & \text{en otro caso} \end{cases} \quad (10)$$

para  $p = 1, \dots, NP$ ,  $m = 1, \dots, n$ .  $CR$  es un parámetro de probabilidad de cruzamiento que controla la tasa de recombinación. Nótese que la comparación se hace variable por variable, por lo que el vector de prueba será una mezcla de los vectores aleatorios ruidosos y originales.

### Selección

Finalmente, la selección se realiza simplemente comparando los vectores de prueba con los originales, de manera que el vector de la generación siguiente será aquel que tenga el mejor valor de función objetivo (Storn y Price, 1997; Chakraborty, 2008; Price *et al.*, 2005):

$$x_p^{g+1} = \begin{cases} t_p^g & \text{si } aptitud(t_p^g) > aptitud(x_p^g) \\ x_p^g & \text{en otro caso} \end{cases} \quad (11)$$

Un algoritmo DE puede ser identificado de acuerdo a tres criterios: el tipo de mutación, el número de vectores diferencia y el tipo de cruce. Hay en general tres tipos de mutación la cual depende de cómo se realiza el muestreo de los vectores a ser mutados: individuos seleccionados al azar, la mejor solución o el vector objetivo en la población. El número de vectores diferencia puede ser más de dos pero en la práctica sólo se usa uno y dos. Hay dos tipos de operadores de cruce: binomial (Ecuación 10) y exponencial (Ecuación 12). Comúnmente, la notación  $DE/x/y/z$  se usa para nombrar a un algoritmo DE. Donde  $x$ , significa la forma en que se elige el vector a ser mutado, y indica el número de vectores diferencia usados,  $z$  es el tipo de cruce diferencial implementado (Storn y Price, 1997, Price *et al.*, 2005). En el Cuadro 3 se presenta un resumen de las variantes de los algoritmos de evolución diferencial implementados en Matlab (López-Cruz *et al.*, 2008).

$$t_{p,m}^g = \begin{cases} n_{p,m}^g & \text{mientras } [randb(m) > CR] \text{ y } m \neq rnbr(p) \\ x_{p,m}^g & \text{posteriormente} \end{cases} \quad (12)$$

$m = 1, \dots, n, p = 1, \dots, NP$

Existe dentro de los algoritmos de evolución diferencial una variante importante de los mismos conocido como algoritmos adaptables, pudiéndose mencionar al JDE, JADE, FADE y SADE. Una de las principales ventajas de estos algoritmos con respecto a los de evolución diferencial clásicos es el ajuste automático de sus parámetros  $F$  (factor de escala de mutación) y  $CR$  (porcentaje o tasa de cruce). Estos parámetros son estratégicamente manipulados, por ejemplo en FADE se usa un algoritmo difuso para sintonizar  $F$  y  $CR$ . En SADE,  $F$  es generado independientemente en cada generación de acuerdo a una distribución normal con media 0.5 y desviación estandar de 0.3,  $CR$  es generado aleatoriamente

mediante una distribución normal con media CRm y desviación estandar 0.1. jDE codifica los parámetros F y CR dentro de cada individuo, de esta manera los parámetros se ajustan en cada iteración, siendo este algoritmo una variante del DE/rand/1/bin (Das y Suganthan, 2011).

**Cuadro 3:** Algoritmos de evolución diferencial (DE) implementados en Matlab.

Algoritmo	Mutación	Cruza
DE/rand/1/exp	$\vec{v}_i = \vec{u}_{r_1} + F(\vec{u}_{r_2} - \vec{u}_{r_3})$	Exponencial
DE/best/1/exp	$\vec{v}_i = \vec{u}_{best} + F(\vec{u}_{r_2} - \vec{u}_{r_3})$	Exponencial
DE/target-to-best/1/exp	$\vec{v}_i = \vec{u}_i + F(\vec{u}_{best} + \vec{u}_{r_1} - \vec{u}_i - \vec{u}_{r_2})$	Exponencial
DE/best/2/exp	$\vec{v}_i = \vec{u}_{best} + F(\vec{u}_{r_1} + \vec{u}_{r_2} - \vec{u}_{r_3} - \vec{u}_{r_4})$	Exponencial
DE/rand/2/exp	$\vec{v}_i = \vec{u}_5 + F(\vec{u}_{r_1} + \vec{u}_{r_2} - \vec{u}_{r_3} - \vec{u}_{r_4})$	Exponencial
DE/target-to-rand/1/exp	$\vec{v}_i = \vec{u}_i + K(\vec{u}_{r_3} + \vec{u}_i) + F(\vec{u}_{r_1} - \vec{u}_{r_2})$	Exponencial
DE/rand/1/bin	$\vec{v}_i = \vec{u}_{r_1} + F(\vec{u}_{r_2} - \vec{u}_{r_3})$	Binomial
DE/best/1/bin	$\vec{v}_i = \vec{u}_{best} + F(\vec{u}_{r_2} - \vec{u}_{r_3})$	Binomial
DE/target-to-best/1/bin	$\vec{v}_i = \vec{u}_i + F(\vec{u}_{best} + \vec{u}_{r_1} - \vec{u}_i - \vec{u}_{r_2})$	Binomial
DE/best/2/bin	$\vec{v}_i = \vec{u}_5 + F(\vec{u}_{r_1} + \vec{u}_{r_2} - \vec{u}_{r_3} - \vec{u}_{r_4})$	Binomial
DE/rand/2/bin	$\vec{v}_i = \vec{u}_i + K(\vec{u}_{r_3} + \vec{u}_i) + F(\vec{u}_{r_1} - \vec{u}_{r_2})$	Binomial
DE/target-to-rand/1/bin	$\vec{v}_i = \vec{u}_i + K(\vec{u}_{r_3} + \vec{u}_i) + F(\vec{u}_{r_1} - \vec{u}_{r_2})$	Binomial
DE/best/1/bin (d)	$F_j = F + 0,001 + [rand_j(0, 1) - 0,5]$ $\vec{v}_i = \vec{u}_{best} + F(\vec{u}_{r_2} - \vec{u}_{r_3})$ d=0.001	Binomial
DE/rand/either-or	$\vec{v}_i = \vec{u}_{r_1} + F(\vec{u}_{r_2} - \vec{u}_{r_3})$ si $rand_i(0, 1) < pf$ de otra manera $\vec{v}_i = \vec{u}_{r_1} + 0,5(F + 1)(\vec{u}_{r_2} + \vec{u}_{r_3} - 2\vec{u}_{r_1})$	

### Estrategias Evolutivas

Las Estrategias Evolutivas (ES, Evolution Strategies) son una subclase de búsqueda directa inspirado en la naturaleza (y optimización), pertenecen a la clase de Algoritmos Evolutivos (EA) los cuales utilizan la mutación, recombinación y selección aplicado a una población de individuos que contienen soluciones candidatas con el fin de desarrollar iterativamente mejores soluciones (Beyer, 2001). Las ES pueden ser aplicadas en todos los campos de optimización incluyendo espacios de búsqueda combinatoria, continua y discreta con o sin restricciones tanto en espacios de búsqueda mixtos.

Las versiones canónicas de las ES se denota por:  $(\mu/p, \lambda) - ES$  y  $(\mu/p + \lambda) - ES$  respectivamente. Aquí  $\mu$  denota el número de padres,  $p = \mu$  el número de mezclas (es decir, el número de padres que participan en la procreación de un hijo o nueva solución) y,  $\lambda$  es el número de descendientes. Los padres son seleccionados determinísticamente (es decir, la selección de sobrevivientes determinista) del (multi) conjunto de cualquiera de los descendientes, conocidos como *comma-selection* (se debe mantener  $\mu < \lambda$ ), o de ambos padres e hijos, conocida como *plus-selection* (Rechenberg, 1994; Beyer, 2001).

La selección se basa en el rango de aptitud de los individuos  $F(y)$  de tomar los mejores individuos (también conocida como la selección de truncamiento). En general, un *ES individual*  $a := (y, s, F(y))$  comprende el vector de parámetros objeto ( $y \in \gamma$ ) a ser optimizado, un conjunto de parámetros de estrategia ( $s$ ), especialmente necesario en los ES auto-adaptables, y la aptitud individual observada  $F(y)$  siendo equivalente a la función objetivo  $f(y)$ , es decir,  $F(y) \equiv f(y)$  en el caso más simple. La distinción entre  $F(y)$  y  $f(y)$  es necesaria, ya que  $F(y)$  puede ser el resultado de un operador de búsqueda local que se aplica a la función a optimizar, o incluso puede ser el resultado de otro ES. Por otra parte, el  $F(y)$  observado puede ser el resultado de un proceso ruidoso de evaluación de  $f(y)$  (Rechenberg, 1994; Beyer, 2001).

El algoritmo conceptual  $(\mu/p + \lambda) - ES$  es el siguiente (Rechenberg, 1994; Hansen y Ostermeier, 2001):

1. Inicializar población original  $P_\mu = \{a_1, \dots, a_\mu\}$ .
2. Generar  $\lambda$  descendencia  $\bar{a}$  que forman la población de descendientes  $\tilde{P}_\lambda = \{\tilde{a}_1, \dots, \tilde{a}_\lambda\}$ , donde cada hijo  $\bar{a}$  es generado de la siguiente manera:
  - a** Seleccionar (aleatoriamente)  $p$  padres de  $P_\mu$  (si  $p = \mu$  toma todos los padres individualmente en su lugar).
  - b** Recombinación de los  $p$  padres seleccionados para formar un individuo  $a$  recombinante  $r$ .
  - c** Mutar el conjunto  $s$  de parámetros de estrategia de los recombinantes  $r$ .
  - d** Mutar el conjunto de parámetros objetivo y de los recombinantes  $r$  utilizando el conjunto de parámetros de estrategia mutado para controlar las propiedades estadísticas de la mutación del parámetro objeto.

3. Seleccionar de la población los nuevos padres (mediante la selección de truncamiento determinista) a partir de la población de descendencia  $\tilde{P}_\lambda$  (esto se conoce como *comma-selection*, por lo general denotado como “ $(\mu, \lambda)$ -selection”), o la población de descendencia  $\tilde{P}_\lambda$  padres  $P_\mu$  (esto se conoce como *plus-selection*, por lo general denotado como “ $(\mu + \lambda)$ -selection”)
4. Regresar al paso dos, hasta que el criterio de término sea cumplido.

Dependiendo del espacio de búsqueda y la función objetivo  $f(y)$ , la recombinación y/o la mutación de los parámetros de estrategia puede o no puede ocurrir en instancias específicas del algoritmo. Por ejemplo,  $(\mu/p + \lambda)$ -ES o su equivalente  $(\mu + \lambda)$ -ES, no utiliza la recombinación. Se basa en sus  $\mu$  nuevos padres para la próxima generación, generados por los  $\mu$  padres anteriores y los  $\lambda$  hijos (generados a partir de estos padres) mediante la adopción de los mejores individuos (con respecto al  $F(y)$  observado) (Rechenberg, 1994; Hansen y Ostermeier, 2001).

Las Estrategias Evolutivas del tipo  $(\mu/p + 1)$  también se conocen como ESs en estado estacionario, es decir, estrategias sin una brecha generacional (producen un sólo descendiente por cada generación). Después de evaluar su adaptabilidad  $F(y)$ , el peor individuo se quita de la población. Estrategias de este tipo son especialmente útiles en cómputo paralelo cuando los tiempos de cálculo de aptabilidad de los individuos no son constantes, lo que permite el procesamiento paralelo asíncrono (Beyer, 2001).

Una variante de las ES es la llamada CMA-ES (Covariance Matrix Adaptation-ES, estrategias evolutivas con matriz de covarianza adaptable) el cual es un algoritmo evolutivo usado para problemas de optimización no-lineales y no convexos en dominio continuo. En este algoritmo, la forma de distribución de la mutación es generada de acuerdo a la matriz de covarianza  $C$  la cual se adapta durante el proceso de evolución. De esta manera, las mutaciones se pueden adaptar a la forma local del marco de aptitud y la convergencia al óptimo se puede incrementar considerablemente (Beyer, 2001; Hansen, 2006). Este algoritmo usa estadísticas especiales acumuladas sobre las generaciones para controlar parámetros específicos del mismo (la matriz de covarianza  $C$  y el tamaño de paso global  $\sigma$ ) (Hansen y Ostermeier, 2001).

## Programación Genética

La Programación Genética (GP, Genetic Programming) es una metodología basada en los algoritmos evolutivos e inspirada en la evolución biológica para construir programas de computación que realicen una tarea definida por el usuario. Es una especialización de los algoritmos genéticos en la que cada individuo de la población es un programa computacional (Koza, 2010). Es por tanto una técnica de aprendizaje automático utilizada para optimizar una población de programas de computadoras de acuerdo a una función de ajuste que evalúa la capacidad de cada programa para llevar a cabo la tarea en cuestión (Willis *et al.*, 1997; Koza, 2003). Un algoritmo GP manipula una población de programas de computadora cada uno de los cuales representa una posible solución a un problema. Con el fin de dar solución a un problema dado utilizando GP, Koza (1992) señala que es necesario plantear lo siguiente:

- El conjunto de terminales (*Terminal Set*): Un conjunto de variables de entrada o constantes.
- El conjunto función (*Function Set*): Un conjunto de funciones de dominio específicas se utiliza junto con el conjunto de terminales para construir posibles soluciones potenciales a un problema dado. Para regresión simbólica, esto podría consistir en un conjunto básico de funciones matemáticas, mientras que operadores booleanos y condicionales pueden ser incluidos para problemas de clasificación.
- La función de adaptabilidad (Fitness function): La adaptabilidad es un valor numérico asignado a cada miembro de una población para proporcionar una medida de la conveniencia de una solución al problema en cuestión.
- Los parámetros del algoritmo de control: Esto incluye el tamaño de la población y las probabilidades de cruce y mutación.
- El criterio de terminación: Este es generalmente un número predefinido de las generaciones o una tolerancia de error de adaptabilidad.

Cabe señalar que los primeros tres componentes determinan el espacio de búsqueda del algoritmo, mientras que los dos últimos componentes afectan la calidad y velocidad de búsqueda (Willis *et al.*, 1997). Estos programas se expresan en la programación genética

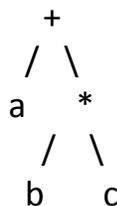


Figura 9: Ejemplo de programación genética básica

como árboles de análisis, en lugar de líneas de código. Así, por ejemplo, el programa simple  $a + (b * c)$  se representaría como se muestra en la Figura 9 (Willis *et al.*, 1997; Koza, 2003).

Los programas se componen de elementos de la *Function Set* y la *Terminal Set*, los cuales son generalmente conjuntos fijos de símbolos seleccionados a ser candidatos para la solución del problema en el dominio de interés. En la GP, la operación de cruce se implementa mediante la adopción de sub-árboles seleccionados al azar en los individuos (seleccionados de acuerdo a la adaptabilidad) y el intercambio entre ellos (Koza, 1992, 2003).

### Programación Evolutiva

La programación Evolutiva (EP, Evolutionary Programming) fue inventada por el Dr. L. J. Fogel (Fogel, 1962, 1963; Fogel, 2006). Él consideró a la inteligencia como una habilidad compuesta para realizar predicciones en un ambiente combinado con la interpretación de cada predicción de acuerdo con un objetivo dado (por ejemplo, maximizar una función de coste). Por lo tanto, consideró a la predicción como una condición previa de comportamiento inteligente. El procedimiento es el siguiente: Una población de FSM (Finite State Machine), que representa organismos individuales de una población de soluciones del problema, se expone al medio ambiente, es decir, la secuencia de imágenes que se ha observado hasta el momento. Como cada imagen de entrada es presentado a la máquina, la imagen de salida correspondiente se compara con la siguiente imagen de entrada. El valor de esta predicción se mide con respecto a la función de coste (por ejemplo, todo-nada, error cuadrático). Después de que la última predicción se realiza, una función de coste para la secuencia de imágenes (por ejemplo, la ganancia media por imagen) indica la adaptabilidad de la máquina o programa.

Las máquinas de descendencia son creadas al azar por mutación de los padres y se

califican de manera similar. Esas máquinas que proporcionan una mayor adaptabilidad se mantienen para convertirse en padres de la generación siguiente, y el proceso se repite. Cuando las nuevas imágenes son predichas, la mejor máquina disponible sirve como base para hacer una predicción y la observación nueva se agrega a la base de datos disponible. Fogel describe este proceso como “programación evolutiva”. Cabe señalar que el EP por lo general no utiliza ningún tipo de cruza como operador genético (Fogel y Fogel, 1986). El método básico de la EP consiste de tres pasos (Repetir hasta que el umbral para las iteraciones se supere, o se obtenga una solución adecuada) (Fogel y Fogel, 1986):

- Elegir una población inicial de soluciones de prueba al azar. El número de soluciones en una población es de gran relevancia para la velocidad de optimización, pero ningún resultado definitivo está disponible ya que muchas soluciones pueden ser apropiadas y otras no lo son.
- Cada solución se reproduce dentro de una nueva población. Cada una de estas soluciones son descendientes mutados de acuerdo a una distribución de tipos de mutación, que van de menor a mayor en un continuo de tipos de mutación de por medio. La severidad de la mutación se juzga en base al cambio funcional impuesta a los padres.
- Cada solución generada (procreada) se evalúa mediante el cálculo de su aptitud. Típicamente, se lleva a cabo un concurso estocástico para determinar las  $N$  soluciones a ser mantenidas en la población de soluciones, aunque en ocasiones se realiza de manera determinística. No hay requisito de que el tamaño de la población se mantenga constante.

### **Aplicaciones al modelado y control del clima de invernaderos**

En el Cuadro 4 se puede apreciar un resumen de algunas aplicaciones de AE en el modelado y control del clima de invernaderos.

#### **2.2.4. Algoritmos Bio-inspirados**

Los algoritmos Bio-Inspirados (Bio-inspired Algorithms, Natural Computing, Biologically Inspired Computing) se basan en emplear analogías con sistemas naturales o sociales

**Cuadro 4:** Aplicaciones de los Algoritmos Evolutivos al modelado y control del clima de invernaderos.

Algoritmo	Aplicación	Referencia
GA's, EP y ES comparados con LSQ (Mínimos Cuadrados) y SQP (Programación Cuadrática Secuencial).	Calibración de parámetros de un modelo climático que describe el comportamiento de la Temperatura y Humedad Relativa.	Guzmán-Cruz <i>et al.</i> (2009).
DE	Modelación del clima de un invernadero usando AE	López-Cruz <i>et al.</i> , (2008)
Algoritmo evolutivo multi-objetivo (x-MOGA) basado en un GA.	Calibración de parámetros desconocidos del modelo a partir de datos experimentales (Temperatura y Humedad Relativa) en cultivo de rosas hidropónico.	Herrero <i>et al.</i> (2007).
GA	Reducción en el consumo de energía y agua para un sistema de refrigeración.	Blasco <i>et al.</i> (2007).
GA Multi-objetivo.	Identificación de RBFNN en modelos acoplados para la humedad y temperatura.	Ferreira <i>et al.</i> (2005).
DE, ACW (Control Ajustable de variación de Peso).	Control óptimo de la concentración de nitratos en lechuga.	López-Cruz <i>et al.</i> (2003b).
DE	Control óptimo del cultivo de lechugas en invernadero	López-Cruz <i>et al.</i> , (2003c)
AE	Optimizar el tamaño de las poblaciones y generaciones a emplear con diferentes horizontes de control para observar el efecto de variación de las variables involucradas.	Ursem <i>et al.</i> (2002).
GA clásico.	Estimar costos de insumos.	Krink <i>et al.</i> (2001).
GEATbx, herramienta de trabajo de Matlab para AE y GA	Predicción de la temperatura, humedad del aire y concentración de $CO_2$ .	Pohlheim y Heißner (1996, 1999).

†GA: Algoritmo genético, EP: Programación evolutiva, ES: Estrategias evolutivas, AE: Algoritmos evolutivos.

para la resolución de problemas. Los Algoritmos Bio-Inspirados simulan el comportamiento de sistemas naturales para el diseño de métodos heurísticos y estocásticos de búsqueda, aprendizaje y comportamiento. Esta técnica, conocida en la literatura como Inteligencia de Enjambre ( SI, Swarm Intelligence) es la disciplina que se ocupa de los sistemas naturales y artificiales compuestos de muchos individuos que se coordinan usando control descentralizado y auto-organización. En particular, la disciplina se centra en los comportamientos colectivos que resultan de las interacciones locales de los individuos entre sí y con su en-

torno (Chen *et al.*, 2008). El sistema típico de inteligencia de enjambre tiene las siguientes propiedades (Kennedy y Eberhart, 1995; Kennedy *et al.*, 2001):

- Se compone de muchos individuos.
- Las interacciones entre los individuos se basan en simples reglas de comportamiento que explotan sólo información local que intercambian los individuos directamente o a través del medio (stigmergia).
- Los individuos son relativamente homogéneos (todos son idénticos o pertenecen a una tipología menor)
- El comportamiento global del sistema resulta de las interacciones de los individuos entre sí y con su entorno, es decir, el grupo se auto-organiza.

Ejemplos de sistemas estudiados por SI son Optimización por Colonia de Hormigas (ACO, Ant Colony Optimization), Optimización por Enjambre de Abejas (BCO, Bee Colony Optimization), Optimización por Enjambre de Partículas (PSO, Particle Swarm Optimization).

### **Optimización por Colonia de Hormigas**

En los años 90 la optimización por colonia de hormigas (ACO, Ant Colony Optimization) se introdujo como un novedoso método bio-inspirado para la solución de problemas de optimización combinatoria (Dorigo, 1992; Dorigo *et al.*, 1996; Dorigo y Stützle, 2004). La fuente inspiradora de la ACO es el comportamiento real de las hormigas. Cuando están buscando comida, las hormigas inicialmente exploran el área que rodea su nido en una forma aleatoria. Tan pronto como una hormiga encuentra una fuente de alimento, esta lo evalúa y lleva una muestra de regreso a su nido. Durante el viaje de retorno, la hormiga deposita un rastro de feromona sobre el suelo. La feromona depositada, que depende de la cantidad y calidad de la comida, guía a otras hormigas a la fuente de alimento. La comunicación indirecta entre las hormigas vía rastros de feromona capacita a estas a encontrar la ruta más óptima entre su nido y la fuente de comida (Jackson y Ratnieks, 2006). La feromona es un material químico depositado por las hormigas, que sirve como un medio de comunicación fundamental entre ellos y los guía para determinar el siguiente movimiento (Niknam, 2008). En general, para seleccionar la ruta de acceso siguiente, la probabilidad

de transición de estado se define de la siguiente manera:

$$P_{ij} = \frac{(\tau_{ij})^{\gamma_2} (1/L_{ij})^{\gamma_1}}{\sum_{j=1, j \neq i}^{NA} (\tau_{ij})^{\gamma_2} (1/L_{ij})^{\gamma_1}} \quad (13)$$

donde  $\tau_{ij}$  y  $L_{ij}$  son la intensidad de feromona y la longitud de la trayectoria entre los nodos  $j, i$ .  $\gamma_1$  y  $\gamma_2$  son los parámetros de control para determinar los pesos de intensidad del rastro de feromona y la longitud de la trayectoria, respectivamente.  $NA$  es el número de hormigas. Después de seleccionar la ruta de acceso siguiente, la intensidad del rastro de feromona se actualiza a:

$$\tau_{ij}(k+1) = \rho \tau_{ij}(k) + \Delta \tau_{ij} \quad (14)$$

En la ecuación (14),  $\rho$  es un coeficiente que representa la evaporación del rastro de feromona entre el tiempo  $k$  y  $k+1$ . Finalmente,  $\Delta \tau_{ij}$  es la cantidad de rastro de feromona añadido a  $\tau_{ij}$  por las hormigas (Dorigo y Stützle, 2004; Niknam, 2008).

Una hormiga artificial es un agente computacionalmente simple que intenta construir una solución a un problema explorando para ello los rastros de feromona disponibles y presenta las siguientes características (Dorigo y Blum, 2005; Dorigo, *et al.*, 2006):

- Busca soluciones válidas de coste mínimo para el problema a resolver.
- Tiene una memoria en la que almacena el camino recorrido para poder reconstruir soluciones válidas, evaluar las soluciones generadas, o bien reconstruir el camino completo llevado por la hormiga.
- Lleva a cabo los movimientos aplicando reglas de transición, en función de los rastros de feromona que están disponibles, de los valores heurísticos, de la memoria privada de la hormiga y de las restricciones del problema.
- Durante su recorrido, modifica el entorno depositando una cantidad de feromona.
- Una vez que una hormiga ha construido una solución válida puede recorrer el camino de vuelta actualizando los espacios visitados.

### **Optimización por Colonia de Abejas**

La Optimización por Colonia de Abejas (BCO, Bee Colony Optimization, ABC, Artificial Bee Colony, BA, Bee Algorithm) es un algoritmo de búsqueda basado en una

población. En él se imita el comportamiento de alimentación de enjambres de abejas. En su versión básica, el algoritmo realiza una especie de búsqueda por vecindad combinada con la búsqueda aleatoria y puede ser utilizado para la optimización combinatoria y optimización funcional (Pham *et al.*, 2006; Sundar y Singh, 2010). En el algoritmo ABC, la colonia artificial de abejas contiene tres grupos importantes de abejas: *abejas obreras*, *observadoras* y *exploradoras*. La posición de una fuente de alimentos representa una posible solución al problema de optimización y la cantidad de néctar de una fuente de alimentación corresponde a la calidad (aptitud) de la solución asociada. El número de las abejas obreras o abejas observadoras es igual al número de soluciones en la población. En el primer paso, el ABC genera una distribución aleatoria de la población inicial  $P(C = 0)$  de soluciones  $SN$  (posiciones de las fuentes de alimentos), donde  $SN$  denota el tamaño de las abejas obreras o las abejas observadoras. Cada solución  $x_i$  ( $i = 1, 2, \dots, SN$ ) es un vector  $D$ -dimensional. Aquí,  $D$  es el número de parámetros de optimización (Karaboga y Basturk, 2007; Karaboga y Akay, 2009; Karaboga y Ozturk, 2011). Una abeja observadora selecciona una fuente de alimento en función del valor de probabilidad  $P_i$  asociado con esa fuente y se calcula mediante la siguiente expresión (Sundar y Singh, 2010):

$$P_i = \frac{aptitud_i}{\sum_{i=1}^{SN} aptitud_n} \quad (15)$$

donde  $aptitud_i$  es el valor de aptitud de la solución  $i$  la cual es proporcional a la cantidad de néctar de la fuente de alimentación en la posición  $i$  y  $SN$  es el número de fuentes de alimento que es igual al número de abejas obreras o abejas observadoras. Con el fin de producir una posición de alimentos candidato con respecto a la anterior almacenada en memoria, la ABC utiliza la siguiente expresión:

$$v_{ij} = X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) \quad (16)$$

donde  $k \in \{1, 2, \dots, SN\}$  y  $j \in \{1, 2, \dots, D\}$  son índices seleccionados aleatoriamente. Aunque  $k$  es determinada aleatoriamente, debe ser diferente de  $i$ .  $\phi_{ij}$  es un número aleatorio considerado dentro del intervalo  $[-1, 1]$ , el cual controla la producción de fuentes de alimento vecinas a  $X_{ij}$  y representa la comparación entre dos posiciones de alimento visibles por una abeja (Karaboga y Akay, 2009; Sundar y Singh, 2010; Karaboga y Ozturk, 2011). Cuando una fuente de alimento es abandonada por falta de néctar, las abejas exploradoras buscan una nueva fuente de alimento para reemplazar la anterior. En el algoritmo ABC, este fenómeno se conoce como *límite de abandono* y se realiza después de un número determinado de ciclos de ejecución. Por lo tanto, se asume que la fuente abandonada es  $X_i$ ,

entonces las exploradoras descubren una nueva fuente de alimento que reemplazará a  $X_i$ . Esto se puede definir como:

$$X_i^j = X_{min}^j + rand[0, 1](X_{max}^j - X_{min}^j) \quad (17)$$

Después de que cada posición de la fuente candidata  $V_{ij}$  es producido y evaluado por la abeja artificial, su desempeño se compara con la fuente anterior. Si la nueva fuente de alimento tiene un néctar igual o mejor que la fuente antigua, se sustituye por la antigua en la memoria. De lo contrario, la antigua fuente se conserva en la memoria. En otras palabras, un mecanismo ambicioso se emplea en la operación de selección entre la fuente anterior y la candidata (Karaboga y Akay, 2009; Stanarevic *et al.*, 2010; Sundar y Singh, 2010; Karaboga y Ozturk, 2011). Los principales pasos del algoritmo son las siguientes:

- Inicializar la Población con soluciones aleatorias.
- Evaluar aptitud de la población.
- Producir nuevas soluciones para las abejas obreras.
- Aplicar el proceso de selección ambiciosa.
- Calcular los valores de probabilidad.
- Producir las nuevas soluciones para los espectadores.
- Seleccionar las abejas más capaces de cada sitio (selección ambiciosa).
- Determinar la solución abandonada por el explorador, y sustituirla por una nueva solución producida al azar.
- Memorizar la mejor solución logrado hasta ahora.
- Repetir hasta que las exigencias hayan sido cumplidas.

### **Optimización por Enjambres de Partículas**

La Optimización por Enjambres de Partículas (PSO, Particle Swarm Optimization) es un enfoque estocástico basado en una población para resolver problemas de optimización discreta y continua. Pertenece a la clase de técnicas de inteligencia de enjambre (SI, Swarm Intelligence). En PSO, los agentes de software simples, llamadas partículas, se mueven en

el espacio de búsqueda del problema analizado (Eberhart y Shi, 2001). La posición de una partícula representa una solución potencial al problema de optimización en cuestión (Sierakowski y Coelho, 2005). Cada partícula busca una mejor posición en el espacio de búsqueda cambiando su velocidad de acuerdo a normas inspiradas originalmente en los modelos de comportamiento de las parvadas de aves (Kennedy y Eberhart, 1995; Eberhart y Shi, 2001).

Las reglas que gobiernan el movimiento de las partículas en el espacio de búsqueda del problema de optimización pueden ser vistas como el modelo de comportamiento social del ser humano en el cual los individuos ajustan sus creencias y actitudes de acuerdo con la de sus compañeros (Kennedy y Eberhart, 1995).

En PSO, el enjambre se compone de un conjunto de partículas  $P = \{p_1, p_2, \dots, p_k\}$ . La posición de una partícula corresponde a una solución candidata del problema de optimización considerado, el cual es representado por una función objetivo  $f(y)$ . En cualquier intervalo de tiempo  $t$ ,  $p_i$  tiene una posición  $\vec{x}_i^t$  y una velocidad  $\vec{v}_i^t$  asociada a ella. La mejor posición que la partícula  $p_i$  (con respecto a  $f(y)$ ) ha tenido en el intervalo de tiempo  $t$  está representada por el vector  $\vec{b}_i^t$  (también conocida como mejor partícula). Por otra parte, una partícula  $p_i$  recibe información de su vecindad  $N_i \subseteq P$ . En el algoritmo estándar PSO, las relaciones de vecindad entre partículas son comúnmente representadas como una gráfica  $G = \{V, E\}$ , donde cada vértice en  $V$  corresponde a una partícula en el enjambre y cada borde en  $E$  establece una relación de vecinos entre un par de partículas. El gráfico resultante se conoce comúnmente como la topología de la población del enjambre (Eberhart y Shi, 2001).

El algoritmo PSO inicia generando una posición aleatoria de las partículas dentro de una región de inicialización  $\Theta' \subseteq \Theta$ . Las velocidades normalmente son inicializadas dentro de  $\Theta'$ , pero pueden ser inicializadas en cero o en pequeños valores aleatorios para evitar que las partículas abandonen el espacio de búsqueda durante la primera iteración. Durante la ejecución del algoritmo, las velocidades y posiciones de las partículas son actualizadas en cada iteración hasta que se cumple un criterio de paro. Las reglas de actualización son las siguientes (Clerc y Kennedy, 2002):

$$\begin{aligned}\vec{v}_i^{t+1} &= W\vec{v}_i^t + \varphi_1 \vec{U}_1^t (\vec{b}_i^t - \vec{x}_i^t) + \varphi_2 \vec{U}_2^t (\vec{l}_i^t - \vec{x}_i^t) \\ \vec{x}_i^{t+1} &= \vec{x}_i^t + \vec{v}_i^{t+1}\end{aligned}\tag{18}$$

donde  $w$  es un parámetro llamado peso inercial,  $\varphi_1$  y  $\varphi_2$  son dos parámetros llamados coeficientes de aceleración,  $\vec{U}_1^t$  y  $\vec{U}_2^t$  son dos matrices diagonales  $n \times n$  en el cual las entradas

de la diagonal principal son números aleatorios uniformemente distribuidos en el intervalo  $[0,1]$ . En cada iteración las matrices son regeneradas. Normalmente, el vector  $\vec{l}_i^t$  se refiere como la mejor vecindad, es la mejor posición encontrada por una partícula en la vecindad de partículas  $P_i$ , es decir,  $f(\vec{l}_i^t) \leq f(\vec{b}_i^t) \quad \forall p_j \in N_i$ . Si los valores de  $w$ ,  $\varphi_1$  y  $\varphi_2$  son seleccionados apropiadamente, se garantiza que la velocidad de las partículas no tiendan a infinito (Clerc y Kennedy, 2002). Los tres términos en las reglas de actualización anteriores caracterizan el comportamiento local que siguen las partículas. El primer término es conocido como inercia o momento y sirve como memoria de la dirección de vuelo previo para evitar un cambio de dirección drástica de las partículas. El segundo término, llamado componente cognitivo, modela la tendencia de las partículas a retornar antes de encontrar las mejores posiciones. El tercer término, llamado componente social cuantifica el rendimiento de una partícula en relación a sus vecinos. Representa una norma del grupo o estándar que debe ser alcanzado. En algunos casos, las partículas pueden ser atraídas a las regiones fuera del espacio de búsqueda factible  $\Theta$ . Por esta razón, se han ideado mecanismos para preservar la viabilidad de soluciones y una operación adecuada de enjambre (Engelbrecht, 2005). Uno de los mecanismos, por lo menos perjudicial, para la preservación de la viabilidad es aquel en el que las partículas que salen de  $\Theta$  no se les permite mejorar su posición para que se sientan atraídas de nuevo al espacio factible en iteraciones posteriores.

### **Aplicación al modelado y control del clima de invernaderos**

Las aplicaciones de los ABI al modelado del ambiente de invernaderos es muy incipiente y por lo tanto esta familia de algoritmos tiene un gran potencial de aplicación futura. Coelho *et al.* (2002, 2005) aplicaron un algoritmo de Optimización por Enjambre de partículas (PSO) para el diseño de un controlador predictivo basado en modelo para el clima de un invernadero sujeto a restricciones. El algoritmo PSO fue comparado con otros algoritmos de optimización como SQP y GA. Para simular el clima interno y externo del invernadero hicieron uso de modelos ARX y series de tiempo auto-regresivos. El conjunto de datos usados fueron obtenidos experimentalmente a través de sensores ubicados estratégicamente dentro y fuera del invernadero. El mejor desempeño fue obtenido con el algoritmo PSO.

### 2.2.5. Algoritmos Híbridos

Los sistemas híbridos combinan dos o más técnicas de Inteligencia artificial para ganar fortaleza y superar vulnerabilidad. Hay tres principales tipos de sistemas híbridos de acuerdo a como son combinados: secuencial, auxiliar y empotrado (Gray y Kilgour, 1997). En los híbridos secuenciales, el primer prototipo pasa su salida al segundo para generar la salida deseada. En un híbrido auxiliar, el primer prototipo obtiene alguna información del segundo y genera la salida. En un híbrido empotrado, los dos prototipos están contenidos uno dentro del otro.

De acuerdo con Chen *et al.* (2008), los sistemas híbridos pueden ser formados por cualquier combinación de técnicas de inteligencia artificial, como ejemplos se pueden citar: GA y ANN, FL y ANN, GA e Inferencia Bayesiana, PSO y ANN, PSO y GA (Cheng-Jian y Yong-Ji, 2007).

Los algoritmos híbridos más comunes son los Neuro-Difusos (AND), constituidos principalmente de ANN y FL lo cual permite una relación simbiótica, aprovechándose el conocimiento de un experto y la capacidad de aprendizaje y eficiencia computacional de la red neuronal, logrando un sistema de decisiones más inteligente. Debido a que cada técnica tiene capacidades y limitaciones, su aplicación en el modelado y control del ambiente en invernaderos los hace aptos para esta tarea una vez que han sido fusionados (López-Cruz y Hernández-Larragoiti, 2010; Cheng-Jian y Yong-Ji, 2007; Tien y Van Straten, 1998). Un AND popular es el ANFIS (Adaptive Network-based Fuzzy Inference System) el cual es un sistema de inferencia difuso implementado en el marco de redes neuronales adaptivas (Jang, 1993; Jang y Chuen-Tsai, 1995), este algoritmo usa un procedimiento de aprendizaje híbrido combinando los algoritmos gradiente descendente y filtro de Kalman, realiza un mapeo entrada-salida basado en conocimiento humano (a base de reglas difusas sí-entonces) y pares de datos entrada-salida estipulados (Jang y Chuen-Tsai, 1995). Otro sistema Neuro-Difuso integrado propuesto por Tien (1997) es el llamado NUFZY, este es un sistema difuso simplificado representado por el modelo Takagi-Sugeno de orden cero. Es un tipo especial de ANN caracterizado por conexiones parciales en su primera y segunda capas. A través de las conexiones de la red el sistema NUFZY realiza un razonamiento difuso particular (Tien y Van Straten, 1998). Los ANDs ofrecen la ventaja de modelar sistema no-lineales y adicionar robustez en situaciones de datos incompletos e imprecisos, donde existe incertidumbre.

## Aplicación al modelado y control del clima de invernaderos

En el Cuadro 5 se puede apreciar un resumen de algunas aplicaciones de los algoritmos híbridos en el modelado y control del clima de invernaderos.

**Cuadro 5:** Aplicaciones de los algoritmos híbridos al modelado y control del clima de invernaderos.

Algoritmo	Aplicación	Referencia
NUFZY.	Identificación del clima, el crecimiento de lechuga y crecimiento del jitomate.	Tien y Van Straten (1995, 1996, 1998).
ANFIS.	Predicción del rendimiento en la producción de jitomate.	Qaddoum e Illiescu (2011).
ANFIS, Adaptive Neuro-Fuzzy training of Sugeno-type Inference System.	Predecir la temperatura y humedad relativa del aire con ventilación natural.	López-Cruz y Hernández-Larragoiti (2010).
Modelo Neuro-Difuso.	Modelar la evapotranspiración instantánea de una planta de jitomate.	Ramos-Fernández <i>et al.</i> (2010).
ANN y GA.	Predecir temperatura y humedad.	He <i>et al.</i> (2007).
ANN y FL.	Modelado de producción de materia seca en el cultivo del jitomate.	Kazheunikau <i>et al.</i> (2004).
FL y GA.	Control de temperatura.	Goggos y King (2000).

†FL: Lógica difusa, GA: Algoritmo genético, ANN: Red neuronal artificial, ANFIS: Red adaptiva basada en un sistema de inferencia difuso, NUFZY: Algoritmo neuro-difuso.

### 2.2.6. Conclusiones

Tanto los sistemas difusos como las redes neuronales se han usado desde hace más de una década para modelar y controlar principalmente el clima de los invernaderos. Se han obtenido mejores resultados con los métodos inteligentes que con los enfoques tradicionales. También se han logrado más avances en la modelación del ambiente invernadero y menos en su control. La mayoría de las investigaciones de modelado se han centrado en el clima y poco en procesos relacionados con el cultivo tales como el crecimiento, desarrollo, fotosíntesis, respiración o transpiración, lo cual es fundamental si se desea optimizar el biosistema invernadero. En el caso de ANN se han usado tanto para modelar como para controlar las redes estáticas sobre las dinámicas, sin embargo para controlar en tiempo real, se requiere de redes dinámicas más complejas.

Los Algoritmos Bio-inspirados solo han empezado a ser estudiados y aplicados en problemas de Ingeniería Agrícola y en el sistema invernadero. En el caso del Cómputo Evolutivo, algunos algoritmos como los GA's y los DE son muy comunes en aplicaciones de modelado pero su aplicación al control es muy incipiente y escasa. Los Algoritmos Híbridos han sido poco usados para el control del clima. Sin embargo, la aplicación al modelado está incrementando considerablemente. Por lo tanto, se considera a los algoritmos Bio-inspirados, evolutivos e híbridos un área bastante prometedora para optimizar y controlar el ambiente de los invernaderos.

---

# CAPÍTULO 3

---

## ESTIMACIÓN DE PARÁMETROS PARA UN MODELO DE CRECIMIENTO DE CULTIVOS USANDO ALGORITMOS EVOLUTIVOS Y BIO-INSPIRADOS. PARTE I

Elmer César Trejo Zúñiga<sup>1</sup>, Irineo L. López Cruz<sup>1</sup>, Agustín Ruíz García<sup>1</sup>

### Resumen

La calibración de modelos dinámicos para crecimiento y desarrollo de cultivos con muchos parámetros, genera estimaciones imprecisas de estos y predicciones erróneas del modelo cuando se usa estimación por Mínimos Cuadrados o Máxima Verosimilitud. El presente estudio muestra el uso de métodos globales de optimización para resolver este problema. Se presenta una comparación del desempeño de un algoritmo evolutivo (Evolución Diferencial, DE) y dos bio-inspirados: Búsqueda Cuco (CS) y Búsqueda Cuco Modificado (MCS). El problema prueba consiste en estimar los 25 parámetros del modelo para crecimiento potencial de cultivos SUCROS (a Simple and Universal CROp growth Simulator). Se usaron datos obtenidos de un experimento de crecimiento de un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. ex Horm.) llevado a cabo en Chapingo, México. El objetivo fue determinar qué algoritmo genera valores para los parámetros del modelo que permitan lograr las predicciones más precisas. Se realizó un análisis de varianza (ANDEVA) para evaluar estadísticamente la eficiencia y efectividad de los algoritmos propuestos. Los resultados mostraron un mejor desempeño del algoritmo DE estándar (DE/rand/1/bin) en términos de eficiencia y eficacia para converger a una solución óptima. Los algoritmos bio-inspirados mostraron un buen desempeño, debido a esto, son confiables y pueden ser aplicados en el proceso de estimación de parámetros de modelos de cultivos.

---

<sup>1</sup>Postgrado en Ingeniería Agrícola y Uso Integral del Agua, Universidad Autónoma Chapingo, Km. 38.5 Carr. México-Texcoco. C.P. 56230. Chapingo, Estado de México.

\* Capítulo ampliado basado en Trejo-Zúñiga *et al.* (2013).

**Palabras clave:** Estimación de parámetros, Evolución Diferencial, Búsqueda Cuco, SU-CROS.

### 3.1. Introducción

La estructura de los modelos dinámicos para crecimiento de cultivos consiste en un conjunto de ecuaciones diferenciales ordinarias de primer orden, en general no-lineales. Estas ecuaciones poseen un conjunto de coeficientes que, generalmente, representan parámetros fisiológicos, los cuales deben ser determinados con precisión para lograr un mejor ajuste entre las predicciones del modelo y los datos observados. Es necesario estimar algunos o todos los parámetros del modelo con datos de campo para mejorar el ajuste de las predicciones a las observaciones (Wallach *et al.*, 2001). La determinación de los valores de los parámetros puede plantearse como un problema de optimización, lo que permite el uso de diversos algoritmos. Generalmente, se utilizan métodos de búsqueda local que usan el gradiente de la función objetivo. Sin embargo, debido a la alta no-linealidad de los modelos de crecimiento y desarrollo de cultivos y la existencia de una posible dependencia entre los parámetros (epistasia), el problema de optimización puede resultar no convexo, multimodal o sobre-parametrizado. Además, cuando se estiman muchos parámetros (más de 10) mediante algoritmos de mínimos cuadrados o de máxima verosimilitud, se pueden generar varianzas muy altas en los parámetros estimados e imprecisiones en las predicciones del modelo (Makowski *et al.*, 2006). Para resolver problemas de optimización multimodal, los métodos globales de optimización como los algoritmos evolutivos (Evolución Diferencial) y bio-inspirados (Búsqueda Cuco) pueden ofrecer buenas aproximaciones al óptimo global (Price *et al.*, 2005; Yang y Deb, 2009, 2010; Walton *et al.*, 2011; Civicioglu y Besdok, 2013). Wallach *et al.* (2001) propusieron un procedimiento automático para estimar los parámetros de modelos de cultivo calibrados con datos de campo, pero no usaron métodos globales de optimización. Pabico *et al.* (1999) estimaron coeficientes de cultivares en modelos de cultivo mediante algoritmos genéticos, los cuales son un método evolutivo global pero ineficiente. Dai *et al.* (2009) realizaron un proceso de estimación de parámetros para modelos de crecimiento de cultivos en invernadero usando también algoritmos genéticos. Recientemente, Guzmán-Cruz *et al.* (2009) realizaron un proceso de calibración de un modelo del clima de un invernadero usando algoritmos evolutivos, pero no usaron Evolución diferencial. De acuerdo a la literatura revisada, no se han aplicado algoritmos de

optimización bio-inspirados para la estimación de parámetros de modelos para crecimiento de cultivos en campo abierto. Por lo tanto, el objetivo de la presente investigación fue evaluar el desempeño, a través de la eficiencia y efectividad, de los algoritmos de Evolución Diferencial (DE) reconocido como uno de los algoritmos evolutivos más eficientes, Búsqueda Cuco (CS) y Búsqueda Cuco Modificado (MCS) para estimar 25 parámetros del modelo de crecimiento SUCROS (Goudriaan y Van Laar, 1994) aplicado a un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. ex Horm.) usando datos obtenidos de un experimento realizado en Chapingo, México, en el verano de 2007 (López-López *et al.*, 2009). La eficiencia o costo computacional de un algoritmo es el número de veces que la función objetivo es evaluada para encontrar un óptimo. La efectividad o capacidad de un algoritmo para alcanzar un óptimo en un problema de optimización multimodal, se calcula contando el número de veces que el algoritmo converge a un óptimo con diferentes valores iniciales para el proceso iterativo (Khompatraporn *et al.*, 2005; Baritomba y Hendrix, 2005).

## 3.2. Materiales y métodos

### 3.2.1. Algoritmo de evolución diferencial

Los algoritmos de evolución diferencial son una clase de algoritmos evolutivos planteados como métodos de optimización global (Storn y Price, 1997; López-Cruz *et al.*, 2003; Price *et al.*, 2005). El método consiste en generar en forma aleatoria una población inicial de vectores de números reales  $P(0) = (u_{i,j}); i = 1, \dots, NP, j = 1, \dots, D$  dentro del dominio de cada una de las variables a ser optimizadas.  $D$  es la dimensión del vector y  $NP$  es el tamaño de la población. Normalmente el dominio de las variables del problema está restringido entre valores mínimos  $b_{j,L}$  y máximos  $b_{j,U}$ . Por lo tanto, cada individuo de la población inicial se genera mediante la ecuación (1):

$$u_{i,j} = rand_j(0,1)(b_{j,U} - b_{j,L}) + b_{j,L} \quad (1)$$

donde  $rand_j(0,1)$  genera un número aleatorio uniformemente distribuido dentro del rango  $[0,1]$ . Después de que cada vector es evaluado, calculando la función objetivo  $f(u_i)$ , se inicia un ciclo interno y otro externo en los cuales se aplican los operadores de evolución diferencial. El ciclo externo se detiene cuando se satisface un criterio de convergencia como el número especificado de iteraciones. El ciclo interno permite que cada individuo de la población sea considerado una vez como padre de las nuevas soluciones y sea usado en

los operadores evolutivos de mutación, cruzamiento y selección diferencial (Figura 5). En la presente investigación se usó el algoritmo de evolución diferencial llamado estándar DE/rand/1/bin, lo cual significa que el vector mutado fue elegido en forma aleatoria, y solamente dos vectores tomados también en forma aleatoria de la población se combinaron para calcular un vector diferencia. El cruzamiento fue implementado mediante el esquema binomial (Price, 1999; Price *et al.*, 2005).

**Paso 1:** Colocar los valores de los parámetros de control del algoritmo DE: factor de escala  $F$ , tasa de cruce  $CR$  y tamaño de la población  $NP$ .

**Paso 2:** Ajustar el número de generación  $G = 0$  e iniciar aleatoriamente una población de  $NP$  individuos  $P_G = \{\vec{x}_{1,G}, \dots, \vec{x}_{NP,G}\}$  con  $\vec{x}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \dots, x_{D,i,G}]$  y cada individuo uniformemente distribuido en el rango  $[\vec{x}_{min}, \vec{x}_{max}]$ , donde  $\vec{x}_{min} = \{x_{1,min}, x_{2,min}, \dots, x_{D,min}\}$  y  $\vec{x}_{max} = \{x_{1,max}, x_{2,max}, \dots, x_{D,max}\}$  con  $i = [1, 2, \dots, NP]$

**Paso 3:** While criterio de paro no sea satisfecho do

**for**  $i = 1$  hasta  $NP$  (hacer para cada individuo secuencialmente)

**Paso 2.1** Mutación

Generar un vector de donantes  $\vec{v}_{i,G} = \{v_{1,i,G}, \dots, v_{D,i,G}\}$  correspondiente al  $i$ -ésimo vector objetivo  $\vec{x}_{i,G}$  a través del esquema de mutación diferencial como:

$$\vec{v}_{i,G} = \vec{x}_{r_1^i,G} + F(\vec{x}_{r_2^i,G} - \vec{x}_{r_3^i,G})$$

**Paso 2.2** Cruza

Generar un vector de ensayo  $\vec{u}_{i,G} = \{u_{1,i,G}, \dots, u_{D,i,G}\}$  correspondiente al  $i$ -ésimo vector objetivo  $\vec{x}_{i,G}$  a través de la cruce binomial de la siguiente manera:

$u_{j,i,G} = v_{j,i,G}$  si  $(rand_{i,j}[0,1] \leq CR \text{ o } j = j_{rand})$

$x_{j,i,G}$ , de otra forma

**Paso 2.3** Selección

Evaluar el vector de ensayo  $\vec{u}_{i,G}$

**if**  $f(\vec{u}_{i,G}) \leq f(\vec{x}_{i,G})$ , **then**  $\vec{x}_{i,G+1} = \vec{u}_{i,G}$

**else**  $\vec{x}_{i,G+1} = \vec{x}_{i,G}$

**end if**

**end for**

**Paso 2.4** Incrementar el contador de generaciones

$G = G + 1$

**end while**

Figura 1: Pseudocódigo para el algoritmo DE con cruce binomial

## Operador de mutación

La mutación consiste en la construcción de  $NP$  vectores aleatorios ( $\vec{v}$ ), los cuales son creados de acuerdo a la ecuación (2):

$$\vec{v}_i = \vec{u}_{r_1} + F(\vec{u}_{r_2} - \vec{u}_{r_3}) \quad (2)$$

donde  $r_1, r_2, r_3 \in [1, 2, \dots, NP]$  son distintos entre sí verificando que se satisfaga las condiciones  $r_1, r_2, r_3 \neq i, r_1 \neq r_2, r_1 \neq r_3, r_2 \neq r_3$ .  $F$  es un parámetro que controla la tasa de mutación, y se encuentra en el rango  $[0, 2]$ .

### Operador de cruzamiento

El cruzamiento consiste en combinar el vector  $\vec{v}_i = [v_{i1}, v_{i2}, \dots, v_{id}]$ , mutado previamente con otro vector llamado el vector blanco  $\vec{u} = [u_{i1}, u_{i2}, \dots, u_{id}]$ , el cual puede ser considerado como un individuo padre de la generación anterior. Se genera así un nuevo vector descendiente  $\vec{u}'_i = [u'_{i1}, u'_{i2}, \dots, u'_{id}]$  llamado vector ensayo. El llamado cruzamiento binomial se implementa mediante la ecuación (3):

$$u'_{ij} = \begin{cases} v_{ij} & \text{si } (randb(j) \leq CR) \quad \text{o} \quad j = rnbr(i) \\ u_{ij} & \text{si } (randb(j) > CR) \quad \text{y} \quad j \neq rnbr(i) \end{cases} \quad j = 1, 2, \dots, d; i = 1, 2, \dots, NP \quad (3)$$

donde  $randb(j) \in [0, 1]$  es la  $j$ -ésima evaluación de un generador de números aleatorios uniforme;  $rnbr(i) \in [1, 2, \dots, d]$  es un índice generado en forma aleatoria.  $CR \in [0, 1]$ , es un parámetro de probabilidad de cruzamiento que controla la diversidad de la población.

### Operador de selección

Esta se realiza simplemente comparando el valor de la función de costo  $f(\vec{u}_i)$  del vector blanco  $u_i$  contra el correspondiente valor de la función objetivo  $f(\vec{u}'_i)$  del vector de ensayo  $\vec{u}'_i$ . El vector que tenga el menor valor de la función de costo pasa a ser miembro de la población en la siguiente generación (Price, 1999; Price *et al.*, 2005; Chakraborty, 2008).

### 3.2.2. Algoritmo Cuco clásico (CS) y Modificado (MCS)

Este algoritmo está inspirado en el comportamiento parasitario de las crías de algunas especies de aves Cuco (*Cuculus canorus*) en combinación con el comportamiento de vuelo Lévy de algunos pájaros y moscas de la fruta (Yang y Deb, 2010). Esto puede ser resumido en tres reglas idealizadas, a saber: 1) Cada Cuco pone un huevo a la vez y lo deposita en un nido elegido al azar; 2) Los mejores nidos con huevos de alta calidad serán transferidos a las próximas generaciones; 3) El número de nidos hospederos es fijo y el huevo puesto por un cuco es descubierto por el ave hospedera con una probabilidad  $pa \in [0, 1]$ . En este caso el hospedero puede desechar el huevo o abandonar el nido y construir uno nuevo. Para simplificar, esta última suposición puede ser aproximada por la fracción  $pa$  de los  $n$  nidos que son reemplazados (nuevas soluciones aleatorias). En su forma más simple cada huevo

en el nido representa una solución y cada huevo puesto por un Cuco representa una nueva solución al problema en cuestión (Gandomi *et al.*, 2013; Civicioglu y Besdok, 2013).

### Vuelos Lévy

Al generar nuevas soluciones  $x^{t+1}$  para un Cuco  $i$ -ésimo se genera un vuelo Lévy dado por la ecuación (4):

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\lambda), \quad (4)$$

donde  $\alpha > 0$  es el tamaño de paso y debe estar relacionado con las escalas del problema en cuestión. En la mayoría de los casos se puede utilizar  $\alpha = 1$ . La ecuación (4) es esencialmente la ecuación estocástica de recorrido aleatorio. En general, un recorrido aleatorio es una cadena de Markov cuya próxima posición depende de la ubicación actual (primer término en la ecuación) y la probabilidad de transición (segundo término). El símbolo  $\oplus$  significa producto con respecto a las entradas. Este recorrido aleatorio a través del vuelo Lévy es más eficiente en la exploración del espacio de búsqueda. El vuelo Lévy representa un recorrido aleatorio, siendo la longitud de paso obtenida mediante una distribución de Lévy (ecuación 5):

$$\text{Lévy} \sim u = t^{-\lambda}, \quad (1 < \lambda \leq 3), \quad (5)$$

la cual tiene una media y varianza infinitas. En este caso, los pasos consecutivos de un Cuco forman un proceso de caminata aleatoria que obedece a una distribución de ley de potencias con longitud de paso de doble cola (Yang y Deb, 2009; Yildiz, 2013). En el caso del algoritmo de Búsqueda Cuco Modificado (MCS) (Walton *et al.*, 2011) el tamaño de paso del vuelo Lévy ( $\alpha$ ) disminuye cuando se incrementa el número de generaciones, esto con la finalidad de estimular más la búsqueda localizada de los individuos o huevos, cuando estén cercanos a la solución. Se selecciona un valor inicial del tamaño de paso del vuelo de Lévy  $A=1$  y en cada generación se calcula un nuevo tamaño de paso usando  $\alpha = A/\sqrt{G}$ , donde  $G$  es el número de generación. Esta búsqueda exploratoria se realiza solamente en la fracción de nidos a ser abandonados. Otra modificación es la adición de intercambio de información entre los huevos con la finalidad de acelerar la convergencia hacia el mínimo, en contraste con el algoritmo Búsqueda Cuco clásico (CS) donde no hay intercambio de información entre individuos y la búsqueda se realiza de forma independiente. En el MCS, una fracción de huevos con mejor aptitud es ubicada en un grupo superior. Para cada uno de estos huevos, se crea un segundo grupo y se elige un nuevo huevo al azar generándose una

línea que une a estos huevos seleccionados. La distancia a lo largo de esta línea se calcula utilizando la inversa de la proporción aurea  $\varphi = (1 + \sqrt{5})/2$  de tal manera que el nuevo huevo está más cerca del huevo de mayor aptitud. En el caso de que ambos huevos tengan la misma aptitud, el nuevo huevo se genera en el punto medio. Existe una posibilidad de que, en este paso, sea generado dos veces el mismo huevo. En este caso, un vuelo de Lévy de búsqueda local es realizado entre los nidos elegidos al azar con tamaño de paso  $\alpha = A/G^2$  (Figuras 3 y 4).

```

Inicializar una población de  $n$  nidos hospederos  $x_i, i = 1, 2, \dots, n$ 
for  $x_i$  do
    Calcular la aptitud  $F_i = f(x_i)$ 
end for
while número de evaluaciones de la función objetivo
    < máximo número de evaluaciones do
    Generar un huevo de Cuco ( $x_j$ ) realizando un vuelo de Lévy desde un nido
    aleatorio  $F_j = f(x_j)$ 
    Seleccionar un nido  $i$  aleatoriamente
    if ( $F_j > F_i$ ) then
         $x_i \leftarrow x_j$ 
         $F_i \leftarrow F_j$ 
    end if
    Abandonar una fracción pa de los peores nidos
    Construir nuevos nidos en nuevos lugares a través de vuelos Lévy para
    reemplazar los nidos perdidos
    Evaluar la aptitud de los nuevos nidos y clasificar todas las soluciones
end while

```

Figura 2: Pseudocódigo del algoritmo Búsqueda Cuco clásico

### 3.2.3. Modelo para crecimiento potencial de cultivos SUCROS

El modelo de crecimiento potencial de cultivos SUCROS ha sido extensamente descrito en Goudriaan y Van Laar, (1994) y Van Laar *et al.*, (1997). Este modelo dinámico predice el comportamiento de biomasa, índice de área foliar y estado de desarrollo a partir de procesos fundamentales como son la fotosíntesis y la respiración del cultivo. El modelo es para crecimiento potencial, ya que supone que, las variables ambientales que afectan los procesos fundamentales son solamente la cantidad de radiación solar, la temperatura así como

las características genéticas del cultivo, todas las otras variables (agua, nutrientes, etc.) se consideran óptimas (Van Ittersum *et al.*, 2003). Las variables de entrada del modelo SUCROS consideradas en este trabajo son: Radiación global ( $Jm^{-2}d^{-1}$ ), Temperatura mínima diaria ( $^{\circ}C$ ) y Temperatura máxima diaria ( $^{\circ}C$ ) y, las variables de salida fueron: Peso seco total ( $gm^{-2}$ ), Peso seco de tallos ( $gm^{-2}$ ), Peso seco de frutos ( $gm^{-2}$ ), Peso seco de hojas ( $gm^{-2}$ ) e Índice de área foliar ( $m^2m^{-2}$ ). En el Cuadro 1 se presenta la relación de los parámetros del modelo SUCROS, su descripción, unidades y valor nominal. El problema de estimación consiste en dadas las mediciones de algunas de las variables que el modelo predice, determinar los valores de los parámetros que permitan lograr el mejor ajuste entre las predicciones y mediciones.

### 3.2.4. Función a optimizar y características de los algoritmos

La función a minimizar fue:

$$p^* = \arg \min f(p) \quad (6)$$

$$f(p) = \sum_{k=1}^M \sum_{i=1}^N w_k (y_{i,k} - \hat{y}_{i,k})^2, \quad (7)$$

donde  $y_{i,k}$  representa la variable medida, biomasa total, biomasa de tallos, biomasa de frutos, biomasa foliar (hojas verdes y secas) así como el índice de área foliar. La variable  $\hat{y}_{i,k}$  representa los valores estimados por el modelo SUCROS,  $w_k$  es un peso asociado con cada variable para hacerlas equiparables. Los coeficientes fueron 1, 1, 1, 1 y 100 respectivamente con el propósito de mantener una relación proporcional entre variables y poderlas comparar.  $N$  es el número de muestreos durante el periodo de cultivo y  $M$  es el número de variables consideradas en la función objetivo. El vector de parámetros estimados consideró los 25 parámetros del modelo (Cuadro 1). Los algoritmos se ejecutaron durante 1000 iteraciones y se efectuaron 20 optimizaciones con inicializaciones aleatorias de las poblaciones dentro de un intervalo del 20% del valor nominal de los parámetros como límites superior e inferior. Los parámetros del algoritmo de evolución diferencial fueron: NP=50, F=0.5 y CR=0.5, como se recomienda en (Price *et al.*, 2005 y Das y Suganthan, 2011). Para el caso del algoritmo de búsqueda cuco clásico los parámetros fueron NP=25 y  $pa=0.25$  como se recomienda en (Yang y Deb, 2009, 2010) y para el algoritmo búsqueda cuco modificado los parámetros fueron NP=80 (valor obtenido experimentalmente),  $pa=0.75$  y SA=0.01 como se recomienda en Walton *et al.* (2011).

```

A ← Máxima longitud de paso de Lévy
φ ← Cociente aureo
Inicializar una población de n nidos hospederos  $x_i, i = 1, 2, \dots, n$ 
for  $x_i$  do
    Calcular la aptitud  $F_i = f(x_i)$ 
end for
Generación  $G \leftarrow 1$ 
while número de evaluaciones de la función objetivo
    < máximo número de evaluaciones do       $G \leftarrow G + 1$ 
    Clasificar nidos por orden de aptitud
    for todos los nidos a ser abandonados do
    Posición actual de  $x_i$ 
    Calcular el tamaño de paso del vuelo de Lévy  $\alpha = A/\sqrt{G}$ 
    Desarrollar vuelos de Lévy desde  $x_i$  para generar nuevos huevos  $x_k$ 
     $x_i \leftarrow x_k$ 
     $F_i \leftarrow f(x_i)$ 
    end for
    for los mejores nidos do
        Actualizar la posición  $x_i$ 
        Seleccionar aleatoriamente otro nido  $x_j$  de los mejores nidos
        if  $x_i = x_j$  then
            Calcular tamaño de paso del vuelo con  $\alpha \leftarrow A/G^2$ 
            Realizar un vuelo de Lévy desde  $x_i$  para generar un nuevo huevo  $x_k$ 
             $F_k = f(x_k)$ 
            Seleccionar aleatoriamente un nido  $l$  de todos los nidos disponibles
            if ( $F_k > F_l$ ) do
                 $x_i \leftarrow x_j$ 
                 $F_l \leftarrow F_k$ 
            end if
        else       $dx = |x_i - x_j|/\varphi$ 
            Mover la distancia  $dx$  desde el peor nido hacia el mejor nido para encontrar  $x_k$ 
             $F_k = f(x_k)$ 
            Seleccionar aleatoriamente un nido  $l$  de todos los nidos disponibles
            if ( $F_k > F_l$ ) do
                 $x_i \leftarrow x_j$ 
                 $F_l \leftarrow F_k$ 
            end if
        end if
    end for
end while

```

Figura 3: Pseudocódigo del algoritmo Búsqueda Cuco Modificado

### 3.3. Resultados y discusión

El Cuadro 2 muestra el rendimiento global promedio de las 20 optimizaciones realizadas para los tres algoritmos estudiados. De acuerdo con los valores de la función objetivo  $f(p)$ ,

Cuadro 1: Parámetros del modelo SUCROS para crecimiento potencial de cultivos.

Parámetro	Símbolo	Unidades	Valor nominal
Tasa de crecimiento relativa durante la fase de crecimiento exponencial del área foliar	RGRL	$^{\circ}C^{-1}d^{-1}$	0.0294
Temperatura base para el crecimiento de área foliar	TBASE	$^{\circ}C$	10.00
Área foliar específica	SLA	$m^2g^{-1}$	0.022
Tasa de asimilación potencial de $CO_2$ a saturación de luz para hojas individuales	AMX	$gm^{-2}h^{-1}$	0.0011
Eficiencia inicial del uso de luz para hojas individuales	EFF	$gJ^{-1}$	$12,5 \times 10^{-6}$
Coefficiente de extinción de la radiación del follaje	KDF	$gg^{-1}$	0.650
Coefficiente de dispersión de las hojas para PAR	SCP	adimensional	0.200
Constante de la función de respiración de mantenimiento	Q10	adimensional	2.000
Temperatura de referencia para la respiración de mantenimiento	TREF	$^{\circ}C$	25.00
Coefficiente de respiración de mantenimiento para las hojas	MAINLV	$gg^{-1}d^{-1}$	0.030
Coefficiente de respiración de mantenimiento para los tallos	MAINST	$gg^{-1}d^{-1}$	0.015
Coefficiente de respiración de mantenimiento para las raíces	MAINRT	$gg^{-1}d^{-1}$	0.015
Coefficiente de respiración de mantenimiento para órganos	MAINSO	$gg^{-1}d^{-1}$	0.010
Requerimiento de asimilados para la producción de biomasa foliar	ASRQLV	$gg^{-1}$	1.463
Requerimiento de asimilados para la producción de biomasa de tallos	ASRQST	$gg^{-1}$	1.513
Requerimiento de asimilados para la producción de biomasa de raíces	ASRQRT	$gg^{-1}$	1.444
Requerimiento de asimilados para la producción de biomasa de órganos	ASRQSO	$gg^{-1}$	1.490
Índice de área foliar crítico antes de que ocurra la muerte por auto-sombreado	LAICR	$m^2m^{-2}$	4.000
Fracción de peso de tallos trasladado a los órganos de almacenamiento	FRTRL	—————	0.200
Fracción de masa de carbono en las hojas	CFLV	$gg^{-1}$	0.459
Fracción de masa de carbono en los tallos	CFST	$gg^{-1}$	0.494
Fracción de masa de carbono en las raíces	CFRT	$gg^{-1}$	0.467
Fracción de masa de carbono en los órganos de almacenamiento	CFSO	$gg^{-1}$	0.471
Parámetro para determinar el incremento en la tasa de mortalidad relativa en hojas	FRDR	—————	1.000
Factor de conversión de hidratos de carbono del tallo en glucosa	CONVL	—————	0.947

el algoritmo de evolución diferencial (DE) fue el método más efectivo ya que no solo alcanzó el valor más cercano a cero, sino que también mostró la desviación estándar más pequeña. Por otro lado, el algoritmo Cuco clásico (CS) superó en efectividad y eficacia al algoritmo Cuco modificado (MCS) mostrando una desviación estándar más pequeña y un valor de la función objetivo cercano al mínimo alcanzado por el algoritmo DE, resultado similar al trabajo realizado por Civicioglu y Besdok, (2013). La eficiencia del algoritmo CS fue ligeramente mayor que la del algoritmo DE, pero de acuerdo a los resultados arrojados por el análisis estadístico y de varianza (ANDEVA), este no es significativo.

**Cuadro 2:** Valor mínimo de la función objetivo ( $f(p)$ ) y número de veces que fue evaluada la función objetivo (NEF) (el número entre paréntesis es la desviación estandar).

	Evolución Diferencial	Búsqueda Cuco	Búsqueda Cuco Modificado
$f(p)$	4886.1391 ( $1,74 \times 10^{-4}$ )	4894.9147 (11.3344)	4941.9907 (19.3086)
NEF	50050 (0.0000)	50000 (0.0000)	80000 (0.0000)

El Cuadro 3 muestra los valores promedio de los parámetros del modelo SUCROS estimados por los tres algoritmos. La desviación estándar más pequeña para cada variable estimada fue casi en su totalidad obtenida por el algoritmo DE a excepción de LAICR, CFLV, CFRT y CFSO. Esto significa que el algoritmo DE tuvo el mejor desempeño en el proceso de estimación. De acuerdo con el análisis de varianza de un solo factor (ANDEVA) el desempeño de los tres algoritmos es significativamente diferente ( $\alpha = 0,05$ ) (Cuadro 3). Tomando en cuenta el proceso de comparación múltiple de Tukey-Kramer el algoritmo CS tuvo un rendimiento similar al DE pero diferente al del algoritmo MCS. Los resultados arrojados por el análisis muestran diferencias estadísticas significativas en la mayoría de los parámetros estimados a excepción de EFF, KDF, ASRQLV, LAICR, CFLV, CFRT y CFSO. Esto significa que son más confiables y precisas las estimaciones obtenidas mediante el algoritmo DE.

Los resultados de las simulaciones llevadas a cabo mediante el modelo SUCROS, usando las estimaciones de los tres algoritmos se muestran en la Figura 4. Se puede observar que en la estimación del peso seco total y peso seco de tallos hubo una sub-estimación y en peso seco de frutos, peso seco de hojas e índice de área foliar hubo sobre-estimación del modelo. En general puede apreciarse un mejor ajuste en el caso de las estimaciones de parámetros obtenidas mediante el algoritmo DE.

En el Cuadro 4 se presentan los resultados estadísticos para cada algoritmo de opti-

**Cuadro 3:** Valores promedio estimados de los parámetros del modelo SUCROS mediante los tres algoritmos de optimización global y resultados del ANDEVA (\*Diferencia significativa  $\alpha = 0,05$ ).

Parámetro	Evolución Diferencial	Búsqueda Cuco	Búsqueda Cuco Modificado	F	$p > F$
RGRL	0.0353 ( $1,58 \times 10^{-15}$ )	0.0315 ( $3,94 \times 10^{-3}$ )	0.0287 ( $1,02 \times 10^{-3}$ )	41.04*	$9,1 \times 10^{-12}$
TBASE	11.8493 ( $1,20 \times 10^{-6}$ )	11.0653 (0.8943)	10.620 ( $2,44 \times 10^{-1}$ )	30.21*	$1,13 \times 10^{-9}$
SLA	0.0176 ( $9,53 \times 10^{-11}$ )	0.0176 ( $6,06 \times 10^{-6}$ )	0.0176 ( $1,02 \times 10^{-6}$ )	5.7*	0.0056
AMX	0.00088 ( $1,28 \times 10^{-10}$ )	0.00088 ( $6,45 \times 10^{-6}$ )	$9,1 \times 10^{-4}$ ( $1,8 \times 10^{-6}$ )	8.59*	0.0006
EFF	$1 \times 10^{-5}$ ( $2,39 \times 10^{-13}$ )	$1 \times 10^{-5}$ ( $5,33 \times 10^{-9}$ )	$1 \times 10^{-5}$ ( $6,12 \times 10^{-10}$ )	2.48	0.0928
KDF	0.52 ( $1,65 \times 10^{-8}$ )	0.5204 ( $9,1 \times 10^{-4}$ )	0.5213 ( $3,27 \times 10^{-3}$ )	2.9	0.0631
SCP	0.24 ( $2,62 \times 10^{-16}$ )	0.2395 ( $4,86 \times 10^{-4}$ )	0.2392 ( $4,86 \times 10^{-4}$ )	9.78*	0.0002
Q10	1.6 ( $2,24 \times 10^{-15}$ )	1.6009 ( $1,35 \times 10^{-3}$ )	1.6008 ( $5,72 \times 10^{-4}$ )	6.06*	0.0041
TREF	20 (0.0000)	20.0026 ( $5,79 \times 10^{-3}$ )	20.002 ( $1,43 \times 10^{-3}$ )	7.21*	0.0016
MAINLV	0.036 ( $7,18 \times 10^{-17}$ )	0.0359 ( $8,8 \times 10^{-5}$ )	0.0353 ( $5,77 \times 10^{-4}$ )	12.67*	$2,8 \times 10^{-5}$
MAINST	0.018 ( $3,55 \times 10^{-18}$ )	0.0179 ( $2,08 \times 10^{-5}$ )	0.0179 ( $2,74 \times 10^{-6}$ )	3.95*	0.0248
MAINRT	0.018 ( $6,61 \times 10^{-17}$ )	0.0178 ( $3,83 \times 10^{-4}$ )	0.0164 ( $8,53 \times 10^{-4}$ )	54.13*	$6,67 \times 10^{-14}$
MAINSO	0.012 ( $6,51 \times 10^{-11}$ )	0.0119 ( $1,26 \times 10^{-5}$ )	0.0119 ( $4,41 \times 10^{-6}$ )	5.5*	0.0065
ASRQLV	1.1704 ( $2,27 \times 10^{-16}$ )	1.1711 ( $2,7 \times 10^{-3}$ )	1.1707 ( $2,34 \times 10^{-4}$ )	1.74	0.1849
ASRQST	1.2104 ( $5,83 \times 10^{-9}$ )	1.2105 ( $2,42 \times 10^{-4}$ )	1.2105 ( $1,13 \times 10^{-4}$ )	10.97*	$9,31 \times 10^{-5}$
ASRQRT	1.1552 ( $6,35 \times 10^{-8}$ )	1.1578 ( $7,99 \times 10^{-3}$ )	1.1566 ( $1,36 \times 10^{-3}$ )	12.92*	$2,36 \times 10^{-5}$
ASRQSO	1.788 ( $5,15 \times 10^{-9}$ )	1.7875 ( $8,37 \times 10^{-4}$ )	1.7879 ( $5,05 \times 10^{-5}$ )	4.62*	0.0138
LAICR	3.9952 (0.5950)	4.0018 (0.6385)	3.8583 (0.2310)	0.09	0.9123
FRTRL	0.16 ( $2,33 \times 10^{-8}$ )	0.1601 ( $2,58 \times 10^{-4}$ )	0.1602 ( $1,57 \times 10^{-4}$ )	16.86*	$1,76 \times 10^{-6}$
CFLV	0.4688 ( $6,63 \times 10^{-2}$ )	0.4706 ( $6,22 \times 10^{-2}$ )	0.4523 ( $2,56 \times 10^{-2}$ )	0.04	0.9587
CFST	0.3952 ( $2,19 \times 10^{-9}$ )	0.3958 ( $1,85 \times 10^{-3}$ )	0.3975 ( $1,40 \times 10^{-3}$ )	22.23*	$7,28 \times 10^{-8}$
CFRT	0.4601 ( $6,43 \times 10^{-3}$ )	0.4604 ( $8 \times 10^{-2}$ )	0.4726 ( $2,97 \times 10^{-2}$ )	2.1	0.1316
CFSO	0.4694 ( $6,92 \times 10^{-2}$ )	0.4839 ( $7,6 \times 10^{-2}$ )	0.4767 ( $2,44 \times 10^{-2}$ )	0.27	0.7679
FRDR	1.1999 ( $3,24 \times 10^{-8}$ )	1.1992 ( $1,6 \times 10^{-3}$ )	1.1986 ( $8,40 \times 10^{-4}$ )	9.12*	0.0004
CONVL	0.7576 ( $6,26 \times 10^{-8}$ )	0.7577 ( $3,44 \times 10^{-4}$ )	0.7609 ( $1,74 \times 10^{-3}$ )	8.74*	0.0005
f(p)	————	————	————	107.97*	$4,11 \times 10^{-20}$

mización y para cada variable de salida del modelo SUCROS. Si se toma de referencia el sesgo generado por los métodos de optimización se corrobora la sub y sobre-estimación del modelo para las diferentes variables de salida. Para el peso seco total [a)] se puede observar que MCS generó una sub-estimación de la variable peso seco total y los algoritmos restantes sobre-estimaron esta variable. En el caso del peso seco de tallos [b)] los tres algoritmos mostraron una sub-estimación. Sin embargo, las variables peso seco de frutos [c)], peso seco de hojas [d)] y el índice de área foliar [e)] fueron sobre-estimados. De acuerdo con los estadísticos MSE, RMSE y MAE el algoritmo de evolución diferencial fue ligeramente mejor que el algoritmo CS y este último fue mejor que el algoritmo MCS. En general, los tres algoritmos presentaron una eficiencia aceptable en el proceso de calibración del modelo a excepción de la estimación del peso seco de frutos y hojas, en el cual se tuvo un rendimiento bajo. El coeficiente de correlación (r) nos indica la existencia de una relación lineal entre datos medidos y estimados, siendo una buena estimación una r cercana a 1 (Wallach, *et al.*, 2006), en este caso se observa un desempeño de los algoritmos similar en la mayoría de los casos.

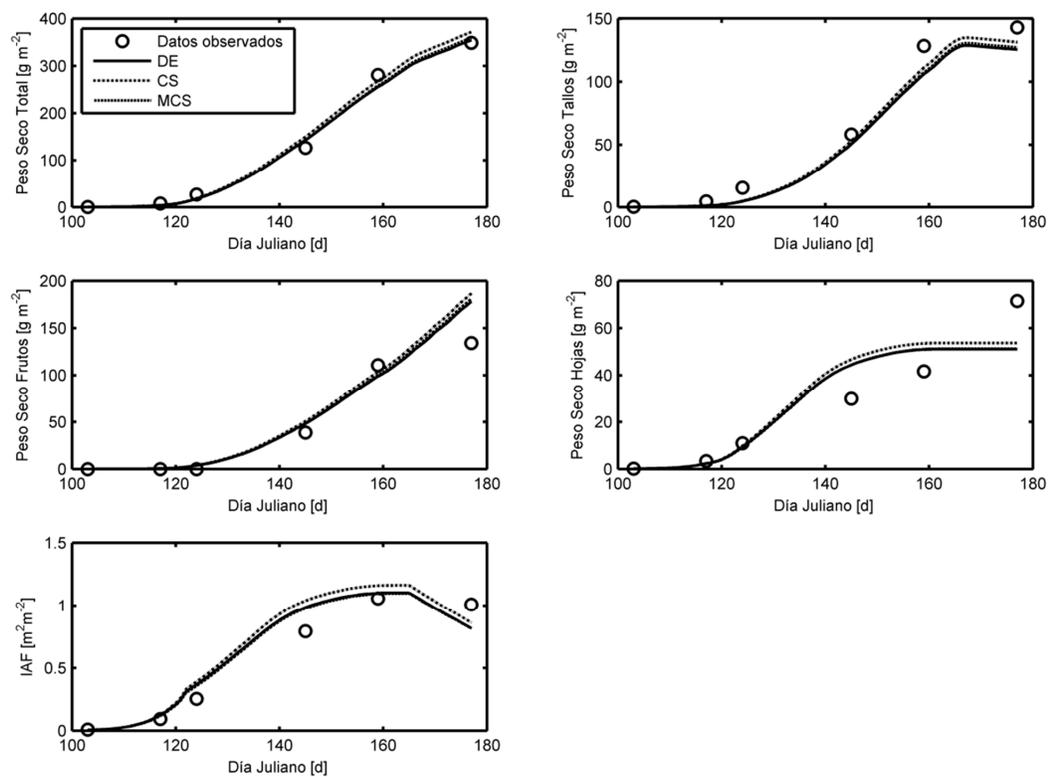


Figura 4: Comparación de estimaciones del modelo SUCROS calibrado con tres métodos de optimización global para las cinco variables de salida consideradas.

**Cuadro 4:** Resultados estadísticos de las cinco variables de salida del modelo SUCROS calibrado con tres métodos de optimización global.

Algoritmo	MSE	RMSE	MAE	Sesgo	Eficiencia	r
<b>a) Peso seco total</b>						
DE	166.5840	12.9067	9.9436	2.6312	0.9911	0.9957
CS	162.6751	12.7544	10.4245	1.3861	0.9914	0.9958
MCS	217.9539	14.7633	11.7852	-3.7135	0.9884	0.9957
<b>b) Peso seco tallos</b>						
DE	167.2770	12.9320	10.3838	10.3838	0.9503	0.9978
CS	148.1333	12.1710	9.8593	9.8593	0.9559	0.9978
MCS	97.4174	9.8700	8.0516	8.0516	0.9710	0.9978
<b>c) Peso seco de frutos</b>						
DE	360.0775	18.9757	11.5079	-7.6266	0.8823	0.9728
CS	399.1242	19.9781	11.8142	-8.3299	0.8695	0.9724
MCS	492.2433	22.1866	12.5807	-10.3124	0.8391	0.9728
<b>d) Peso seco de hojas</b>						
DE	118.9583	10.9068	7.7999	-0.1276	0.8087	0.8994
CS	118.6315	10.8918	7.8293	-0.1450	0.8092	0.8997
MCS	123.3043	11.1042	8.0993	-1.4544	0.8017	0.8992
<b>e) Índice de área foliar</b>						
DE	0.0150	0.1225	0.0959	-0.0307	0.9194	0.9617
CS	0.0142	0.1190	0.0915	-0.0256	0.9238	0.9631
MCS	0.0188	0.1370	0.1116	-0.0611	0.8991	0.9615

† MSE: Error Cuadrático Medio; RMSE: Raíz del Error Cuadrático Medio; MAE: Error Medio Promedio; r: Coeficiente de correlación; DE: Evolución Diferencial; CS: Búsqueda Cuco clásico; MSC: Búsqueda Cuco Modificado.

### **3.4. Conclusiones**

El algoritmo evolutivo y dos algoritmos bio-inspirados permitieron estimar todos los parámetros del modelo SUCROS para crecimiento potencial de cultivos. Aunque los resultados fueron bastante similares el algoritmo de evolución diferencial mostró el mejor desempeño ya que obtuvo el valor más pequeño de la función objetivo. En cuanto a eficiencia, los tres algoritmos presentaron un desempeño similar. De acuerdo con los resultados obtenidos los algoritmos bio-inspirados son confiables y pueden ser aplicados en el proceso de estimación de parámetros de modelos de cultivos. Sin embargo, es necesario investigar con mayor detalle si los resultados obtenidos en la presente investigación son corroborables en el caso de una estimación de parámetros en modelos más complejos con varias decenas o cientos de parámetros a ser determinados.

---

# CAPÍTULO 4

---

## ESTIMACIÓN DE PARÁMETROS PARA UN MODELO DE CRECIMIENTO DE CULTIVOS USANDO ALGORITMOS EVOLUTIVOS Y BIO-INSPIRADOS. PARTE II

Elmer César Trejo Zúñiga<sup>1</sup>, Irineo L. López Cruz<sup>1</sup>, Agustín Ruíz García<sup>1</sup>

### Resumen

Todos los modelos dinámicos para crecimiento y desarrollo de cultivos tienen varios parámetros cuyos valores se determinan, por lo general, mediante el uso de mediciones procedentes del sistema real. El problema de estimación de parámetros se plantea como un problema de optimización y se utilizan algoritmos de búsqueda para resolverlo. Sin embargo, debido a que generalmente el modelo es no-lineal el problema de optimización es probable que sea multimodal y por lo tanto los métodos clásicos de búsqueda local fallan en la localización del óptimo global y, como consecuencia los parámetros del modelo podrían ser estimados incorrectamente. En este trabajo se presenta una comparación entre algoritmos evolutivos (EA) y algoritmos bio-inspirados (BIA), considerados como métodos de optimización global, tales como: evolución diferencial (DE), estrategias evolutivas con matriz de covarianza adaptable (CMA-ES), optimización por enjambre de partículas (PSO) y colonia artificial de abejas (ABC) en la estimación de parámetros del modelo de crecimiento de cultivos SUCROS (a Simple and Universal CROp Growth Simulator). Posteriormente, el modelo SUCROS para crecimiento potencial se aplicó a un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. Ex Horm.) usando datos procedentes de un experimento llevado a cabo en Chapingo, México. El objetivo fue determinar qué algoritmo genera valores de los parámetros que dan la mejor predicción del modelo. Un análisis de varianza (ANDEVA)

---

<sup>1</sup>Postgrado en Ingeniería Agrícola y Uso Integral del Agua, Universidad Autónoma Chapingo, Km. 38.5 Carr. México-Texcoco. C.P. 56230. Chapingo, Estado de México.

\* Capítulo ampliado basado en Trejo-Zúñiga, E. C., I. L. López-Cruz, and A. Ruíz-García. 2013. Parameter estimation for crop growth model using evolutionary and bio-inspired algorithms. Enviado a la revista Applied Soft Computing.

se llevó a cabo para evaluar estadísticamente la eficiencia y la eficacia de los algoritmos estudiados. La eficiencia del algoritmo se evaluó contando el número de veces que la función objetivo es requerida para aproximarse a un valor óptimo. Por otra parte, la eficacia se evaluó contando el número de veces que el algoritmo convergió a un óptimo. Resultados de la simulación muestran que el algoritmo estándar DE/rand/1/bin logró el mejor desempeño.

**Palabras clave:** Estimación de parámetros, Algoritmos evolutivos, Algoritmos bio-inspirados, modelo SUCROS.

## 4.1. Introducción

Los modelos matemáticos en la agricultura son herramientas poderosas para describir y comprender sistemas complejos. Estos modelos se han utilizado en la agronomía para simular los efectos de los cambios en las características morfológicas y fisiológicas de los cultivos que ayudan en la identificación de ideotipos para diferentes entornos. La estructura de los modelos dinámicos de crecimiento de cultivos consiste en un conjunto de ecuaciones diferenciales ordinarias de primer orden que se caracterizan por la no linealidad, dinámica multivariable, complejidad e incertidumbre (Chalabi y Zhou, 1996). En general, estas ecuaciones tienen un conjunto de coeficientes que representan parámetros fisiológicos cuyos valores deben ser determinados con precisión para obtener un buen ajuste entre predicciones y mediciones. En el proceso de desarrollo de un modelo matemático de crecimiento de cultivos, la estimación de parámetros (calibración) es esencial para conseguir el objetivo antes mencionado. Lo cual implica, la aplicación de algoritmos de optimización, para estimar los valores de los parámetros que afectan más el comportamiento del modelo. El proceso para determinar los valores de los parámetros se puede establecer como un problema de optimización que permite que una variedad de algoritmos puedan ser usados para buscar una solución. En general, se utilizan métodos locales como LSE (Mínimos Cuadrados Ordinarios) y SQP (programación cuadrática secuencial). Sin embargo, estos algoritmos tienen el inconveniente de que conducen a valores inexactos de los parámetros estimados (alta varianza de los estimadores de parámetros) y predicciones del modelo inexactos (sobre-parametrización) cuando se aplican en la estimación de un gran número de parámetros del modelo (mayor a 10). Este problema se debe a que este tipo de algoritmos a menudo no convergen al valor óptimo cuando los parámetros son demasiado numerosos

(Makowski *et al.*, 2006). También, debido a la no-linealidad alta de los modelos de crecimiento de cultivos y una posible dependencia entre los parámetros (epístasis), el problema de optimización puede ser no-convexo o multimodal.

Con el fin de resolver problemas de optimización multi-modal, los métodos de optimización global tales como: algoritmos evolutivos (DE, Differential Evolution y CMA-ES, Covariance Matrix Adaptation Evolution Strategy) y algoritmos inspirados en el comportamiento de sistemas biológicos (PSO, Particle Swarm Optimization y ABC, Artificial Bee Colony) pueden proporcionar buenas aproximaciones al óptimo global (Michalewicz, 1996; Storn y Price, 1997; Eiben y Smith, 2003; Hansen y Kern, 2004; Poli *et al.*, 2007; Karaboga y Ozturk, 2011). En la literatura muchos trabajos dedicados a la estimación de parámetros se han centrado en modelos de crecimiento de cultivos en invernadero, utilizando principalmente algoritmos evolutivos (por ejemplo, Guzmán-Cruz *et al.*, 2009; Herrero *et al.*, 2007; Dai *et al.*, 2009). Sin embargo, se han realizado pocos trabajos para la estimación de parámetros de modelos de crecimiento de cultivos en campo abierto. Ioslovich y Gutman (2007) y Wallach *et al.* (2001) propusieron un procedimiento para la estimación de parámetros de modelos de cultivos calibrados con datos de campo, pero no usaron métodos de optimización global. Pabico *et al.* (1999) formularon un algoritmo genético (GA) para calcular el coeficiente de cultivares de modelos de cultivos. Sin embargo, hoy en día las limitaciones de los algoritmos genéticos como optimizadores son bien conocidos (Salomon, 1996). De acuerdo a la literatura, no se han aplicado algoritmos de optimización bio-inspirados para la estimación de parámetros de modelos de cultivos en campo abierto. El objetivo de la presente investigación fue evaluar el desempeño, a través de la eficiencia y efectividad de los algoritmos DE, CMA-ES, PSO y ABC, para estimar 25 parámetros del modelo SUCROS (a Simple and Universal CROp Growth Simulator) (van Laar *et al.*, 1997) aplicado a un cultivo de tomate de cáscara (*Physalis ixocarpa* Brot. Ex Horm.), con el fin de analizar qué algoritmo genera valores de los parámetros que dan la mejor predicción del modelo, no sólo para aumentar el conocimiento del sistema de cultivo del tomate de cáscara, sino también el potencial de desarrollo de aplicaciones prácticas. La eficiencia o coste computacional de un algoritmo es el número de veces que la función objetivo se evalúa para encontrar un óptimo. La efectividad o el rendimiento de un algoritmo, para alcanzar un óptimo en un problema de optimización multi-modal, se calcula contando el número de veces que el algoritmo converge al mismo óptimo con diferentes valores iniciales para el proceso iterativo (Baritomba y Hendrix, 2005).

## 4.2. Materiales y métodos

### 4.2.1. El modelo SUCROS para crecimiento potencial

El modelo SUCROS (a Simple and Universal CROp growth Simulator) es un modelo mecanicista que explica el crecimiento de cultivos sobre la base de procesos subyacentes, tales como la asimilación de dióxido de carbono ( $CO_2$ ) y la respiración, tanto como la influencia de las condiciones ambientales (Goudriaan y van Laar, 2003). Simula el crecimiento potencial de un cultivo, es decir, la acumulación de materia seca con un amplio suministro de agua y nutrientes, en un entorno libre de plagas, enfermedades y malas hierbas, bajo las condiciones climáticas prevaescentes. El modelo de crecimiento potencial de cultivos se ha descrito ampliamente en (van Laar *et al.*, 1997; Goudriaan y van Laar, 2003) y un breve resumen para destacar sus principales propiedades se da en los Cuadros 1-2. Las siete ecuaciones diferenciales ordinarias que componen el modelo son:

$$\frac{dx_{ed}}{dt} = f(x_{es}, T) \quad (1)$$

$$\frac{dx_{IAF}}{dt} = C_{IAF} - M_{IAF} \quad (2)$$

$$\frac{dx_{hv}}{dt} = C_{foliar} - M_{foliar} \quad (3)$$

$$\frac{dx_{hs}}{dt} = M_{foliar} \quad (4)$$

$$\frac{dx_r}{dt} = (1 - fr_b(x_{ed}))C_{tot} \quad (5)$$

$$\frac{dx_t}{dt} = fr_t(x_{ed})fr_b(x_{ed})C_{tot} \quad (6)$$

$$\frac{dx_{org}}{dt} = fr_{org}(x_{ed})fr_b(x_{ed})C_{tot} \quad (7)$$

donde:  $x_{ed}$  (adimensional) es el estado de desarrollo,  $x_{IAF}$  ( $m^2 m^{-2}$ ),  $x_r$  ( $gr m^{-2}$ ) es la biomasa de raíces,  $x_{hv}$  ( $gr m^{-2}$ ) es la biomasa de hojas secas,  $x_t$  ( $gr m^{-2}$ ) es la biomasa de tallos,  $x_{org}$  ( $gr m^{-2}$ ) se refiere a los órganos de almacenamiento. Las principales ecuaciones que completan el modelo se presentan en el Cuadro 1. Donde:  $\Delta t$  (d) es el intervalo de integración,  $T_{max}$  ( $^{\circ}C$ ) la temperatura máxima diaria,  $T_{min}$  ( $^{\circ}C$ ) la temperatura mínima diaria,  $T_{base}$  ( $^{\circ}C$ ) la temperatura base para desarrollo,  $fr_h(x_{ed})$  (adimensional) fracción de biomasa aérea para hojas,  $fr_b(x_{ed})$  (adimensional) fracción de biomasa total para el follaje,  $M_{hv}$ ,  $M_t$ ,  $M_r$  y  $M_{org}$  ( $g g^{-1} d^{-1}$ ), son los coeficientes de respiración de mantenimiento para

hojas, tallos, raíces y órganos de almacenamiento respectivamente.  $f(x_t, x_{ed}, T_{avgd})$ ,  $f_1(x_{ed})$  y  $f_2(x_{IAF})$  son funciones empíricas.  $f(d_{emerg})$  es una función que reproduce la unidad cuando el tiempo de simulación es igual al día de emergencia del cultivo,  $T_{avgd}$  ( $^{\circ}C$ ), temperatura promedio diaria,  $T_{ref}$  ( $^{\circ}C$ ), es la temperatura de referencia,  $Q10$  es un factor de crecimiento con un incremento de  $10^{\circ}C$  de temperatura. En el Cuadro 2 se presenta la relación de los parámetros del modelo SUCROS, su descripción, unidades, valor nominal e intervalo de incertidumbre (20% del valor nominal).

**Cuadro 1:** Ecuaciones principales del modelo para crecimiento potencial de cultivos SUCROS.

Función	Descripción	Unidades
$C_{IAF} = SLA \cdot C_{foliar}$	Tasa de crecimiento IAF para la fase lineal	$m^2 m^{-2} d^{-1}$
$C_{IAF} = \frac{x_{IAF} \cdot \exp(RGRL \cdot T_{eff} \cdot \Delta t) - 1}{\Delta t}$	Tasa de crecimiento IAF para la fase exponencial	$m^2 m^{-2} d^{-1}$
$C_{foliar} = fr_h(x_{ed}) \cdot fr_b(x_{ed}) \cdot C_{tot}$	Tasa de crecimiento foliar	$g m^{-2} d^{-1}$
$C_{tot} = \frac{A_{tot} - R_m + CONVL \cdot w_{tl} \cdot CFST(\frac{30}{12})}{ASRQ}$	Tasa de crecimiento de la biomasa total	$g m^{-2} d^{-1}$
$A_{tot} = \frac{30}{44} AMX \cdot (1 - \exp(-EFF \cdot PAR_{abs} / AMX))$	Tasa de asimilación de $CO_2$	$g m^{-2} d^{-1}$
$R_m = (MAINLV \cdot x_{hv} + MAINST \cdot x_t + MAINRT \cdot x_r + MAINSO \cdot x_{org}) \cdot f_{resp}(T) \cdot f_{resp}(x_{ed}) \cdot f(d_{emerg})$	Tasa de respiración de mantenimiento	$g m^{-2} d^{-1}$
$f_{resp}(T) = Q10^{\frac{T_{avgd} - T_{ref}}{10}}$	Efecto de la Temperatura sobre la tasa de respiración	—
$f_{resp}(x_{ed}) = \frac{x_{hv}}{x_{hv} + x_{hs}}$	Efecto del estado de desarrollo sobre la tasa de respiración	—
$M_{foliar} = x_{hv} \cdot \frac{M_{IAE}}{x_{IAF}}$	Tasa de mortalidad foliar	$g m^{-2} d^{-1}$
$Mh_{IAF} = x_{IAF} \max(f_1(x_{ed}), f_2(x_{IAF}))$	Tasa de mortalidad de área foliar	$m^2 m^{-2} d^{-1}$
$w_{tl} = FRTRL \cdot x_t \cdot f(x_t, x_{ed}, T_{avgd})$	Tasa de translocación de biomasa de los tallos a órganos de almacenamiento	$g m^{-2} d^{-1}$
$PAR_{abs} = 1 - \exp(-KDF \cdot x_{IAF})$	Función de intercepción de la radiación	$J m^{-2} s^{-1}$
$T_{eff} = \max(0, \frac{1}{2}(T_{max} + T_{min}) - T_{base})$	Temperatura efectiva diaria	$^{\circ}C$

## 4.2.2. Descripción del sitio experimental

El tomate de cáscara (*Physalis ixocarpa* Brot. Ex Horm.) se cultivó durante el verano de 2007 en Chapingo, México ( $19^{\circ} 16' 52'' LN$  y  $99^{\circ} 39' 0'' LW$ ). El lugar tiene un clima templado, con una temporada de lluvias durante el verano y un tiempo de sequía durante

Cuadro 2: Parámetros del modelo SUCROS para crecimiento potencial de cultivos.

Parámetro	Símbolo	Unidades	Valor nominal	Intervalo de incertidumbre	
				Máx	Mín
Tasa de crecimiento relativa durante la fase de crecimiento exponencial del área foliar	RGRL	$(^{\circ}C d)^{-1}$	0.0294	0.0353	0.0235
Temperatura base para el crecimiento de área foliar	TBASE	$^{\circ}C$	10.00	12	8
Área foliar específica	SLA	$m^2 g^{-1}$	0.022	0.0264	0.0176
Tasa de asimilación potencial de $CO_2$ a saturación de luz para hojas individuales	AMX	$g m^{-2} h^{-1}$	0.0011	0.0013	0.00088
Eficiencia inicial del uso de luz para hojas individuales	EFF	$g J^{-1}$	$12,5 \times 10^{-6}$	0.000015	0.00001
Coefficiente de extinción de la radiación del follaje	KDF	$g g^{-1}$	0.650	0.78	0.52
Coefficiente de dispersión de las hojas para PAR	SCP	adimensional	0.200	0.24	0.16
Constante de la función de respiración de mantenimiento	Q10	adimensional	2.000	2.4	1.6
Temperatura de referencia para la respiración de mantenimiento	TREF	$^{\circ}C$	25.00	30	20
Coefficiente de respiración de mantenimiento para las hojas	MAINLV	$g g^{-1} d^{-1}$	0.030	0.036	0.024
Coefficiente de respiración de mantenimiento para los tallos	MAINST	$g g^{-1} d^{-1}$	0.015	0.018	0.012
Coefficiente de respiración de mantenimiento para las raíces	MAINRT	$g g^{-1} d^{-1}$	0.015	0.018	0.012
Coefficiente de respiración de mantenimiento para órganos	MAINSO	$g g^{-1} d^{-1}$	0.010	0.012	0.008
Requerimiento de asimilados para la producción de biomasa foliar	ASRQLV	$g g^{-1}$	1.463	1.7556	1.1704
Requerimiento de asimilados para la producción de biomasa de tallos	ASRQST	$g g^{-1}$	1.513	1.8156	1.2104
Requerimiento de asimilados para la producción de biomasa de raíces	ASRQRT	$g g^{-1}$	1.444	1.7328	1.1552
Requerimiento de asimilados para la producción de biomasa de órganos	ASRQSO	$g g^{-1}$	1.490	1.788	1.192
Índice de área foliar crítico antes de que ocurra la muerte por auto-sombreado	LAICR	$m^2 m^{-2}$	4.000	4.8	3.2
Fracción de peso de tallos trasladado a los órganos de almacenamiento	FRTRL	—————	0.200	0.24	0.16
Fracción de masa de carbono en las hojas	CFLV	$g g^{-1}$	0.459	0.5928	0.3952
Fracción de masa de carbono en los tallos	CFST	$g g^{-1}$	0.494	0.5508	0.3672
Fracción de masa de carbono en las raíces	CFRT	$g g^{-1}$	0.467	0.5604	0.3736
Fracción de masa de carbono en los órganos de almacenamiento	CFSO	$g g^{-1}$	0.471	0.5652	0.3768
Parámetro para determinar el incremento en la tasa de mortalidad relativa en hojas	FRDR	—————	1.000	1.2	0.8
Factor de conversión de hidratos de carbono del tallo en glucosa	CONVL	—————	0.947	1.1364	0.7576

el invierno. La temperatura media anual es de  $15,5^{\circ}C$  y la precipitación anual de 664 mm. La variedad de tomate de cáscara fue CHF1-Chapingo proveniente de los semilleros de plantas de la Universidad de Chapingo. El día del trasplante fue el 30 de marzo con una densidad de 16,122 plantas  $ha^{-1}$ . Con el fin de garantizar las condiciones de crecimiento potencial, se aplicó el riego usando como referencia la evapotranspiración calculada con el método de Penman-Monteith y un sistema 100% automatizado de fertigración. La fórmula de fertilización fue 200-110-50 que es suficiente para evitar la limitación de ni-

trógeno y fósforo. Los fertilizantes se aplicaron cada semana. Una estación meteorológica (GroWeather, Davis, EE.UU.) registró diariamente las variables climáticas radiación solar global ( $^{\circ}\text{C}$ ), temperaturas mínimas y máximas ( $^{\circ}\text{C}$ ) (Figura 1). Un método destructivo se utilizó para calcular la biomasa y el índice de área foliar de seis muestras (Día Juliano 103, 117, 124, 145, 159 y 177) durante el período de crecimiento. En cada muestreo 3-10 plantas de tomate de cáscara se recolectaron al azar (López-López *et al.*, 2009).

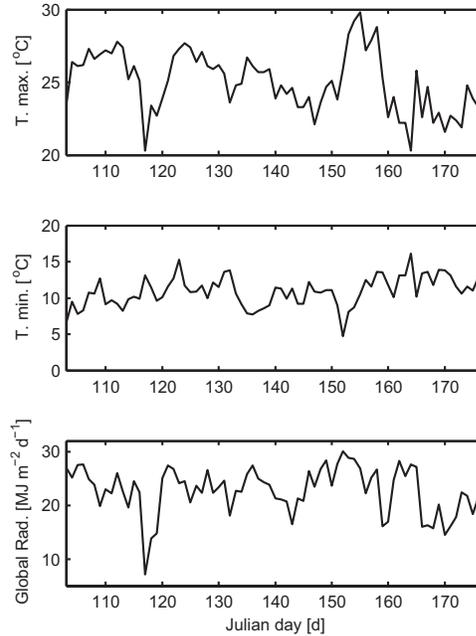


Figura 1: Variables de entrada del modelo SUCROS.

### 4.2.3. Descripción del problema a optimizar

Con referencia al problema de optimización de parámetros, una función objetivo se expresa comúnmente como:

$$p^* = \arg \min f(p) \quad (8)$$

$$f(p) = \sum_{j=1}^M \sum_{i=1}^N w_j (y_{i,j} - \hat{y}_{i,j})^2 \quad (9)$$

donde  $y_{i,j}$  representa la variable medida (biomasa total, tallos, frutos y foliar (hojas verdes y hojas muertas) y el índice de área foliar). La variable  $\hat{y}_{i,j}$  representa los valores

pronosticados por el modelo SUCROS.  $w_j$  es el peso asociado a cada variable para hacerlos equiparables. Los coeficientes fueron seleccionados como 1, 1, 1, 1, 100, respectivamente para poder ser comparados en igualdad de condiciones. N es el número de muestras durante el período de crecimiento ( $N = 6$ ). M es el número de variables de salida considerados en la función objetivo ( $M = 5$ ). El vector  $\mathbf{p}$  de parámetros estimados es (Tabla 2): [RGRL, TBASE, SLA, AMX, FEP, KDF, SCP, Q10, TREF, MAINLV, MAINST, MAINRT, MAINSO, ASRQLV, ASRQST, ASRQRT, ASRQSO, LAICR, FRTRL, CFST, CFLV, CFRT, CFSO, FRDR, CONVL]. Las variables de salida considerados en este trabajo son: peso seco total ( $g m^{-2}$ ), peso seco de tallos ( $g m^{-2}$ ), peso seco de frutos ( $g m^{-2}$ ), peso seco de hojas ( $g m^{-2}$ ) y el IAF ( $m^2 m^{-2}$ ). La ecuación (8) indica cuán bien un modelo simula los datos observados al minimizar el error cuadrático medio entre las mediciones y predicciones.

#### 4.2.4. Ajustes e implementación de los algoritmos

Para estimar los valores de los 25 parámetros (D, dimensión del problema) del modelo SUCROS, se aplicaron los algoritmos de evolución diferencial (DE), Estrategias evolutivas con matriz de covarianza adaptable (CMA-ES), Optimización por enjambre de partículas (PSO) y Colonia artificial de abejas (ABC) como se describen en (Michalewicz, 1996; Storn y Price, 1997; Eiben y Smith, 2003; Hansen y Kern, 2004; Poli *et al.*, 2007; Karaboga y Ozturk, 2011). Se realizaron un total de 20 ejecuciones seleccionando al azar un vector de parámetros inicial (se propuso un 20% del valor nominal de los parámetros como límites superior e inferior, Cuadro 2). En todos los experimentos, se seleccionaron los valores de los parámetros comunes que se utilizan en cada algoritmo de búsqueda global, tales como el número máximo de generaciones (MaxIter = 1000). Los otros parámetros específicos de cada algoritmo se seleccionaron como sigue:

**DE:** En el algoritmo DE, F es una constante real que afecta a la variación diferencial entre dos soluciones y se establece en 0,5 en nuestros experimentos. El valor de la tasa de cruzamiento (CR), que controla el cambio de la diversidad de la población, fue elegido en 0,5. El tamaño de la población fue de 50 después de que varias pruebas fueran realizadas con diferentes poblaciones, 50, 75, 150, 200 y 250. En este caso se aplicó el algoritmo DE/rand/1/bin estándar (Storn y Price, 1997; Das y Suganthan, 2011).

**CMA-ES:**  $\sigma$  determina las coordenadas de las desviaciones estándar iniciales para la búsqueda, fue elegido en 25 (Hansen y Kern, 2004). El tamaño de la población fue de 250

después de realizar varias pruebas con diferentes poblaciones, 50, 75, 150, 200 y 250. En este proyecto se utilizó el archivo script *cmaes.m* de Matlab, versión 2.35, disponible en la página web de Internet [http://www.Iri.fr/hansen/cmaes\\_inmatlab.html](http://www.Iri.fr/hansen/cmaes_inmatlab.html).

**PSO:** Los componentes cognitivo y social ( $\phi_1$  y  $\phi_2$ ) son constantes que se pueden utilizar para cambiar la ponderación entre experiencia personal y grupal de la población, respectivamente. En nuestros experimentos estos componentes cognitivo y social se establecieron en 1.53. El peso inercial ( $w$ ), el cual determina como la velocidad previa de la partícula influye en la velocidad en la siguiente iteración, fue ajustada a 1. El tamaño de la población fue de 250. Este valor se obtuvo después de que varias pruebas se llevaron a cabo con diferentes poblaciones, 25, 50, 75, 125, 200, 250 y 300.

**ABC:** El ABC básico utiliza sólo un parámetro de control, que se denomina límite de abandono. Una fuente de alimento no será explotado más y se supone que será abandonada cuando se excede el límite para la fuente. Esto significa que la solución obtenida y que excede el valor límite no se puede mejorar más. Hemos definido una relación para el valor límite con la dimensión del problema y el tamaño de la colonia:  $limit = PS * D$ , donde  $D$  es la dimensión del problema y  $PS$  es el tamaño de la colonia (número de fuentes de alimento o abejas obreras). En nuestro experimento  $PS$  fue de 250. Este valor se obtuvo después de realizar varias pruebas con diferente cantidad de individuos en la colonia, 25, 50, 75, 125, 150, 200, 250 y 300. El programa de cómputo ABC utilizado en este estudio está disponible en Internet (<http://mf.erciyes.edu.tr/abc>).

**LSE:** LSE resuelve el problema de mínimos cuadrados no-lineal. Se utilizó el algoritmo *lsqnonlin.m* programado en las herramientas de optimización de Matlab, que implementa dos algoritmos diferentes: el algoritmo “trust region reflective” y “Levenberg-Marquardt” (The MathWorks Inc., 1999-2006). En este experimento se utilizó la opción Levenberg-Marquardt. El número máximo de generaciones fue 20.000 y el número máximo de evaluaciones de la función objetivo fue 150.000.

Los valores de los parámetros de los algoritmos evolutivos y bio-inspirados fueron derivados empíricamente después de realizar una serie de experimentos en los que cada uno de ellos tuvo el mejor desempeño para el problema en cuestión. Las reglas heurísticas encontradas en la literatura también se tomaron en consideración. El número de individuos de la población utilizados para cada algoritmo descrito anteriormente es necesario para obtener una buena convergencia al óptimo global, una población más pequeña genera divergencia y una población más grande no mejora el rendimiento de los algoritmos.

### 4.3. Resultados y discusión

El Cuadro 3 muestra el rendimiento global observado de los cinco algoritmos estudiados durante 20 optimizaciones. De acuerdo con los correspondientes valores promedio de la función objetivo, el algoritmo de evolución diferencial fue el método más eficaz ya que alcanzó no sólo el valor más bajo, sino que también muestra la menor desviación estándar. En contraste, el método local LSE obtuvo la peor solución y también fue el menos eficaz. Los cuatro métodos globales mostraron un comportamiento similar en cuanto a eficacia. Por otro lado, teniendo en cuenta el número promedio de evaluaciones de la función objetivo (NEF) necesarios para resolver el problema de estimación de parámetros, al parecer, el método local resultó ser el más eficiente. Sin embargo, este resultado es engañoso ya que la convergencia del algoritmo LSE probablemente fue a un mínimo local, debido a su alta desviación estándar. Los resultados que se muestran en el Cuadro 3 sobre algoritmos evolutivos y bio -inspirados están de acuerdo con los resultados obtenidos en el proceso de comparación de los algoritmos DE con PSO y CMA-ES (Civicioglu y Besdok, 2013). Teniendo en cuenta que la función objetivo de mínimos cuadrados ponderados (ecuación 9) se puede separar o descomponer, el algoritmo de evolución diferencial tuvo un mejor rendimiento que CMA-ES. Tal vez este hecho puede explicarse por la baja complejidad del espacio asociado con DE en comparación con el espacio que le corresponde a CMA-ES (Hansen y Ostermeier, 2001). Debido a que la función (9) es separable y unimodal, es lógico que se esperase un buen desempeño del algoritmo DE con un valor relativamente pequeño para el parámetro de probabilidad de cruzamiento (Das y Suganthan, 2011) . Sin embargo, se obtuvo el mejor rendimiento en términos de velocidad utilizando los valores estándar para el parámetro CR. El uso de un valor relativamente mayor de cruza ( $CR = 0.5$ ) explica por qué DE superó el algoritmo ABC que es bien conocido por ser muy eficiente en la solución de funciones separables (Karaboga y Basturk, 2007; Karaboga y Akay, 2009; Civicioglu y Besdok, 2013). El presente estudio también muestra que, aunque la función objetivo es separable, es difícil de optimizar por un método de optimización local, tal como la estimación por mínimos cuadrados debido a la sensibilidad del LSE a los valores iniciales y por lo tanto converge a mínimos locales, como se confirma por la desviación estándar más alta obtenida por este algoritmo (Cuadro 3). Una segunda limitación de un método local es el problema de sobre-parametrización. Por el contrario, los dos algoritmos bio-inspirados y DE fueron bastante similares en cuanto a eficiencia.

**Cuadro 3:** Valor mínimo promedio ( $f(p)$ ) y número de veces que fue evaluada la función objetivo (NEF) (Los números entre paréntesis son la desviación estandar).

	DE	CMA-ES	PSO	ABC	LSE
$f(p)$	4886.1391 ( $1.74 \times 10^{-4}$ )	5205.3650 (50.1159)	5026.4773 (272.4644)	4949.6307 (43.4292)	5562.8932 (102.1990)
NEF	50050 (0.0000)	127633 (5984.2824)	250250 (0.0000)	250125 (0.6863)	19247 (45256.1184)

El Cuadro 4 muestra los valores promedio de los parámetros estimados para todos los algoritmos estudiados. Al observar sus desviaciones estándar asociadas, es evidente que el DE proporcionó las estimaciones más precisas de todos los parámetros del modelo SU-CROS. CMA-ES y los otros dos algoritmos bio-inspirados muestran una precisión mayor o igual que el método local LSE. Las mayores desviaciones estándar están asociadas a las estimaciones del método local lo que significa que tiene más problemas de convergencia. Esto es consistente con los resultados mostrados en el Cuadro 3 con respecto a los valores de la función objetivo. De acuerdo con el análisis de varianza (ANDEVA) entre los algoritmos (Cuadro 6) se encontraron diferencias estadísticamente significativas para la estimación de los parámetros RGRL, TBASE, AMX, SCP, TREF, MAINST, MAINRT, MAINSO, ASRQSO, FRTRL, CFLV, CFST y FRDR. Esto significa que los parámetros deben ser estimados con cuidado por los algoritmos de optimización global. De acuerdo con el análisis de varianza de un solo factor (ANDEVA) el rendimiento de los cinco algoritmos es significativamente diferente ( $\alpha = 0,05$ ) (Cuadro 6). Teniendo en cuenta la comparación múltiple de medias de Tukey-Kramer, DE tuvo el mejor desempeño en el proceso de estimación.

Un análisis estadístico realizado para evaluar la calidad de la estimación de parámetros para todos los algoritmos utilizando los valores de los parámetros mostrados en el Cuadro 4 es presentado en el Cuadro 5. En él se resume el análisis estadístico llevado a cabo teniendo en cuenta todas las variables de estado [ a) peso seco total, b) peso seco de tallos, c) peso seco de frutos, d) peso seco de hojas y e) índice de área foliar]. De acuerdo con las estadísticas más importantes (MSE, RMSE y MAE), en general, el algoritmo DE tuvo el mejor rendimiento. Los algoritmos CMA-ES y ABC tuvieron un comportamiento de buena calidad. Posteriormente, el método local LSE y el algoritmo PSO tuvieron el peor desempeño. Se observó una diferencia en el sesgo: peso seco total fue sub-estimado por el

**Cuadro 4:** Valores promedio de los parámetros estimados del modelo SUCROS (Números entre paréntesis son la desviación estandar).

Parámetro	LSE	DE	CMA-ES	PSO	ABC
RGRL	0.0331 (0.0032)	0.0353 ( $1.58 \times 10^{-15}$ )	0.0330 (0.0016)	0.0327 (0.0048)	0.0283 (0.0026)
TBASE	11.6424 (0.6764)	11.8493 ( $1.20 \times 10^{-3}$ )	11.5606 (0.3510)	11.2778 (1.2168)	10.4799 (0.6867)
SLA	0.0176 ( $3.72 \times 10^{-5}$ )	0.0176 ( $9.53 \times 10^{-11}$ )	0.0176 ( $3.08 \times 10^{-5}$ )	0.0176 (0.0000)	0.0176 (0.0002)
AMX	0.0009 ( $2.51 \times 10^{-5}$ )	0.00088 ( $1.28 \times 10^{-10}$ )	0.00088 ( $5.93 \times 10^{-6}$ )	0.0008 ( $2.36 \times 10^{-5}$ )	0.0009 ( $5.47 \times 10^{-5}$ )
EFF	0.00001 ( $5.34 \times 10^{-15}$ )	0.00001 ( $2.39 \times 10^{-13}$ )	0.00001 ( $8.83 \times 10^{-9}$ )	$1.03 \times 10^{-5}$ ( $1.12 \times 10^{-6}$ )	0.00001 ( $4.11 \times 10^{-8}$ )
KDF	0.5267 (0.0246)	0.52 ( $1.65 \times 10^{-8}$ )	0.5218 ( $3.02 \times 10^{-3}$ )	0.5218 (0.0083)	0.5283 (0.0126)
SCP	0.2399 (0.0001)	0.24 ( $2.62 \times 10^{-16}$ )	0.2378 ( $2.56 \times 10^{-3}$ )	0.24 ( $1.42 \times 10^{-16}$ )	0.24 ( $7.8 \times 10^{-11}$ )
Q10	1.6022 (0.0065)	1.6 ( $2.24 \times 10^{-15}$ )	1.6 ( $6.48 \times 10^{-3}$ )	1.64 (0.1788)	1.6 ( $1.1 \times 10^{-14}$ )
TREF	20.0060 (0.0131)	20 (0.0000)	23.2383 (0.8027)	20 (0.0000)	20 ( $8.99 \times 10^{-11}$ )
MAINLV	0.0358 (0.0006)	0.036 ( $7.18 \times 10^{-17}$ )	0.0354 ( $5.27 \times 10^{-4}$ )	0.0354 (0.0026)	0.0358 (0.0003)
MAINST	0.0179 ( $2.6 \times 10^{-5}$ )	0.018 ( $3.55 \times 10^{-18}$ )	0.0179 ( $1.31 \times 10^{-4}$ )	0.018 ( $3.56 \times 10^{-18}$ )	0.018 ( $2.89 \times 10^{-13}$ )
MAINRT	0.0179 ( $3.62 \times 10^{-6}$ )	0.018 ( $6.61 \times 10^{-17}$ )	0.0174 ( $4.79 \times 10^{-4}$ )	0.0168 (0.0024)	0.0178 (0.0003)
MAINSO	0.0119 ( $5.54 \times 10^{-6}$ )	0.012 ( $6.51 \times 10^{-11}$ )	0.0174 ( $1.57 \times 10^{-4}$ )	0.012 ( $3.56 \times 10^{-18}$ )	0.012 ( $4.79 \times 10^{-18}$ )
ASRQLV	1.1826 (0.0501)	1.1704 ( $2.27 \times 10^{-16}$ )	1.1785 ( $6.46 \times 10^{-3}$ )	1.1996 (0.1308)	1.2091 (0.0632)
ASRQST	1.2104 (0.0002)	1.2104 ( $5.83 \times 10^{-9}$ )	1.2136 (0.0051)	1.2406 (0.1353)	1.2118 (0.0057)
ASRQRT	1.1628 (0.0255)	1.1552 ( $6.35 \times 10^{-8}$ )	1.1652 (0.0119)	1.1552 ( $2.28 \times 10^{-16}$ )	1.1643 (0.0408)
ASRQSO	1.7870 (0.0031)	1.788 ( $5.15 \times 10^{-9}$ )	1.7863 ( $1.67 \times 10^{-3}$ )	1.788 ( $4.56 \times 10^{-16}$ )	1.7878 (0.0005)
LAICR	4.1044 (0.4843)	3.8954 (0.5950)	3.8645 (0.4257)	4.0009 (0.7586)	3.9022 (0.4859)
FRTRL	0.16 (0.0004)	0.16 ( $2.33 \times 10^{-8}$ )	0.1621 ( $2.48 \times 10^{-3}$ )	0.16 ( $2.85 \times 10^{-17}$ )	0.16 ( $2.85 \times 10^{-17}$ )
CFLV	0.4750 (0.0531)	0.4688 (0.0663)	0.4488 (0.0392)	0.4931 (0.0835)	0.4283 (0.0479)
CFST	0.3954 (0.0010)	0.3952 ( $2.19 \times 10^{-9}$ )	0.4001 (0.0070)	0.3952 (0.0000)	0.3952 ( $4.08 \times 10^{-10}$ )
CFRT	0.4499 (0.0582)	0.4601 (0.0643)	0.4783 (0.0491)	0.4341 (0.0859)	0.4525 (0.0540)
CFSO	0.4468 (0.0567)	0.4694 (0.0692)	0.4723 (0.0561)	0.4623 (0.0942)	0.4702 (0.0567)
FRDR	0.8098 (0.0357)	1.1999 ( $3.24 \times 10^{-8}$ )	1.1943 ( $6.03 \times 10^{-3}$ )	1.18 (0.0894)	1.2 ( $8.21 \times 10^{-10}$ )
CONVL	0.7614 (0.0163)	0.7576 ( $6.26 \times 10^{-8}$ )	0.7789 (0.0337)	0.7765 (0.0847)	0.7576 ( $2.01 \times 10^{-13}$ )

DE y CMA-ES y sobre-estimado por PSO, ABC, y LSE (Figura 2). El peso seco de tallos fue sobre-estimado por todos los algoritmos (Figura 3). El peso seco de frutos fue sub-estimado en general (Figura 4). El peso seco de hojas fue sub-estimado por el algoritmo CMA-ES y sobre-estimado por el resto de los algoritmos (Figura 5). Por último, el índice de área foliar (IAF) se sobre-estimó (Figura 6).

Todos los algoritmos obtuvieron de manera similar la misma eficiencia de modelado (EF) y coeficiente de correlación ( $r$ ). Sin embargo, este estudio muestra que con el fin de evaluar la estimación de parámetros generada por cualquier algoritmo, se requiere una evaluación cuantitativa mediante el uso de varias medidas estadísticas. Este resultado está de acuerdo con las recomendaciones de Wallach *et al.* (2006) relativa a la evaluación de los modelos de cultivos.

Cuadro 5: Resultados estadísticos después de la calibración usando AE's y BI's.

Algorithm	MSE	RMSE	MAE	Bias	EF	r
<b>a)</b> Peso seco total						
DE	166.5839	12.9067	9.9436	2.6312	0.9911	0.9957
CMA-ES	170.0312	13.0396	10.4109	2.5954	0.9910	0.9957
PSO	403.9712	20.0990	13.5706	-9.0089	0.9785	0.9956
ABC	169.8817	13.0339	10.9542	-0.4404	0.9910	0.9959
LSE	179.5415	13.3993	11.2489	-0.8739	0.9905	0.9958
<b>b)</b> Peso seco de tallos						
DE	167.2770	12.9320	10.3838	10.3838	0.9503	0.9978
CMA-ES	160.2816	12.6602	10.2865	10.2865	0.9523	0.9977
PSO	57.3549	7.5733	6.1427	6.1427	0.9829	0.9977
ABC	124.2359	11.1461	9.1152	9.1152	0.9630	0.9978
LSE	118.8889	10.9036	8.9616	8.9616	0.9646	0.9978
<b>c)</b> Peso seco de frutos						
DE	360.0775	18.9757	11.5079	-7.6266	0.8823	0.9728
CMA-ES	418.5855	20.4594	11.9678	-8.1952	0.8631	0.9706
PSO	664.3558	25.7751	13.7441	-12.7859	0.7827	0.9716
ABC	444.3872	21.0805	12.1363	-9.1895	0.8547	0.9723
LSE	482.8671	21.9742	12.4283	-9.6200	0.8421	0.9714
<b>d)</b> Peso seco de hojas						
DE	118.9583	10.9068	7.7999	-0.1276	0.8087	0.8994
CMA-ES	118.0583	10.8655	7.6873	0.5024	0.8101	0.9003
PSO	129.7128	11.3891	8.3030	-2.3674	0.7913	0.8993
ABC	118.6198	10.8913	7.8804	-0.3678	0.8092	0.9000
LSE	117.9933	10.8625	7.8473	-0.2172	0.8102	0.9003
<b>e)</b> Índice de área foliar						
DE	0.0150	0.1225	0.0959	-0.0307	0.9194	0.9617
CMA-ES	0.0133	0.1155	0.0827	-0.0093	0.9283	0.9638
PSO	0.0219	0.1480	0.1193	-0.0790	0.8822	0.9627
ABC	0.0143	0.1194	0.0927	-0.0282	0.9233	0.9634
LSE	0.0084	0.0918	0.0695	-0.0267	0.9547	0.9791

‡MSE: Error Cuadrático Medio; RMSE: Raíz del Error Cuadrático Medio; MAE: Error Medio Promedio; r: Coeficiente de correlación.

Cuadro 6: ANDEVA de los cinco algoritmos evaluados sobre el modelo SUCROS (\*Diferencias significativas  $\alpha = 0,05$  ).

Parámetro	F	$p > F$
RGRL	15.1*	$1.368 \times 10^{-9}$
TBASE	11.26*	$1.61 \times 10^{-7}$
SLA	1.42	0.2325
AMX	41.88*	$3.3663 \times 10^{-20}$
EFF	0.98	0.4223
KDF	1.51	0.2046
SCP	14.11*	$4.49 \times 10^{-9}$
Q10	0.93	0.4524
TREF	325.13*	$1.67 \times 10^{-54}$
MAINLV	0.9	0.4661
MAINST	9.75*	$1.1963 \times 10^{-6}$
MAINRT	4.17*	0.0037
MAINSO	3.8*	0.0066
ASRQLV	1.07	0.3769
ASRQST	0.93	0.4506
ASRQRT	1	0.4095
ASRQSO	4.27*	0.0032
LAICR	0.56	0.6937
FRTRL	13.83*	$6.326 \times 10^{-9}$
CFLV	3.45*	0.011
CFST	9.28*	$2.24 \times 10^{-6}$
CFRT	1.28	0.2829
CFSO	0.47	0.7592
FRDR	316.87*	$5.206 \times 10^{-54}$
CONVL	1.28	0.2815
f(p)	82.99*	$3.77 \times 10^{-30}$
NFE	565.12*	$2.7316 \times 10^{-65}$

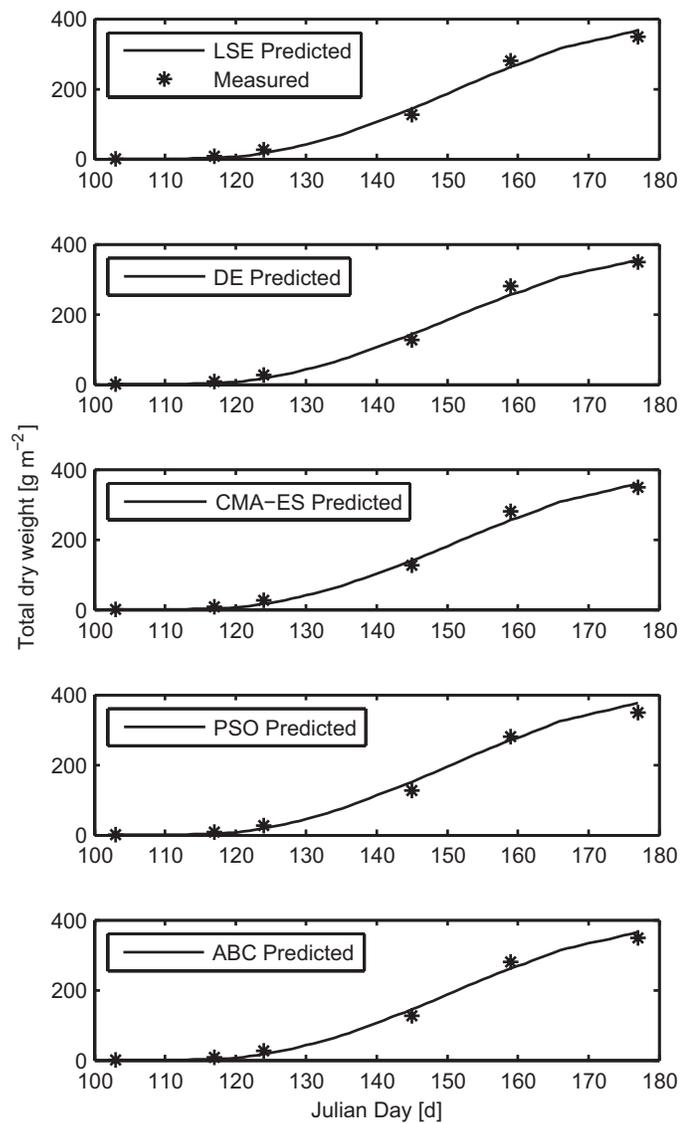


Figura 2: Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca total del tomate de cáscara.

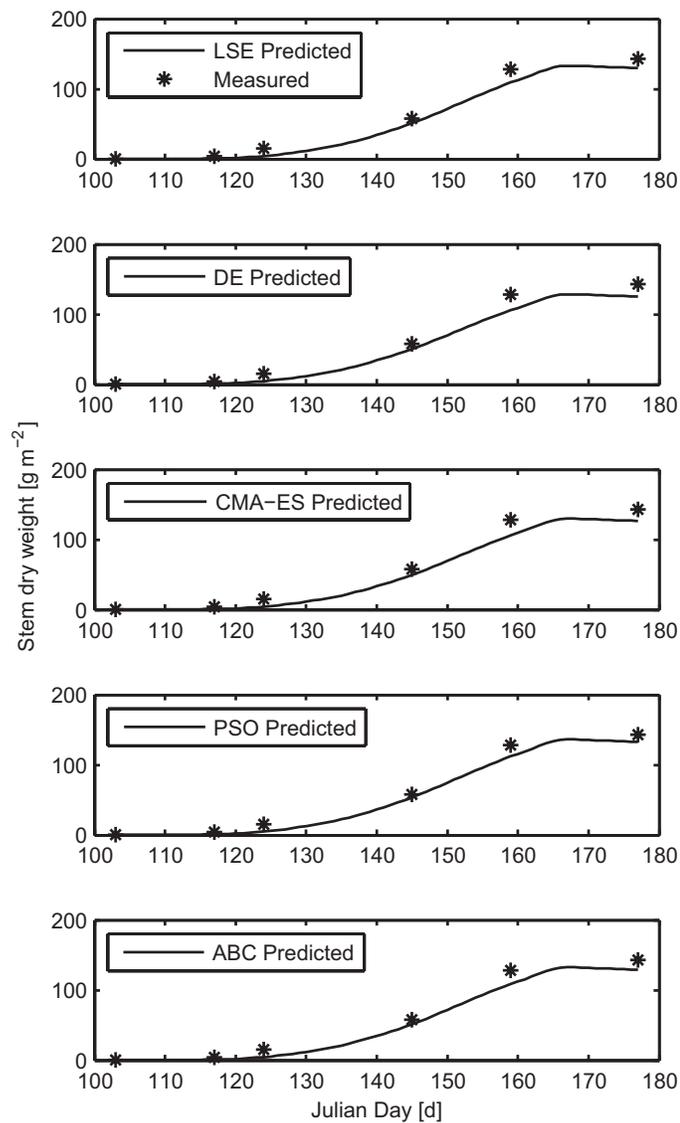


Figura 3: Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca de tallos del tomate de cáscara.

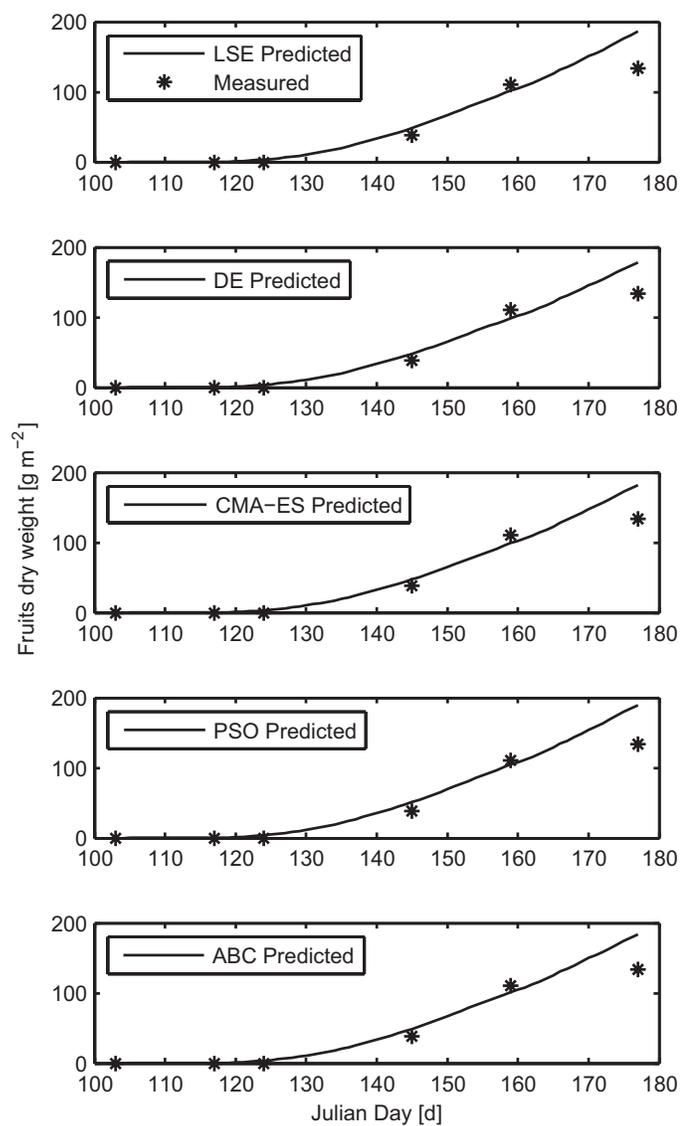


Figura 4: Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca de frutos del tomate de cáscara.

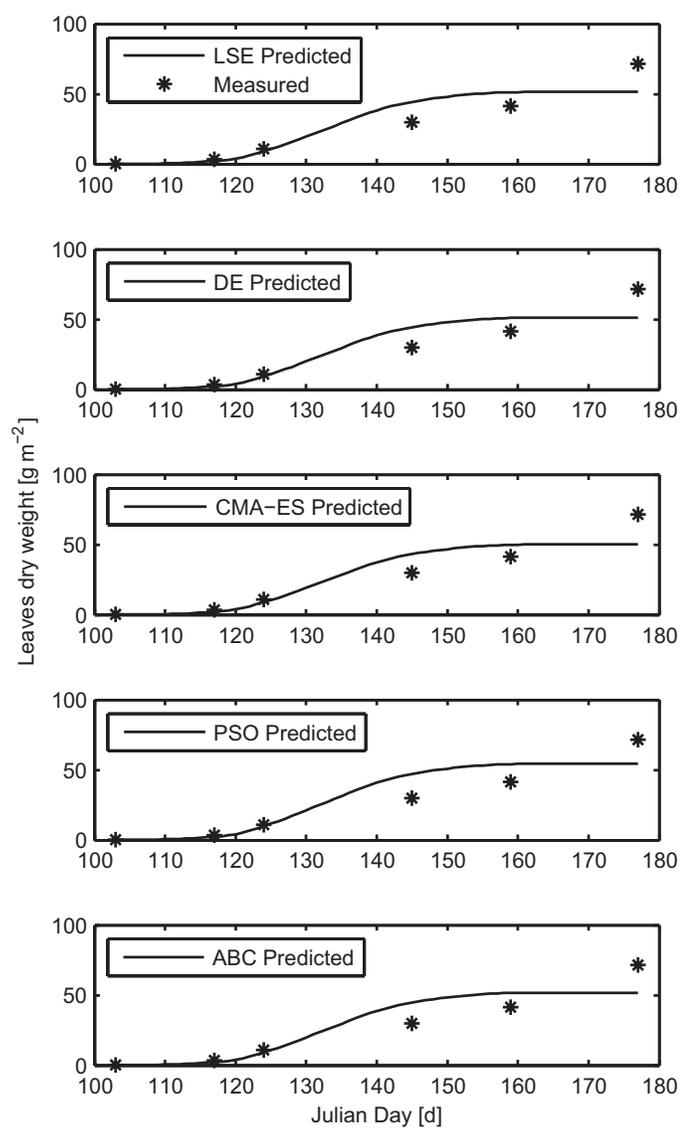


Figura 5: Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para la biomasa seca de hojas del tomate de cáscara.

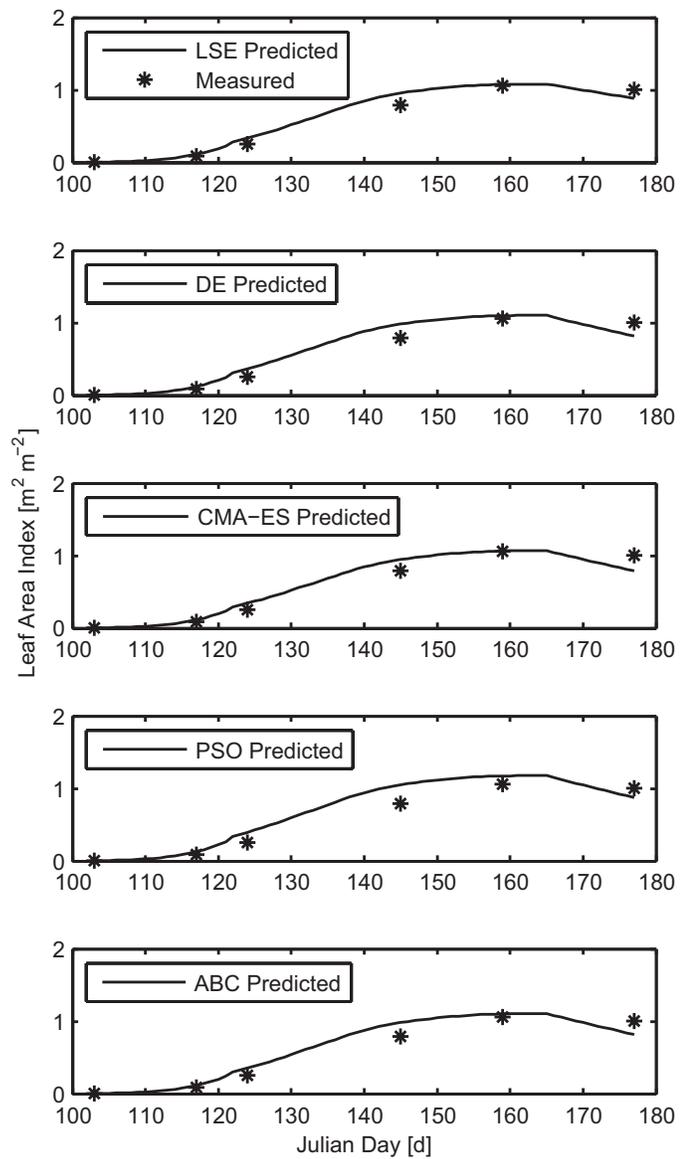


Figura 6: Predicción del modelo SUCROS calibrado con los algoritmos LSE, DE, CMA-ES, PSO y ABC para el índice de área foliar del cultivo tomate de cáscara.

## 4.4. Conclusión

El problema de estimación de parámetros de un modelo dinámico de crecimiento de cultivos puede enfrentar algunos inconvenientes cuando es abordado por algoritmos de optimización local. A partir de los resultados obtenidos, el algoritmo evolución diferencial fue el método más eficaz y relativamente eficiente para resolver el problema de calibración del modelo SUCROS. Sin embargo, es necesario investigar más a fondo si los resultados obtenidos en este estudio pueden ser corroborados en el caso de la estimación de parámetros en modelos más complejos con decenas o cientos de parámetros a determinar. Al parecer, una estimación de parámetros más precisa y fiable se puede obtener utilizando métodos de optimización global, aunque se requiere un poco de tiempo para encontrar valores adecuados de los parámetros de los algoritmos evolutivos y bio-inspirados. Por esto, una dirección para trabajos futuros es el uso de algoritmos de evolución diferencial adaptables con el fin de reducir la dificultad de ajustar sus parámetros de diseño.

---

# CAPÍTULO 5

---

## DISCUSIÓN GENERAL

Las Técnicas de Inteligencia Artificial o Computación suave son herramientas poderosas para resolver problemas complejos y no-lineales los cuales son difíciles o casi imposibles de resolver por métodos convencionales tales como: mínimos cuadrados, máxima verosimilitud, programación cuadrática secuencial, etc., ya que pueden converger a soluciones sub-óptimas o mínimos locales cuando se aborda un problema de optimización multimodal. Para resolver este tipo de problemas, es necesario el uso de algoritmos conocidos como algoritmos de búsqueda global. La Lógica difusa y Redes neuronales están siendo aplicadas en la solución de problemas para la ingeniería agrícola principalmente en modelado y control del clima en invernaderos (Capítulo 2). Las variables climáticas más importantes consideradas en la mayoría de los trabajos de investigación son la temperatura, humedad relativa y concentración de  $CO_2$  en el interior del biosistema. Mantener estas variables dentro de rangos óptimos implica el uso de estrategias de control que garanticen un crecimiento y desarrollo del cultivo dentro del invernadero. Es por esto que algunos controladores difusos asociados a controladores clásicos como PI (Proporcional-Integral) o PID (Proporcional-Integral-Derivativo) están siendo aplicados para controlar las variables climáticas en invernaderos. Por otro lado, las Redes neuronales han tenido éxito en aplicaciones de modelado y control del clima en invernaderos debido a que tienen características de generalización y habilidad para aprender de la experiencia sin requerir un modelo matemático a priori de las características del sistema y habilidad para procesar datos degradados o incompletos. Sin embargo, las ANN requieren de un gran número de muestras y mucho tiempo de entrenamiento para un aprendizaje adecuado. La relevancia de las redes neuronales en control radica en que se adaptan a los cambios del sistema y de esta manera tienen un comportamiento satisfactorio en procesos variables no lineales (Huang *et al.*, 2010). Las Redes neuronales han sido aplicadas en procesos de identificación y estimación de crecimiento de cultivos (Córdova y Yu, 2009; Fitz-Rodríguez y Giacomelli,

2009). Otro grupo importante de algoritmos son los algoritmos evolutivos, estos han sido moderadamente aplicados al modelado y control del clima en invernaderos (Capítulo 2). Un Algoritmo Evolutivo es una técnica de búsqueda de soluciones inspirada en la evolución de los seres vivos, en él se define una población de soluciones potenciales para un problema. A través de un proceso llamado mutación, cruza y selección se generan soluciones nuevas que se aproximan al valor deseado (posible solución óptima) (Beyer, 2001). Esta característica importante de los algoritmos evolutivos ha permitido su aplicación en procesos de sintonía de parámetros en Redes neuronales, procesos de identificación de sistemas, calibración de modelos, control óptimo, etc. Recientemente, los algoritmos bio-inspirados están siendo aplicados a la ingeniería agrícola principalmente al problema de identificación y estimación de parámetros para modelos de crecimiento de cultivos (Trejo-Zuñiga *et al.*, 2013). Los Algoritmos Bio-Inspirados simulan el comportamiento de sistemas naturales (banco de peces, parvadas de aves, etc.). En particular, la disciplina se centra en los comportamientos colectivos que resultan de las interacciones locales de los individuos entre sí y con su entorno (Chen *et al.*, 2008). Como ejemplo se puede mencionar al PSO (Optimización por enjambre de partículas), el cual ha sido aplicado para diseñar un controlador predictivo basado en modelo para el clima de un invernadero (Cohelo *et al.*, 2002, 2005). Otros importantes algoritmos bio-inspirados son: ABC (Colonia artificial de abejas) y ACO (Optimización por colonia de hormigas). Las aplicaciones de los algoritmos bio-inspirados al modelado y control del ambiente de invernaderos es muy incipiente y por lo tanto esta familia de algoritmos tiene un gran potencial de aplicación futura.

Los procesos esenciales en un cultivo, necesarios para describir el índice de cosecha de frutos, son la fotosíntesis, la distribución de asimilados, el crecimiento y la respiración de mantenimiento. La fotosíntesis describe la producción de asimilados y es una función de la radiación, temperatura de la hoja, concentración de  $CO_2$ , humedad y el índice de área foliar. La distribución de asimilados describe la distribución de los asimilados entre las diferentes partes de la planta, como raíces, tallos, hojas y frutos. La respiración de crecimiento es proporcional al crecimiento, que a su vez depende de la cantidad de asimilados disponibles y la temperatura. La respiración de mantenimiento es la combustión de asimilados necesarios para mantener la planta. Es una función del peso seco de raíz, tallos, hojas, frutos y la temperatura. La radiación, temperatura, concentración de  $CO_2$  y la humedad pueden ser vistas como entradas para la planta. Por consiguiente, las cantidades restantes deben ser pronosticadas por el modelo, es decir, el peso seco de las raíces, tallos, hojas, frutos, índice

de área foliar, promedio de temperaturas y radiación media.

Para lograr una buena predicción para las variables de salida, es necesario realizar un proceso de estimación de los parámetros que constituyen el modelo. La estimación de parámetros implica la realización de experimentos para poder disponer de mediciones del sistema real. En particular se requiere primero del diseño de un experimento. Su ejecución y la obtención de mediciones de las variables de salida del modelo. Posteriormente el planteamiento y solución de un problema de optimización.

El problema de estimación de parámetros (Ioslovich *et al.*, 2002; van Straten, 2012) consiste en encontrar un conjunto de valores para los parámetros ( $p$ ) del modelo dinámico, de tal forma que minimice un criterio ( $V_p$ ), normalmente el cuadrado medio del error entre mediciones obtenidas del sistema real ( $z_k$ ) y las predicciones del modelo ( $y(k | p)$ ) usando un algoritmo de optimización o método de búsqueda (Capítulo 1).

Los algoritmos locales de búsqueda empleados son: mínimos cuadrados, máxima verosimilitud, programación cuadrática secuencial, etc. Estos funcionan en forma adecuada cuando los modelos tienen pocos parámetros (menos de 10), pero generan estimaciones imprecisas cuando el modelo tiene muchos parámetros. Además, estos algoritmos pueden converger a soluciones sub-óptimas o mínimos locales. Por estas razones, recientemente se han empezado a utilizar métodos globales de búsqueda. Algunos algoritmos globales de búsqueda son: algoritmos evolutivos (algoritmos genéticos, estrategias evolutivas, programación evolutiva, evolución diferencial, etc.) y algoritmos bio-inspirados (Optimización con enjambres de partículas, optimización con colonias de hormigas, optimización con colonias artificiales de abejas, búsqueda Cuco, etc.). Estos algoritmos presentan características importantes dentro de su estructura lo que permite realizar un proceso de búsqueda multipunto, explorar y explotar el área de búsqueda y converger posiblemente a una solución óptima del problema (Capítulo 3 y 4). Existe una gran variedad de algoritmos bio-inspirados además de los antes mencionados (Fister *et al.*, 2013) los cuales pueden ser usados para resolver problemas en ingeniería agrícola (Figura 1). La elección del algoritmo a aplicar, está en función del problema a resolver, de la facilidad para programarlo y del tiempo de cómputo que se necesita para ejecutarlo.

Para realizar la estimación de los 25 parámetros que constituye el modelo de crecimiento potencial de cultivos SUCROS (Capítulos 3 y 4) fue necesario la implementación de algoritmos evolutivos (Evolución diferencial y estrategias evolutivas con matriz de covarianza adaptable) y bio-inspirados (Optimización por enjambre de partículas, PSO; Colonia

Swarm intelligence based algorithms		Bio-inspired (not SI-based) algorithms	
Algorithm	Author	Algorithm	Author
Accelerated PSO	Yang et al.	Atmosphere clouds model	Yan and Hao
Ant colony optimization	Dorigo	Biogeography-based optimization	Simon
Artificial bee colony	Karaboga and Basturk	Brain Storm Optimization	Shi
Bacterial foraging	Passino	Differential evolution	Storn and Price
Bacterial-GA Foraging	Chen et al.	Dolphin echolocation	Kaveh and Farhoudi
Bat algorithm	Yang	Japanese tree frogs calling	Hernández and Blum
Bee colony optimization	Teodorović and Dell'Orco	Eco-inspired evolutionary algorithm	Parpinelli and Lopes
Bee system	Lucic and Teodorovic	Egyptian Vulture	Sur et al.
BeeHive	Wedde et al.	Fish-school Search	Lima et al.
Wolf search	Tang et al.	Flower pollination algorithm	Yang
Bees algorithms	Pham et al.	Gene expression	Ferreira
Bees swarm optimization	Drias et al.	Great salmon run	Mozaffari
Bumblebees	Comellas and Martinez	Group search optimizer	He et al.
Cat swarm	Chu et al.	Human-Inspired Algorithm	Zhang et al.
Consultant-guided search	Iordache	Invasive weed optimization	Mehrabian and Lucas
Cuckoo search	Yang and Deb	Marriage in honey bees	Abbass
Eagle strategy	Yang and Deb	OptBees	Maia et al.
Fast bacterial swarming algorithm	Chu et al.	Paddy Field Algorithm	Premaratne et al.
Firefly algorithm	Yang	Roach infestation algorithm	Havens
Fish swarm/school	Li et al.	Queen-bee evolution	Jung
Good lattice swarm optimization	Su et al.	Shuffled frog leaping algorithm	Eusuff and Lansey
Glowworm swarm optimization	Krishnanand and Ghose	Termite colony optimization	Hedayatzadeh et al.
Hierarchical swarm model	Chen et al.	<b>Physics and Chemistry based algorithms</b>	
Krill Herd	Gandomi and Alavi	Big bang-big Crunch	Zandi et al.
Monkey search	Mucherino and Seref	Black hole	Hatamlou
Particle swarm algorithm	Kennedy and Eberhart	Central force optimization	Formato
Virtual ant algorithm	Yang	Charged system search	Kaveh and Talatahari
Virtual bees	Yang	Electro-magnetism optimization	Cuevas et al.
Weightless Swarm Algorithm	Ting et al.	Galaxy-based search algorithm	Shah-Hosseini
<b>Other algorithms</b>		Gravitational search	Rashedi et al.
Anarchic society optimization	Shayeghi and Dadashpour	Harmony search	Geem et al.
Artificial cooperative search	Civicioglu	Intelligent water drop	Shah-Hosseini
Backtracking optimization search	Civicioglu	River formation dynamics	Rabanal et al.
Differential search algorithm	Civicioglu	Self-propelled particles	Vicsek
Grammatical evolution	Ryan et al.	Simulated annealing	Kirkpatrick et al.
Imperialist competitive algorithm	Atashpaz-Gargari and Lucas	Stochastic diffusion search	Bishop
League championship algorithm	Kashan	Spiral optimization	Tamura and Yasuda
Social emotional optimization	Xu et al.	Water cycle algorithm	Eskandar et al.

Figura 1: Clasificación de los algoritmos bio-inspirados (adaptado de Fister *et al.*, 2013)

artificial de abejas, ABC; Búsqueda cuco clásico, CS; y búsqueda cuco modificado, MCS) con la finalidad de analizar qué algoritmo genera valores de los parámetros que den las mejores predicciones del modelo para las variables de salida consideradas [Peso seco total ( $gm^{-2}$ ), Peso seco de tallos ( $gm^{-2}$ ), Peso seco de frutos ( $gm^{-2}$ ), Peso seco de hojas ( $gm^{-2}$ ) e Índice de área foliar ( $m^2m^{-2}$ )], en función de las variables de entrada del modelo [Radiación global ( $Jm^{-2}d^{-1}$ ), Temperatura mínima diaria ( $^{\circ}C$ ) y Temperatura máxima diaria ( $^{\circ}C$ )].

Los valores promedio obtenidos en este proceso después de realizadas 20 optimizaciones, fueron plasmados en los Cuadros 2 (Capítulo 3) y 3 (Capítulo 4). De acuerdo con los valores de la función objetivo  $f(p)$ , el algoritmo de evolución diferencial (DE) fue el método más efectivo ya que no solo alcanzó el valor más cercano a cero, sino que también mostró la desviación estándar más pequeña. Buenos resultados fueron obtenidos por los algoritmos de búsqueda cuco clásico y modificado, seguido por el algoritmo colonia artificial de abejas. En menor grado de aproximación quedaron los algoritmos CMA-ES y PSO.

Un análisis estadístico y de varianza (ANDEVA) fue llevado a cabo para mostrar la existencia de posibles diferencias estadísticas entre los valores de los parámetros estimados [Cuadro 3 (Capítulo 3) y Cuadro 6 (Capítulo 4)]. Los resultados arrojados por el análisis mostraron diferencias estadísticas significativas en la mayoría de los parámetros estimados a excepción de EFF, KDF, ASRQLV, LAICR, CFLV, CFRT y CFSO. Esto significa que son más confiables y precisas las estimaciones obtenidas mediante el algoritmo DE.

En las Figuras 4 (Capítulo 3), 2, 3, 4, 5 y 6 (Capítulo 4) se puede observar cómo los algoritmos estudiados sobre-estimaron o sub-estimaron las diferentes variables de salida del modelo no lográndose apreciar con claridad una diferencia entre las variables estimadas. Se observa que en la estimación del peso seco total y peso seco de tallos hubo una sub-estimación y en peso seco de frutos, peso seco de hojas e índice de área foliar hubo sobre-estimación del modelo. En general puede apreciarse un mejor ajuste en el caso de las estimaciones de parámetros obtenidas mediante el algoritmo DE.

Para poder realizar un proceso comparativo más eficiente entre los algoritmos estudiados se hizo uso de herramientas estadísticas tales como: MSE (error cuadrático medio), RMSE (Raíz del error cuadrático medio), MAE (Error medio absoluto), sesgo, eficiencia de modelado (EF) y coeficiente de correlación ( $r$ ) [Cuadro 4 (Capítulo 3) y Cuadro 5 (Capítulo 4)]. En general, el algoritmo de evolución diferencial generó valores estadísticos más cercanos a cero en función de los errores, seguido por los algoritmos búsqueda cuco clásico y modificado. El algoritmo ABC tuvo un mejor desempeño que CMA-ES y PSO en el proceso de estimación de parámetros del modelo de crecimiento potencial de cultivos SUCROS.

---

# CAPÍTULO 6

---

## CONCLUSIONES

- I. Tanto los sistemas difusos como las redes neuronales se han usado desde hace más de una década para modelar y controlar principalmente el clima de los invernaderos. Se han obtenido mejores resultados con los métodos inteligentes que con los enfoques tradicionales. También se han logrado más avances en la modelación del ambiente invernadero y menos en su control. La mayoría de las investigaciones de modelado se han centrado en el clima y poco en procesos relacionados con el cultivo tales como el crecimiento, desarrollo, fotosíntesis, respiración o transpiración, lo cual es fundamental si se desea optimizar el biosistema invernadero. En el caso de ANN se han usado tanto para modelar como para controlar las redes estáticas sobre las dinámicas, sin embargo para controlar en tiempo real, se requiere de redes dinámicas más complejas.
- II. Los Algoritmos Bio-inspirados han empezado a ser estudiados y aplicados en problemas de Ingeniería Agrícola y en el sistema invernadero. En el caso del Cómputo Evolutivo, algunos algoritmos como los GA's y los DE son muy comunes en aplicaciones de modelado pero su aplicación al control es muy incipiente y escasa. Los Algoritmos híbridos han sido poco usados para el control del clima. Sin embargo, la aplicación al modelado está incrementando considerablemente. Por lo tanto, se considera a los algoritmos Bio-inspirados, evolutivos e híbridos un área bastante prometedora para optimizar y controlar el ambiente de los invernaderos.
- III. El problema de estimación de parámetros de un modelo dinámico de crecimiento de cultivos puede enfrentar algunos inconvenientes cuando es abordado por algoritmos de optimización local. A partir de los resultados obtenidos, el algoritmo evolución diferencial fue el método más eficaz y relativamente eficiente para resolver el problema de calibración del modelo SUCROS. Sin embargo, es necesario investigar más a

fondo si los resultados obtenidos en este estudio pueden ser corroborados en el caso de la estimación de parámetros en modelos más complejos con decenas o cientos de parámetros a determinar. Al parecer, una estimación de parámetros más precisa y fiable se puede obtener utilizando métodos de optimización global, aunque se requiere un poco de tiempo para encontrar valores adecuados de los parámetros de los algoritmos evolutivos y bio-inspirados. Por esto, una dirección para trabajos futuros es el uso de algoritmos de evolución diferencial adaptables con el fin de reducir la dificultad de ajustar sus parámetros de diseño.

# Referencias

- [1] Arahal, M. R., F. Rodríguez, A. Ramírez-Arias, and M. Berenguel. 2005. Discrete-time nonlinear FIR models with integrated variables for greenhouse indoor temperature simulation. Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville, Spain, December 12-15.
- [2] Auger, A., and N. Hansen. 2005. A Restart CMA Evolution Strategy With Increasing Population Size, Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1769-1776. DOI:10.1109/CEC.2005.1554902
- [3] Banharnsakun, A., T. Achalakul, and B. Sirinaovakul. 2011. The best-so-far selection in Artificial Bee Colony algorithm. Applied Soft Computing 11: 2888-2901.
- [4] Baritomba, B., and E. M. T. Hendrix. 2005. On the investigation of Stochastic global optimization algorithms, Journal of Global Optimization 31:567-578. DOI: 10.1007/s10898-004-9966-0
- [5] Beasley, D., D. R. Bull, and R. R. Martin. 1993. An Overview of Genetic Algorithms: Part 1, Fundamentals. University Computing, 15(2): 58-69.
- [6] Bernardo, J. I., M. Silverio, S. M. Amancio, R. T. Atarassi, and A. R. Hirakawa. 2006. Analytical modeling in greenhouse for simulation and optimized multivariable control. Computers in Agriculture and Natural Resources, 4th World Congress Conference, Proceedings of the 24-26 July 2006 (Orlando, Florida USA).
- [7] Beyer, H. G. 2001. The Theory of Evolution Strategies. Natural Computing Series. Springer, Berlin 2001.
- [8] Beyer, H. G., and H. P. Schwefel. 2002. Evolution Strategies: A Comprehensive Introduction. In Natural Computing, 1(1): 3-52.

- [9] Blasco, X., M. Martínez, J. M. Herrero, C. Ramos, and J. Sanchis. 2007. Model-based predictive control of greenhouse climate for reducing energy and water consumption. *Computers and Electronics in Agriculture* 55: 49-70. DOI: 10.1016/j.compag.2006.12.001
- [10] Bouman, B.A.M., H. van Keulen, H. H. van Laar, and R. Rabbinge. 1996. The 'School of the Wit' crop growth simulation models: a pedigree and historical overview. *Agricultural Systems* 52:171-198.
- [11] Braga, W. 2006. Optimization of radiative heat transfer inside Greenhouses. Proceedings of the 11th Brazilian Congress of Thermal Sciences and Engineering, ENCIT 2006 Braz. Soc. of Mechanical Sciences and Engineering, ABCM, Curitiba, Brazil,- Dec. 5-8.
- [12] Bussab, M. A., J. I. Bernardo, and A. R. Hirakawa. 2007. Greenhouse Modeling Using Neural Networks. Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19.
- [13] Carrillo-Reveles, D. A., J. L. Vázquez-Minjares, R. Villela-Varela, A. Beltrán-Telles, y C. Reyes-Rivas. 2008. Control del clima de un invernadero. XII JORNADAS DE INVESTIGACIÓN, Revista Investigación Científica, Vol. 4, No. 2, Nueva época, mayo-agosto.
- [14] Castañeda-Miranda R., E. Jr. Ventura-Ramos, R. R. Peniche-Vera, and G. Herrera-Ruíz. 2007. Analisis and simulation of a Greenhouse physical model under weather conditions of the central region of Mexico. *Agrociencia* 41: 317-335.
- [15] Castañeda-Miranda R., E. Ventura-Ramos Jr., R. del R. Peniche-Vera, and G. Herrera-Ruíz. 2006. Fuzzy Greenhouse Climate Control System based on a Field Programmable Gate Array. *Biosystems Engineering* 94 (2): 165-177.
- [16] Center, B., and B. P. Verma. 1997. A Fuzzy photosynthesis model for Tomato. *American Society of Agricultural Engineers*, VOL. 40(3):815-821.
- [17] Chakraborty, U. K. 2008. *Advances in Differential Evolution. Studies in Computational Intelligence* 143, Springer-Verlag Berlin Heidelberg.

- [18] Chalabi, Z. S., and W. Zhou. 1996. Optimal control methods for agricultural systems, *Acta Horticulturae* 406: 221-228.
- [19] Chao-gang, Y., Y. Yi-bin, W. Jian-ping, N. Jamal, and Y. Jia. 2005. Determining heating pipe temperature in greenhouse using proportional integral plus feedforward control and radial basic function neural-networks. *Journal of Zhejiang University*, 6A(4): 265-269.
- [20] Chen, S. H., A. J. Jakeman, and J. P. Norton. 2008. Artificial Intelligence techniques: An introduction to their use for modelling environmental systems. *Mathematics and Computers in Simulation* 78: 379-400.
- [21] Cheng-Jian, L., and X. Yong-Ji. 2007. Design of Neuro-Fuzzy Systems Using a Hybrid Evolutionary Learning Algorithm. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING* 23: 463-477.
- [22] Chiong, R. (Ed.). 2009. *Nature-Inspired Algorithms for Optimisation*. Studies in Computational Intelligence, Volume 193.
- [23] Civicioglu, P., and E. Besdok. 2013. A conceptual comparison of the Cuckoo-Search, Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony algorithms. *Artif. Intell Rev*, 39: 315-346.
- [24] Clerc, M., and J. Kennedy. 2002. The particle swarm-explosion, stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58-73. DOI: 10.1109/4235.985692
- [25] Coelho, J. P., P. de Moura-Oliveira, and J. B. Cunha. 2002. Greenhouse air temperature control using the particle swarm optimisation algorithm. *IFAC*.
- [26] Coelho, J. P., P. de Moura-Oliveira, and J. B. Cunha. 2005. Greenhouse air temperature predictive control using the particle swarm optimisation algorithm. *Computers and Electronics in Agriculture* 49: 330-344.
- [27] Cook, R., and L. Calvin. 2006. Greenhouse tomato change the dynamics of the North American fresh tomato industry. *USDA, Economic Research Report No. 288 p.*

- [28] Cordova, J. J., and W. Yu. 2009. Stable Fourier Neural Networks with Application to Modeling Lettuce Growth. Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19.
- [29] Costa, E., P. A. M. Leal, and R. R. do Carmo Junior. 2004. Simulation model of air temperature and relative humidity in to plastic greenhouses. Eng. AgrÃc., Jaboticabal, v.24, n.1: p.57-67, jan/abr.
- [30] Cunha, J. B. 2003. Greenhouses climate models: An overview. EFITA Conference, 5-9. July, Debrecen, Hungary.
- [31] Cunha, J. B., and J. P. de Moura-Oliveira. 2003. Optimal management of greenhouse environments. EFITA Conference 5-9. July, Debrecen, Hungary.
- [32] Dai, C., M. Yao, Z. Xie, C. Chen, and J. Liu. 2009. Parameter optimization for growth model of greenhouse crop using genetic algorithms. Applied Soft Computing 9: 13-19. DOI: 10.1016/j.asoc.2008.02.002
- [33] Das, S., and P. N. Suganthan. 2011. Differential Evolution: A Survey of the State of the Art. IEEE Transactions on evolutionary computation, 15: 4-31.
- [34] Dorigo, M. 1992. Optimization, Learning and Natural Algorithms (in Italian), Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, IT.
- [35] Dorigo, M., and C. Blumb. 2005. Ant colony optimization theory: A survey. Theoretical Computer Science 344: 243-278.
- [36] Dorigo, M., M. Birattari, and T. StÃ¼tzle. 2006. Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. IEEE Computational Intelligence Magazine, 1(4): 28-39.
- [37] Dorigo, M., and T. StÃ¼tzle. 2004. Ant Colony Optimization. MIT Press, Cambridge, MA.
- [38] Dorigo, M., V. Maniezzo, and A. Colorni. 1996. Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics - Part B, 26(1): 29-41.

- [39] Dumitrascu, A. 2007. Applications development in greenhouse environment. CEAI, Vol. 9, No. 1: pp. 47-54.
- [40] Eberhart, R. C., and Y. Shi. 2001. Particle Swarm Optimization: Developments, Applications and Resources. *Evolutionary Computation. Proceedings of the 2001 Congress*, 1:81-86. DOI:10.1109/CEC.2001.934374
- [41] Eiben, A. E., and J. E. Smith. 2003. *Introduction to Evolutionary Computing*, Springer, Berlin, Germany, 299 p.
- [42] El Aoud, M. M., M. Franceschi, and M. Maher. 2007. Self-Tuning Method of Fuzzy System: An Application on Greenhouse Process. *World Academy of Science, Engineering and Technology* 31.
- [43] Engelbrecht, A., P. 2005. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, Chichester, UK.
- [44] Errahmani, A., M. Benyakhlef, and I. Boumhidi. 2009. Decentralized Fuzzy Control Applied to a Greenhouse. *ICGST-ACSE Journal*, ISSN 1687-4811, Volume 9, Issue II, October.
- [45] Errahmani, A., M. Benyakhlef, and I. Boumhidi. 2009. Greenhouse Model Identification based on Fuzzy Clustering Approach. *ICGST-ACSE Journal*, ISSN 1687-4811, Volume 9: Issue II.
- [46] Eslamian, S. S., J. Abedi-Koupai, M. J. Amiri, and S. A. Gohari 2009. Estimation of Daily Reference Evapotranspiration Using Support Vector Machines and Artificial Neural Networks in Greenhouse. *Research Journal of Environmental Sciences* 3(4): 439-447.
- [47] Ferentinos, K. P., and L. D. Albright. 2002. Predictive Neural network modeling of pH and electrical conductivity in deep-trough hydroponics. *American Society of Agricultural Engineers*, Vol. 45(6): 2007-2015.
- [48] Ferreira, P. M., E. A. Faria, and A. E. Ruano. 2000a. Application of Radial Basis Function Neural Networks to a Greenhouse inside air temperature model. *Proceedings of IFAC Agricontrol 2000, International Conference on Modelling and Control*

in Agriculture, Horticulture and Post-Harvest Processing, July 2000, Wageningen, the Netherlands, pp 172-177.

- [49] Ferreira, P. M., E. A. Faria, and A. E. Ruano. 2000b. Neural Network Models in Greenhouse Environmental Control. Proceedings of EANN'2000, the 6th International Conference on Engineering Applications of Neural Networks, July 2000, Kingston University, England, pp 100-107.
- [50] Ferreira, P. M., E. A. Faria, and A. E. Ruano. 2000c. Comparison of on-line learning algorithms for RBF models in Greenhouse environmental control problems. *Controlo'2000: 4th Portuguese Conference on Automatic Control*. p. 342-347.
- [51] Ferreira, P. M., and A. E. Ruano. 2001a. Greenhouse air temperature modeling with Radial Basis Function Neural Networks. Proceedings of the Workshop on Management, Identification and Control of Agriculture Buildings, January, Vila Real, Portugal.
- [52] Ferreira, P. M., and A. E. Ruano. 2001b. Predicting the Greenhouse inside air temperature with RBF Neural Networks. Proceedings of ICAA, the 2nd IFAC-CIGR Workshop on Intelligent Control for Agricultural Applications, August, Bali, Indonesia, pp 67-72.
- [53] Ferreira, P. M., and A. E. Ruano. 2002a. Choice of RBF model structure for predicting greenhouse inside air temperature. IFAC, 15th Triennial World Congress, Barcelona, Spain.
- [54] Ferreira, P. M., E. A. Faria, and A. E. Ruano. 2002b. Neural network models in greenhouse air temperature prediction. *Neurocomputing* 43: 51-75.
- [55] Ferreira, P. M., and A. E. Ruano. 2004. Predicting solar radiation with RBF Neural Networks. Proceedings of CONTROLO'2004, the 6th Portuguese Conference on Automatic Control, June, Faro, Portugal.
- [56] Ferreira, P. M., A. E. Ruano, and C. M. Fonseca. 2005. Evolutionary Multiobjective design of Radial Basis Function Networks for Greenhouse environmental control. Proc. of the 16th IFAC World.

- [57] Ferreira, P. M., and A. E. Ruano. 2008. Model based predictive control of greenhouse air temperature and relative humidity. Proceedings of CONTROLO'2008, the 8th Portuguese Conference on Automatic Control, July 2008, Vila Real, Portugal.
- [58] Fister Jr., I, X. Yang, I. Fister, J. Brest, and D. Fister. 2013. A Brief Review of Nature-Inspired Algorithms for Optimization. *Elektrotehniski Vestnik*, Vol 80(3): 1-7.
- [59] Fitz-Rodríguez, E., and G. A. Giacomelli. 2009. Yield Prediction and Growth Mode Characterization of Greenhouse with Neural Networks and Fuzzy Logic. *Transactions of the ASABE (American Society of Agricultural and Biological Engineers)*, Vol 52(6): 2115-2128.
- [60] Fogel, L. J. 1962. Autonomous automata. *Industrial Research*, Vol. 4: pp. 14-19.
- [61] Fogel, L. J. 1963. *Biotechnology: Concepts and Applications*, Prentice-Hall, Englewood, NJ.
- [62] Fogel, L. J., and D. B. Fogel. 1986. Artificial intelligence through evolutionary programming. Final Report, Contract PO-9-X56-1102C-1, U.S. Army Research Institute.
- [63] Fogel, D. 2006. *Evolutionary Computation. Toward a new philosophy of machine intelligence*. IEEE Press.
- [64] Fourati, F., and M. Chtourou. 2007. A greenhouse control with feed-forward and recurrent neural networks. *Simulation Modelling Practice and Theory* 15: 1016-1028.
- [65] Frausto, H. U., J. G. Pieters, and J. M. Deltour. 2003. Modelling Greenhouse Temperature by means of Auto Regressive Models. *Biosystems Engineering*, 84 (2): 147-157.
- [66] Gates, R. S., K. Chao, and N. Sigrimis. 1999. Fuzzy control simulation of plant and animal environments. *ASAE Annual International Meeting*, July 18-21, Sheraton Centre Hotel, Toronto, ON Canada.
- [67] Gates, R. S., K. Chao, and N. Sigrimis. 2001. Identifying design parameters for fuzzy control of staged ventilation control systems. *Computers and Electronics in Agriculture* 31: 61-74.

- [68] Gill, P. E., W. Murray, and M. H. Wright. 1981. *Practical Optimization*. Academic Press Inc. San Diego, USA, 401 p.
- [69] Goggos, V., and R.E. King. 2000. Qualitative-evolutionary design of greenhouse environment control agents. *Computers and Electronics in Agriculture* 26: 271-282.
- [70] Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA 412 pp.
- [71] Gray, A., and R. Kilgour. 2007. Frequently Asked Questions: Hybrid Systems, <http://www.cs.missouri.edu/research/rsun/hybrid-FAQ.html>.
- [72] Goudriaan, J., and H. H. van Laar. 2003. *Modelling potential crop growth processes. Textbook with exercises*, Kluwer Academic Publisher, The Netherlands.
- [73] Guzmán-Cruz R., R. Castañeda-Miranda, J. J. García-Escalante, I. L. López-Cruz, A. Lara-Herrera, and J. I. de la Rosa. 2009. Calibration of a greenhouse climate model using evolutionary algorithms. *Biosystem Engineering* 104: 135-142. DOI: 10.1016/j.biosystemseng.2009.06.006
- [74] Hagan, M. T., H. B. Demuth, and M. Beale. 1996. *Neural Network Desing*. China Machine Press, Citic Publishing House.
- [75] Hahn, F. 2011. Fuzzy controller decreases tomato cracking in greenhouses. *Computers and Electronics in Agriculture*, 77:21-27.
- [76] Hanan, J. J. 1997. *Greenhouses: Advanced Technology for Protected Horticulture*. Kluwer Academic Publishers, The Netherlands.
- [77] Hansen, N., and A. Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. In *Evolutionary Computation*, 9(1): 159-195. DOI: 10.1162/106365601750190398
- [78] Hansen, N., S. D. Müller, and P. Koumoutsakos. 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). In *Evolutionary Computation*, 11(1): 1-18. DOI:10.1162/106365603321828970

- [79] Hansen, N., and A. Ostermeier. 1997. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The  $(\mu/\mu_I, \lambda)$ -CMA-ES, In Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing pp. 650-654.
- [80] Hansen, N., and S. Kern. 2004. Evaluating the CMA Evolution Strategy on Multimodal Test Functions, Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings, pp. 282-291, Berlin: Springer. DOI: 10.1007/978-3-540-30217-9\_29
- [81] Hansen, N. 2006. The CMA Evolution Strategy: A Comparing Review, *StudFuzz* 192: 75-102. DOI: 10.1007/3-540-32494-1\_4
- [82] Haykin, S. 1998. *Neural Networks: A Comprehensive Foundation*. 2nd Edition, Prentice-Hall, Englewood Cliffs, NJ, p. 299.
- [83] He, F., C. Ma, J. Zhang, and Y. Chen. 2007. Greenhouse Air Temperature and Humidity Prediction Based on Improved BP Neural Network and Genetic Algorithm. D. Liu et al. (Eds.): *ISNN 2007, Part III, LNCS 4493*, pp. 973-980, Springer-Verlag Berlin Heidelberg 2007.
- [84] He, F., and C. Ma. 2010. Modelling greenhouse air humidity by means of artificial neural networks and principal components analysis. *Computers and Electronics in Agriculture* 71: 19-23.
- [85] Herrero, J. M., X. Blasco, M. Martínez, C. Ramos, and J. Sánchez. 2007. Non-linear robust identification of a greenhouse model using multi-objective evolutionary algorithms. *Biosystems Engineering* 98: 335-346. DOI: 10.1016/j.biosystemseng.2007.06.004
- [86] Holland, J., H. 1975. *Adaptation in Natural and Artificial Systems*. Univ. Michigan Press, Ann Arbor, MI.
- [87] Holland, J., H. 1992. Genetic algorithms. *Sci. Am.* July: 44-50.
- [88] Huang, Y., Y. Lan, S. J. Thomson, A. Fang, W. C. Hoffmann, and R. E. Lacey. 2010. Development of soft computing and applications in agricultural and biological engineering. *Computers and electronics in Agriculture*, 71: 107-127.

- [89] Ioslovich, I., I. Seginer, and A. Baskin. 2002. Fitting the NICOLET lettuce growth model to plant-spacing experimental data. *Biosystems Engineering* 83(3): 361-371.
- [90] Ioslovich, I., and P. O. Gutman. 2007. Evaluation of experiments for estimation of dynamical crop model parameters. *Bulletin Math. Biol.* 69:1603-1614. DOI: 10.1007/s11538-006-9181-x
- [91] Jackson, D. E., and F. L. W. Ratnieks. 2006. Communication in ants. *Current Biology*, 16: 570-574.
- [92] Jang, J., S. R. 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23: 665-685.
- [93] Jang, J., S. R. 1995. Adaptive Neuro-Fuzzy Inference Systems (ANFIS) for Noise Cancellation. In *Proceedings of the International Joint Conference of CF-SA/IFIS/SOFT*, edited by Weiling Chiang and Jonathan Lee, 127-132, Singapore, World Scientific.
- [94] Jang, J. S. R., and S. Chuen-Tsai. 1995. Neuro- Fuzzy Modeling and Control. *Proceeding of the IEEE*.
- [95] Jones, C.A., and J. R. Kiniry. 1986. CERES-Maize: a simulation model of maize growth and development. Texas A& M University Press. College Station , TX. USA.
- [96] Kalinli, A., M. C. Acar, and Z. Gündüz. 2011. New approaches to determine the ultimate bearing capacity of shallow foundations based on artificial neural networks and ant colony optimization. *Engineering Geology* 117: 29-38.
- [97] Karaboga, D., and B. Akay. 2009. A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation* 214: 108-132. DOI: 10.1016/j.amc.2009.03.090
- [98] Karaboga, D., and B. Basturk. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39:459-471. DOI:10.1007/s10898-007-9149-x
- [99] Karaboga, D., and C. Ozturk. 2011. A novel clustering approach: Artificial Bee Colony (ABC) algorithm, *Applied Soft Computing* 11: 652-657. DOI:10.1016/j.asoc.2009.12.025

- [100] Kazheunikau, M., P. M. Ferreira, and A. E. Ruano. 2004. Neuro-fuzzy modeling of plant growth in a Hidroponic Greenhouse. Proceedings of CONTROLO'2004, the 6th Portuguese Conference on Automatic Control, June 2004, Faro, Portugal.
- [101] Kennedy, J., R. C. Eberhart, and Y. Shi. 2001. Swarm Intelligence. Morgan Kaufmann, San Francisco, CA.
- [102] Kennedy, J., and R. C. Eberhart. 1995. Particle Swarm Optimization. Proc. IEEE Int. Conf. on N.N., pp: 1942-1948.
- [103] Kia, P. J., A. T. Far, M. Omid, R. Alimardani, and L. Naderloo. 2009. Intelligent Control Based Fuzzy Logic for Automation of Greenhouse Irrigation System and Evaluation in Relation to Conventional Systems. World Applied Sciences Journal 6 (1): 16-23.
- [104] Kirk, E. D. 1998. Optimal control theory. An introduction. Dover Publications, Inc. 452 pp.
- [105] Koza, J., R. 1992. Genetic programming: On the programming of computers by means of natural selection. The MIT Press, USA.
- [106] Koza, J., R. 2003. Introduction to Genetic Programming. Fall 2003 BMI 226 / CS 426 Notes, D-1/D-45.
- [107] Koza, J., R. 2010. Human-competitive results produced by genetic programming. Genet Program Evolvable Mach, 11:251-284.
- [108] Krink, T., R. K. Ursem, and B. Filipic. 2001. Evolutionary Algorithms in Control Optimization: The Greenhouse Problem. In Proceedings of the Genetic and Evolutionary Computation Conference, pp. 440-447.
- [109] Lafont, F., and J. F. Balmat. 2002. Optimized fuzzy control of a greenhouse. Fuzzy Sets and Systems 128: 47-59.
- [110] Lafont, F., and J. F. Balmat. 2004. Fuzzy Logic to the identification and the command of the multidimensional systems (invited paper), International Journal of Computational cognition, Vol 2, Num 3:21-47.

- [111] Lanfang, P., W. Wanliang, and W. Qidi. 2000. Application of Adaptive Fuzzy Logic System to Model for Greenhouse Climate. Proceedings of the 3rd World Congress on Intelligent Control and Automation, June 28-July 2, Hefei, P.R. China.
- [112] Latimer and Scoggins. 2003. Greenhouse operator's training manual. Virginia Flower Grower Association. USA. 420 p.
- [113] Lee, K., H. 2005. First Course on Fuzzy Theory and Applications. Springer-Verlag Berlin Heidelberg.
- [114] Linker, R., I. Seginer, and P. O. Gutman. 1998. Optimal  $CO_2$  control in a greenhouse modeled with neural networks. Computers and Electronics in Agriculture 19: 289-310.
- [115] Linker, R., and I. Seginer. 1999. Adaptive hybrid physical/neural network modeling, and its application to greenhouse climate optimization. Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99) Haifa, Israel - June 28-30.
- [116] Linker, R., and I. Seginer. 2004. Greenhouse temperature modeling: a comparison between sigmoid neural networks and hybrid models. Mathematics and Computers in Simulation 65: 19-29.
- [117] Linker, R., P. O. Gutman, and I. Seginer. 2000. Robust model-based failure detection and identification in greenhouses. Computers and Electronics in Agriculture 26: 255-270.
- [118] Ljung, L., and T. Glad. 1994. Modeling of Dynamic Systems. P T R Prentice Hall, Englewood Cliffs, New Jersey.
- [119] López-Cruz, I., L. 2002. Efficient evolutionary algorithms for optimal control. PhD Thesis. Wageningen University. The Netherlands.
- [120] López-Cruz, I., L. 2004. Introducción a la simulación de crecimiento y desarrollo de cultivos usando For-tran simulation Translator (FST). Primer congreso nacional de fenología agrícola y modelos de simulación aplicados a la agricultura. Universidad de Chapingo, México.

- [121] López-Cruz, I. L., and L. Hernández-Larragoiti. 2010. Neuro-fuzzy models for Air Temperature and Humidity of arched and venlo type Greenhouses in central México. *Agrociencia* 44: 791-805.
- [122] López-Cruz, I. L., L. G. Van Willigenburg, and G. Van Straten. 2003a. Efficient Differential Evolution algorithms for multimodal optimal control problems. *Applied Soft Computing* 3: 97-122. DOI:10.1016/S1568-4946(03)00007-3
- [123] López-Cruz, I. L., L. G. Van Willigenburg, and G. Van Straten. 2003b. Optimal control of nitrate in lettuce by a hybrid approach: differential evolution and adjustable control weight gradient algorithms. *Computers and Electronics in Agriculture* 40: 179-197. DOI:10.1016/S0168-1699(03)00019-X
- [124] López-Cruz, I. L., A. Rojano-Aguilar, and A. Ramírez-Arias. 2008. Differential Evolution Algorithms: a Matlab implementation. In *Seminario de Ingeniería Agrícola. Departamento de Irrigación. Universidad Autónoma Chapingo, Chapingo, México.*
- [125] López-Cruz, I.L., A. Rojano-Aguilar, W. Ojeda-Bustamante, y R. Salazar-Moreno. 2007. Modelos ARX para predecir la temperatura de un invernadero: una metodología. *Agrociencia* 41(2): 181-192.
- [126] López-López, R., R. Arteaga-Ramírez, M. A. Vázquez-Peña, I. L. López-Cruz, and I. Sánchez-Cohen. 2009. Husk tomato (*Physalis ixocarpa* Brot.) production based on irrigation volume and plastic mulching, *Revista Chapingo Serie Horticultura*, 15: 83-89.
- [127] Makowski, D., J. Hillier, D. Wallach, B. Andrieu, and M. H. Jeuffroy. 2006. Parameter estimation for crop models, *Working with Dynamic Crop Models*, (D. Wallach, D. Makowski, J.W. Jones, eds.), Elsevier 55-100.
- [128] Mamdani, E., H. 1974. Applications of fuzzy algorithms for control a simple dynamic plant. *Proceedings of IEE* 121, 12: 1585-1588.
- [129] Mamdani, E. H., and S. Assilian. 1975. An experiment in linguistic synthesis with a fuzzy logic controller . *International Journal of Man-Machine studies* 7: 1-13.
- [130] Michalewicz, Z. 1996. *Genetic algorithms + Data structures = evolution programs.* Springer-Verlag. Berlin.

- [131] Munakata, T. 2008. *Fundamentals of the New Artificial Intelligence, Neural, Evolutionary, Fuzzy and More*. Second Edition, Springer-Verlag London Limited.
- [132] Nachidi, M., A. Benzaouia, and F. Tadeo. 2006. Temperature and humidity control in greenhouses using the Takagi-Sugeno fuzzy model. *Proceedings of the 2006 IEEE International Conference on Control Applications Munich, Germany, October 4-6*.
- [133] Niknam, T. 2008. A new approach based on ant colony optimization for daily Volt/Var control in distribution networks considering distributed generators. *Energy Conversion and Management* 49: 3417-3424.
- [134] Pabico, J. P., G. Googenhoom, and R. W. McClendon. 1999. Determination of cultivar coefficients of crop models using a genetic algorithm: a conceptual framework, *Am. Soc. Agric. Eng.* 42: 223-232.
- [135] Pasgianos, G. D., K. G. Arvanitis, P. Polycarpous, and N. Sigrimis. 2003. A nonlinear feedback technique for greenhouse environmental control. *Computers and Electronics in Agriculture* 40: 153-177.
- [136] Patil, S. L., H. J. Tantau, and V. M. Salokhe. 2008. Modelling of tropical greenhouse temperature by auto-regressive and neural network models. *Biosystems Engineering* 99: 423-431.
- [137] Patiño, H. D., J. Pucheta, and S. Tosetti. 2008. Approximate optimal control for crop production in intelligent greenhouses. *37 JAIIO, Jornadas de inf. Ind. Agroinformática (JII 2008 Agroinf)*, pp. 66-84.
- [138] Pessel, N., and J. F. Balmat. 2008. Principal Component Analysis for Greenhouse Modelling. *WSEAS TRANSACTIONS on SYSTEMS*, Issue 1, Volume 7.
- [139] Penning de Vries, F.W.T., D. M. Jansen, H. F. M. ten Berge, and A. Bakema. 1989. *Simulation of ecophysiological processes of growth in several annual crops*. Pudoc Wageningen. 271pp.
- [140] Pham, D. T., A. Afify, and E. Koc. 2007. Manufacturing cell formation using the Bees Algorithm. *IPROMS 2007 Innovative Production Machines and Systems Virtual Conference, Cardiff, UK*.

- [141] Pham, D. T., A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. 2006. The Bees Algorithm - A Novel Tool for Complex Optimisation Problems, Proceedings of IPROMS 2006 Conference, pp: 454-461.
- [142] Pohlheim, H., and A. Heiner. 1996. Optimal control of greenhouse climate using Genetic Algorithms. MENDEL'96 - 2nd International Conference on Genetic Algorithms. Brno, Czech Republik, pp. 112-119.
- [143] Pohlheim, H., and A. Heiner. 1999. Optimal Control of Greenhouse Climate using Real-World Weather Data and Evolutionary Algorithms. In Banzhaf, W. (ed.): GECOCO'99 - Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, CA: Morgan Kaufmann, p. 1672-1677.
- [144] Poli, P., J. Kennedy, and T. Blackwell. 2007. Particle swarm optimization an overview, *Swarm Intell*, 1: 33-57. DOI: 10.1007/s11721-007-0002-0
- [145] Ponce-Cruz, P. 2010. *Inteligencia Artificial con aplicaciones a la ingeniera*. Editorial Alfaomega, Mxico, 348 pgs.
- [146] Price, K. V., R. M. Storn, and J. A. Lampinen. 2005. *Differential Evolution. A practical Approach to Global Optimization*. Natural Computing Series, Springer-Verlang Berlin Heidelberg.
- [147] Price, K. V. 1999. An introduction to Differential Evolution. In Corne, Dorigo and Glover Editors. *New Ideas in Optimization*.
- [148] Poluektov, R.A., and A. G. Topaj. 2001. Crop modeling: nostalgia about present or reminiscence about future. *Agronomy Journal* 93: 653-659.
- [149] Qaddoum, K., E. Hines, and D. Illiescu. 2011. Adaptive Neuro-Fuzzy Modeling For Crop Yield Prediction. AIKED'11 Proceedings of the 10th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases, 198-204.
- [150] Rabbinge, R., and C. T. De Wit. 1989. Systems, models and simulation. In *Simulation and Systems Management in Crop Protection*. Edited by R. Rabbinge *et al.* Wageningen: Pudoc.

- [151] Ramos-Fernández, J. C., V. López-Morales, F. Lafont, G. Enea, and J. Duplaix. 2010. A Neurofuzzy Structure Modelling Evapotranspiration in a Greenhouse Crop. *Ingeniería Investigación y Tecnología*. Vol. XI. N.ºm.2: 127-139.
- [152] Rechenberg, I. 1994. <http://www.bionik.tu-berlin.de/institut/xs2evost.html>.
- [153] Riadi, R., R. Tawegoum, A. Rachid, and G. Chassériaux. 2007. Decentralized Temperature Fuzzy Logic Control of a Passive Air Conditioning Unit. Proceedings of the 15th Mediterranean Conference on Control and Automation, July 27-29, Athens-Greece.
- [154] Ritchie, J.T., and S. Otter. 1985. Description and performance of Ceres-wheat: a user-oriented wheat yield model. US Department of Agriculture. *ARS*, 38:159-175.
- [155] Robleño-Jara, D., y G. Gaytán-Hernández. 2005. Agricultura y Sistemas de Control Climático. *Revista Horticultura Internacional*.
- [156] Rojas, R. 1996. *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin.
- [157] Ross, P. 1999. *Neural Networks: an introduction*. AI Applications Institute University of Edinburgh, 168p. <http://www.itu.dk/courses/MVAI/E2008/notes.pdf>.
- [158] Russell, S. J., and P. Norvig. 1995. *Artificial Intelligence A Modern Approach*. Prentice Hall, Englewood Cliffs, New Jersey 07632.
- [159] Salazar-Moreno, R., A. Rojano-Aguilar, I. L. López-Cruz, U. Schmidt, and C. Huber. 2009. Temperature and  $CO_2$  prediction to control greenhouse environment. International Symposium on High Technology for Greenhouse System Management: Greensys2009.
- [160] Salazar-Moreno, R., A. Rojano-Aguilar, and I. L. López-Cruz. 2007. A Neural Network Model to Predict Temperature and Relative Humidity in a Greenhouse. *ISHS Acta Horticulturae* 801: 539-546. International Symposium on High Technology for Greenhouse System Management: Greensys2007.
- [161] Salgado, P., and J. B. Cunha. 2005. Greenhouse climate hierarchical fuzzy modelling. *Control Engineering Practice* 13: 613-628.

- [162] Salomon, R. 1996. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms, *Biosystems* 39: 263-278.
- [163] Saridakis, G., D. Kolokotsa, and S. Dolianitis. 2006. Development of an intelligent indoor environment and energy management system for greenhouses using a fuzzy logic controller and LonWorks protocol. International Workshop on Energy Performance and Environmental Quality of Buildings, July, Milos Island, Greece.
- [164] Schmid, C. 2005. Course on Dynamics of Multidisciplinary and Controlled Systems. <http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/main.html>.
- [165] Seginer, I. 1997. Some artificial neural network applications to greenhouse environmental control. *Computers and Electronics in Agriculture* 18: 167-186.
- [166] Seginer, I., T. Boulard, and B. J. Bailey. 1994. Neural network models of the greenhouse climate. *Journal of Agriculture Engineering Research*, 59: pp. 203-216.
- [167] Shi, Y., and R. C. Eberhart. 1999. Empirical study of particle swarm optimization. In: *Proceedings of the 1999 IEEE Congress of Evolutionary Computation*, vol. 3, pp. 1945-1950. DOI:10.1109/CEC.1999.785511
- [168] Sierakowski, C. A., and L. dos Santos Coelho. 2005. Study of two Swarm Intelligence techniques for path planning of mobile robots. *IFAC*.
- [169] Socha, K., and M. Dorigo. 2008. Ant colony optimization for continuous domains. *European Journal of Operational Research* 185: 1155-1173.
- [170] Sonmez, M. 2011. Artificial Bee Colony algorithm for optimization of truss structures. *Applied Soft Computing* 11: 2406-2418.
- [171] Souissi, M., M. Chaabane, and A. Toumi. 2005. Greenhouse Climate Discrete Control with Guaranteed  $H_\infty$  Performance. *International Journal of Computational Cognition* (<http://www.YangSky.com/yangijcc.htm>), Vol. 3, No. 4.
- [172] Spitters, C.J.T., H. van Keulen, and D.W.G. van Kraalingen. 1989. A simple and universal crop growth simulator: SUCROS87, en Rabbinge R., S.A. Ward and H.H. van Laar (editors), *Simulation and systems management in crop protection*, PUDOC, Wageningen.

- [173] Sriraman, A., and R. V. Mayorga. 2004. A Fuzzy Inference System Approach for Greenhouse Climate Control. *Environmental Informatics Archives*, Volume 2: 699-710.
- [174] Stanarevic, N., M. Tuba, and N. Bacanin. 2010. Enhanced Artificial Bee Colony Algorithm Performance. *Latest Trends on Computers (Volume II)*.
- [175] Storn, R., and K. Price. 1997. Differential Evolution - A simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11: 341-359.
- [176] Sugeno, M. 1985. An introductory survey on fuzzy control. *Information Sciences*, 36: 59-83.
- [177] Sundar, S., and A. Singh. 2010. A swarm intelligence approach to the quadratic minimum spanning tree problem. *Information Sciences* 180: 3182-3191.
- [178] Takagi, T., and M. Sugeno. 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man and Cybernetics* 15(1): 116-132.
- [179] Tap, F. 2000. Economics-based optimal control of greenhouse tomato crop production. PhD Thesis. Wageningen University. The Netherlands. 127 pp.
- [180] The MathWorks Inc., Optimization Toolbox User's guide Version 3, Natick, MA, USA, 1990-2006, 419 p.
- [181] Tien, B., T. 1997. Neural-Fuzzy approach for system identification. PhD Thesis. Wageningen University the Netherlands. 155 p.
- [182] Tien, B. T., and G. Van Straten. 1996. Recursive Prediction Error Algorithm for the NUFZY System to Identify Nonlinear Systems. In *Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE-96*, 569-574. Fukuoka, Japan, June 4-7.
- [183] Tien, B. T., and G. Van Straten. 1998. A Neuro-Fuzzy Approach to Identify Lettuce Growth and Greenhouse Climate. *Artificial Intelligence Review* 12: 71-93.

- [184] Tien, B. T., and G. Van Straten. 1995. Neural-Fuzzy Systems for Non-Linear System Identification-Orthogonal Least squares training Algorithms and fuzzy Rule Reduction. In Prepring of the 2nd IFAC/IFIP/EurAgEng Workshop on AI in Agriculture, 249-254. Wageningen, The Netherlands, May 29-31.
- [185] Thornley, J.H.M., and I. R. Johnson. 2000. Plant and crop modeling. A mathematical approach to plant and crop physiology, The Blackburn Press.
- [186] Thornley, J.H.M., and J France. 2007. Mathematical Models in Agriculture, Quantitative Methods for the Plant, Animal and Ecological Sciences. Second Edition, London, UK.
- [187] Trabelsi, A., F. Lafont, M. Kamoun, and G. Enea. 2007. Fuzzy identification of a greenhouse. *Applied Soft Computing* 7: 1092-1101.
- [188] Trejo-Perea, M., G. Herrera-Ruiz, J. Rios-Moreno, R. C. Miranda, and E. Rivas-Araiza. 2009. Greenhouse energy consumption prediction using neural networks models. *Int. J. Agric. Biol.*, 11: 1-6.
- [189] Trejo-Zúñiga, E. C., and I. L. López-Cruz. 2011a. Técnicas de inteligencia artificial en modelado y control del ambiente de invernaderos: Estado del Arte, Parte I. *Ingeniería Agrícola y Biosistemas* 3: 75-80.
- [190] Trejo-Zúñiga, E. C., and I. L. López-Cruz. 2011b. Técnicas de inteligencia artificial en modelado y control del ambiente de invernaderos: Estado del Arte, Parte II. *Ingeniería Agrícola y Biosistemas* 3: 81-86.
- [191] Trejo-Zúñiga, E. C., I. L. López-Cruz, and A. Ruíz-García. 2013. Estimación de parámetros para un modelo de crecimiento de cultivos usando algoritmos evolutivos y bio-inspirados. *Agrociencia* 47: 671-682.
- [192] Trejo-Zúñiga, E. C., I. L. López-Cruz, and A. Ruíz-García. 2013. Parameter estimation for crop growth model using evolutionary and bio-inspired algorithms. Enviado a la revista *Applied Soft Computing*.
- [193] Ursem, R. K., B. Filipic, and T. Krink. 2002. Exploring the Performance of an Evolutionary Algorithm for Greenhouse Control. *Journal of Computing and Information Technology - CIT* 10, 3: 195-201.

- [194] Van Laar, H. H., J. Goudriaan, and H. van Keulen. 1997. SUCROS97: simulation crop growth for potential and water-limited production situations, AB-DLO Wageningen, The Netherlands, 75 pag.
- [195] van Henten, E. J. 1994. Greenhouse climate management: an optimal control approach. PhD Thesis. Wageningen University. The Netherlands. 329 pp.
- [196] Van Henten, E. J. 2004. Greenhouse climate management: an optimal control approach. Greenhouse Technology Group Agrotechnology and Food Innovations B.V.
- [197] van Henten, E. J., and J. Bontsema. 1996. Greenhouse climate control: a two-time scale approach. *Acta Horticulturae* 406: 213-220.
- [198] van Henten, E.J. J. Bontsema, and G. van Straten. 1997. Improving the efficiency of greenhouse climate control: an optimal control approach. *Netherlands Journal of Agricultural Science* 45 : 109-125.
- [199] van Keulen, H., F.W.T. Penning de Vries, and E. M. Drees. 1982. A summary model for crop growth. En F.W.T. Penning de Vries and H.H. van Laar (Eds.) *Simulation of plant growth and crop production. Simulation monographs.* Wageningen.
- [200] van Straten, G., I. L. López-Cruz, I. Seginer, and F. Buwalda. 1999. Calibration and sensitivity analysis of a dynamic model for control of nitrate in lettuce. *Acta Horticulturae* 507:149-156.
- [201] van Straten, G., L. G. van Willigenburg, and R. F. Tap. 2002. The significance of crop co-states for receding horizon optimal control of greenhouse climate. *Control Engineering Practice* 10: 625-632.
- [202] van Straten, G., G. van Willigenburg, E. van Henten, and R. van Osteghem. 2011. *Optimal control of greenhouse cultivation.* CRC Press, Taylor & Francis Group.
- [203] van Straten, G. 2012. *Systems Concepts and Terminology - an Overview.* Notas del Curso Systems Dynamics for Bio-Engineers.
- [204] Vázquez-Cruz, M.A., R. Guzmán-Cruz, I. L. López-Cruz, I. Cornejo-Pérez, I. Torres-Pacheco, and R. G. Guevara-González, 2013. Global sensitivity analysis by

means of EFAST and Sobol methods and calibration of reduced state-variable TOM-GRO model using genetic algorithms. *Computers and Electronics in agriculture* 100: 1:12.

- [205] Wallach, D., D. Makowski, and J. W. Jones. 2006. *Working with Dynamic Crop Models: Evaluation, Analysis, Parameterization and Applications*, Elsevier Amsterdam.
- [206] Wallach, D., B. Goffinet, J. E. Berguez, D. Leenhardt, and J. N. Aubertot. 2001. Parameter estimation for crop models: a new approach and application to a corn model, *Agronomy Journal*, 93: 757-766.
- [207] Williams, J.R., C. A. Jones, J. R. Kiniry, and D. Spanel. 1989. The Epic crop growth model. *Transactions of ASAE* 32(2): 497-511.
- [208] Willis, M. J., H. G. Hiden, P. Marenbach, B. McKay, and G. A. Montague. 1997. *Genetic Programming: An Introduction and Survey of Applications*. Symbolic Optimisation Research Group (SORG), Dept. of Chemical and Process Engineering University of Newcastle upon Tyne NE1 7RU, UK.
- [209] Zadeh, L., A. 1965. Fuzzy sets. *Information and Control*, 8: 338-353.
- [210] Zadeh, L., A. 1988. Fuzzy Logic. *Computer*, 1(4): 83-93.