



Institución Universitaria

**DISTRIBUCIÓN DEL RIEGO Y EL  
FERTIRRIEGO POR GOTEO PARA UN  
MONOCULTIVO EN UN TERRENO PLANO  
CON RED HIDRÁULICA DE LAZO ABIERTO  
APLICANDO ALGORITMOS EVOLUTIVOS**

**Bayardo Emilio Cadavid Gómez**

Instituto Tecnológico Metropolitano  
Facultad de Ingenierías  
Medellín, Colombia  
Julio, 2015



# **DISTRIBUCIÓN DEL RIEGO Y EL FERTIRRIEGO POR GOTEO PARA UN MONOCULTIVO EN UN TERRENO PLANO CON RED HIDRÁULICA DE LAZO ABIERTO APLICANDO ALGORITMOS EVOLUTIVOS**

**Bayardo Emilio Cadavid Gómez**

Tesis de investigación presentada como requisito parcial para optar al título  
de:

**Magister en Automatización y Control Industrial**

Director

Ph.D., Jesús Hernández Rivero

Codirector

Ph.D., Jorge Alberto Jaramillo Garzón

Línea de Investigación:

Máquinas Inteligentes y Reconocimiento de Patrones

Grupo de Investigación:

Automática, Electrónica y Ciencias Computacionales

Instituto Tecnológico Metropolitano

Facultad de ingenierías

Medellín, Colombia

Julio, 2015

Dedico este gran logro a los pilares de mi existencia:  
Laura, Alejandro, mis padres y mis hermanos.

Y a mi Aura del alma.

## **Agradecimientos**

A Jorge Alberto Jaramillo Garzón profesor del Instituto tecnológico Metropolitano y a Jesús Antonio Hernández Rivero, profesor de la Universidad Nacional de Colombia por sus valiosos aportes como directores de esta tesis de maestría, a su motivación constante para terminar este trabajo. A ellos mi gratitud.



## TABLA DE CONTENIDO

<b>Introducción.....</b>	<b>17</b>
<b>Estado del arte.....</b>	<b>22</b>
<b>Objetivos.....</b>	<b>29</b>
<b>1. Marco teórico.....</b>	<b>30</b>
1.1  Sistemas de riego.....	30
1.2  Componentes de un sistema de riego por goteo .....	34
1.3  Estrategia empírica para la distribución de una lámina de riego .....	37
1.4  Optimización .....	40
1.5  Metaheurísticas.....	43
1.5.1  Algoritmo genético GA .....	45
1.5.2  Optimización por enjambre de partículas <i>PSO</i> .....	47
1.5.3  Búsqueda <i>cuckoo</i> (CS).....	50
1.6  Grafos y dígrafos.....	52
1.6.1  Grafos dirigidos y no dirigidos.....	53
1.6.2  Definiciones en los grafos.....	54
1.6.3  Representación de los grafos .....	55
<b>2. Marco experimental.....</b>	<b>57</b>
2.1  Metodología Propuesta. ....	57
2.1.1  Características y parámetros del sistema típico de riego objeto de estudio ...	58
2.1.2  Modelamiento del cultivo típico con riego por goteo obtención de la función de aptitud.....	63
2.1.3  Metodología de optimización aplicando técnica de computación evolutiva. ...	73
2.1.4  Validación de los resultados .....	74
2.2  Materiales y métodos.....	75
2.2.1  Algoritmo <i>Cuckoo Search</i> en Matlab.....	75
2.2.2  Algoritmo genético <i>Global Optimization toolbox</i> de Matlab.....	76
2.2.3  Optimización por enjambre de partículas <i>PSO</i> , <i>Trelea</i> tipo2 .....	77
2.3  Resultados .....	80
2.3.1  Simulación con el algoritmo CS .....	81
2.3.2  Simulación con el algoritmo genético (GA) .....	85
2.3.3  Simulación con el algoritmo optimización por enjambre de partículas ( <i>PSO</i> ). ..	89
2.4  Comparación de los resultados entre las estrategias evolutivas .....	94
<b>3. Discusión y Conclusiones .....</b>	<b>97</b>
3.1  Aporte de la metodología propuesta.....	97
3.2  Conclusiones.....	98
3.3  Trabajo futuro.....	100
<b>A. Anexo: Simulaciones realizadas con el Algoritmo Genético. ....</b>	<b>102</b>
<b>B. Anexo: simulaciones con el <i>PSO</i>.....</b>	<b>108</b>
<b>C. Anexo: Simulaciones con el algoritmo <i>Cuckoo Search</i> .....</b>	<b>119</b>

---

<b>D. Anexo: Especificaciones de los componentes de la red hidráulica .....</b>	<b>125</b>
<b>E. Anexo: Aproximación al cálculo de una lámina de riego .....</b>	<b>126</b>
<b>F. Anexo: Funciones y rutinas de llamado a los algoritmos evolutivos.....</b>	<b>128</b>
<b>Bibliografía .....</b>	<b>133</b>

## ÍNDICE DE FIGURAS

Figura 1-1:	Riegos superficiales .....	30
Figura 1-2:	Riego por aspersión .....	31
Figura 1-3:	Riego localizado .....	32
Figura 1-4:	Tubería abierta o en lazo abierto .....	33
Figura 1-5:	Red de tubería en bucle .....	33
Figura 1-6:	Red de dos tuberías: riego y fertirriego.....	34
Figura 1-7:	Esquema de un cultivo con un sistema de riego por goteo.....	35
Figura 1-8:	Diagrama de flujo algoritmo <i>PSO</i> [40, 41]. .....	49
Figura 1-9:	Representación gráfica de un grafo. (a) Grafo dirigido o dígrafo, (b) Grafo no dirigido (c) Grafo acíclico dirigido.....	54
Figura 1-10:	Representación Matricial de un grafo. (a) Grafo dirigido o dígrafo, (b) Grafo no dirigido, (c) Grafo acíclico dirigido ponderado.....	56
Figura 2-11:	Diagrama de flujo de la metodología utilizada.....	57
Figura 2-12:	Esquema simplificado de un cultivo con riego por goteo.....	58
Figura 2-13:	Grafo representativo del cultivo donde los nodos son las válvulas y las aristas las distancias .....	64
Figura 2-14:	Matriz de pesos o de distancias.....	64
Figura 2-15:	Matriz de unidades básicas de longitud o volumen (“Lotes”).....	65
Figura 2-16:	Matriz de nodos y lotes .....	66
Figura 2-17:	Error y desperdicio en riego y en fertirriego con el <i>CS</i> .....	82
Figura 2-18:	Error calculado por el algoritmo <i>eval</i> para todas las cadenas de lotes ...	82
Figura 2-19:	Error y desperdicio en riego y en fertirriego con el <i>GA</i> .....	86
Figura 2-20:	Error en lámina y error de minimización según cadena de lotes enviados. ....	86
Figura 2-21:	Error y desperdicio en riego y en fertirriego con el <i>PSO</i> .....	90
Figura 2-22:	Error calculado por la función de aptitud para cada envío de lotes con el <i>PSO</i> .....	90
Figura 2-23:	Tendencia del mejor valor para cada época. ....	91
Figura A-24:	Simulación 1 AG con Población 300 individuos; Generaciones 200 ....	103
Figura A-25:	AG Errores lámina y error algoritmo <i>eval</i> .....	103
Figura A-26:	Simulación 2. GA con Población 500 individuos; Generaciones 200....	104

---

Figura A-27:	A. G Errores lámina y error <i>eval</i> .....	104
Figura A-28:	Simulación 3. AG con Población 1000 individuos; Generaciones 200..	105
Figura A-29:	A. G Errores lámina y error <i>eval</i> .....	105
Figura A-30:	Simulación AG. Población 1500 individuos, Generaciones 200 .....	106
Figura A-31:	AG Errores lámina y error <i>eval</i> .....	106
Figura A-32:	Simulación 5. GA con Población 3000 individuos; Generaciones 200..	107
Figura A-33:	AG Errores lámina y error <i>eval</i> .....	107
Figura B-34:	Error y desperdicio PSO Épocas: 200; Población: 100 .....	109
Figura B35:	Error en riego y fertirriego .....	110
Figura B-36:	Error de minimización PSO Épocas: 200; Población: 100 .....	110
Figura B-37:	Error y desperdicio Épocas: 200. Población: 300 .....	111
Figura B-38:	Error en riego y fertirriego .....	112
Figura B-39:	Comportamiento del error de minimización PSO .....	112
Figura B-40:	Error y desperdicio Épocas 200; población 500 .....	113
Figura B-41:	Erro en riego y fertirriego Épocas: 200; Población: 100 .....	114
Figura B-42:	Error de minimización Épocas: 200; Población: 100 .....	114
Figura B-43:	Error y desperdicio. Épocas: 500; Población: 1000 .....	115
Figura B-44:	Error en riego y fertirriego. Épocas: 500 ; Población: 1000 .....	116
Figura B-45:	Error de minimización. Épocas: 500; Población: 1000 .....	116
Figura B-46:	Error y desperdicio. Épocas 200. Población 2000 .....	117
Figura B-47:	Error en riego y fertirriego. Épocas: 200 ; Población: 2000 .....	118
Figura B-48:	Error de minimización. Épocas 200. Población 2000 .....	118
Figura C-49:	Error y desperdicio. N = 100 .....	120
Figura C-50:	Error en riego y fertirriego N = 100.....	120
Figura C-51:	Error y desperdicio. N= 300 .....	121
Figura C-52:	Error en riego y Fertirriego. N= 300 .....	121
Figura C-53:	Error y desperdicio. N = 500 .....	122
Figura C-54:	Error en riego y fertirriego. N = 500.....	122
Figura C-55:	Error y desperdicio. N = 1000 .....	123
Figura C-56:	Error en riego y fertirriego. N = 1000.....	123
Figura C-57:	Error y desperdicio. N = 1500 .....	124
Figura C-58:	Error en riego y fertirriego. N = 1500.....	124

---

Figura F-59: Pseudocódigo del programa principal para llamar al algoritmo <i>Cuckoo Search</i> .	128
Figura F-60: Pseudocódigo del programa principal para llamar al algoritmo Genético. . .	129
Figura F-61: Pseudocódigo del programa principal para llamar al algoritmo <i>PSO</i> ....	131

---

## ÍNDICE DE TABLAS

Tabla 2-1:	Longitudes entre nodos.....	60
Tabla 2-2:	Programa de riego y fertirriego (Lámina deseada) para distribuir en el cultivo. ....	80
Tabla 2-3:	Cadena de 27 lotes para realizar la distribución de la lámina con el CS....	84
Tabla 2-4:	Resumen de la distribución de la lámina programada con el CS.....	85
Tabla 2-5:	Cadena de 26 lotes para realizar la distribución de la lámina con el AG....	88
Tabla 2-6:	Resumen de la distribución entregada por el AG .....	89
Tabla 2-7:	Cadena de 26 lotes para realizar la distribución de la lámina de riego con el <i>PSO</i> .....	92
Tabla 2-8:	Resumen de la distribución de la lámina de riego fertirriego programada con el <i>PSO</i> . ....	94
Tabla 2-9:	Comparación de resultados entre las estrategias evolutivas <i>GA</i> , <i>PSO</i> , <i>CS95</i>	
Tabla E-10:	Programa de riego fertirriego (Lámina deseada) a distribuir en el cultivo. ....	126
Tabla E-11:	Lámina deseada expresada en unidades de lotes.....	127

---

## ÍNDICE DE ALGORITMOS

Algoritmo 2-1:	Pseudocódigo algoritmo <i>ruta</i> .....	71
Algoritmo 2-2:	Pseudocódigo algoritmo <i>eval</i> .....	72

## Resumen

Este trabajo tiene como objetivo proponer una solución al problema de la distribución más adecuada del riego y el fertirriego programado, para un cultivo con riego por goteo automatizado con red hidráulica de lazo abierto y plántulas de diferentes estados fenológicos. La metodología construida para la solución es la siguiente: se caracterizó un cultivo típico de manera numérica y se representó matemáticamente mediante un grafo acíclico dirigido en forma matricial. Posteriormente se diseñó e implementó una función de evaluación para cuantificar el error y el desperdicio ocasionado para distribuciones dadas de la lámina y se desarrolló un proceso de optimización para encontrar la mejor distribución mediante herramientas meta-heurísticas. La estrategia se validó con tres herramientas diferentes: algoritmos genéticos, optimización por enjambres de partículas y búsqueda *cuckoo*. Los resultados indican que la metodología propuesta permite encontrar la distribución más adecuada de una lámina programada de riego y fertirriego con un error y desperdicio mínimo, alcanzando soluciones consistentes con las diferentes herramientas meta-heurísticas, lo que prueba su robustez. Este aporte es importante porque permite aumentar la uniformidad del riego-fertirriego, optimizar la logística de la fertirrigación e implementar esta técnica a los programadores/controladores de riego.

**Palabras clave:** Distribución del fertirriego, algoritmos evolutivos, búsqueda *cuckoo*, algoritmos genéticos, optimización por enjambres de partículas, red hidráulica, teoría de grafos.

## Abstract

This work proposes a solution to the problem of finding the most suitable distribution of programmed irrigation and fertigation, for an automated drip irrigation crop with open loop hydraulic network and seedlings of different phenological stages. The methodology used for the solution process is the following: a typical crop was numerically characterized and mathematically modeled by a directed acyclic graph in matrix form. Then, an evaluation function for quantifying the error and waste caused by given distributions was developed and implemented, and an optimization process based on meta-heuristic tools was developed in order to find the optimal distribution. The strategy was validated using three meta-heuristic tools: Genetic Algorithms, Particle Swarm Optimization and Cuckoo Search. The results indicate that this methodology allows to find the most suitable solution of a programmed irrigation and fertigation distribution with minimal waste and failure, reaching comparable solutions with the three meta-heuristics. This results allow to increase the uniformity of fertigation, optimize the logistics of fertigation and to implement this strategy on irrigation computers/controllers.

**Keywords:** Fertigation distribution, evolutionary algorithms, cuckoo search, genetic algorithms, particle swarm optimization, hydraulic network graph, graph theory.



# Introducción

Hay varias razones por las que este trabajo presenta importancia. La primera, es que plantea desde el punto de vista teórico el problema de la distribución de un volumen de agua o fertirriego [1] a aplicar a un lote de plántulas (lámina) a través de una red de lazo abierto. La segunda, presenta una metodología versátil y rápida para encontrar una solución a la distribución de una lámina. Una tercera, es que inicia el aporte de información nueva, a un tema que no ha tenido mucho tratamiento público. Por lo cual la literatura existente respecto a la distribución de una lámina de riego con la situación problemática que se describirá en este trabajo no es abundante [2, 3].

En Colombia la mayoría de los grandes cultivos como son los destinados a la producción de flores, están implementados en terrenos con topología plana o con algún grado de inclinación [1, 3]. Muchos de ellos tienen riego automatizado y más concretamente el riego por goteo, pero presentan situaciones problemáticas como las siguientes: las redes hidráulicas consisten en una sola tubería de distribución en lazo abierto, prioridad a las ordenes de producción, los lotes de plántulas sembradas que presentan diversas condiciones fenológicas, por ejemplo: lotes de plántulas con diferentes tiempos de siembra que demandan diferentes requerimientos de riego o fertirriego y áreas que no tendrán siembra.

Una lámina de riego y fertirriego generalmente la calcula y la ajusta el agrónomo mediante alguna de las metodologías que ofrece la ingeniería agronómica [1, 3, 4]. En los sistemas automatizados, al tenerse calculada la lámina debe procederse a planear como enviarse y distribuirse a las plántulas a través de la red hidráulica, lo cual se programa en un controlador/programador de riego. El envío y la distribución de la lámina consiste en hacer llegar, un volumen de agua (riego) [1], y/o fertilizante más agua (fertirriego) [1] a los lotes de plántulas, mediante la determinación previa, de cual válvula del cultivo debe activarse y el tiempo que debe permanecer en este estado, así como el momento y el intervalo de tiempo que dura la inyección de fertilizante a la tubería. Hacer esta distribución y generar el menor desperdicio posible (este se presenta al desalojar riego o fertirriego no programado dentro o fuera del cultivo) es muy difícil debido a la problemática que se ha descrito líneas arriba. Por lo tanto, planear una estrategia de distribución es un reto que demanda tiempo y experiencia en este asunto.

El propósito de este trabajo fue encontrar una solución para la distribución del riego y el fertirriego en un cultivo que presenta las siguientes características: terreno plano, riego por goteo automatizado, red hidráulica en lazo abierto, lotes de plántulas con diferentes tiempos de siembra, y áreas sin plántulas.

Para lograr este propósito se establecieron cuatro estrategias. La primera, fue seleccionar los componentes con sus parámetros y especificaciones de un sistema típico de riego por goteo. La segunda estrategia consistió en representar el sistema típico mediante un grafo en su forma gráfica, simbólica y matricial y la construcción de una función de aptitud que permitiera optimizar la operación de la distribución del riego y el fertirriego. La tercera estrategia fue determinar una metodología de optimización basada en algoritmos evolutivos que generara las posibles soluciones para el envío y distribución de riego y el fertirriego. La cuarta en validar la solución alcanzada, mediante otras técnicas de optimización evolutiva.

Para la primera estrategia se tuvieron en cuenta los siguientes componentes: la red hidráulica radial de lazo abierto, el sistema de inyección de fertilizante ubicado fuera del punto de bombeo de agua, las válvulas de riego y las válvulas para el desperdicio [1, 3, 5].

La segunda estrategia consistió en representar el sistema típico mediante un grafo en su forma gráfica, simbólica y matricial y en la elaboración de una función de aptitud que permitiera optimizar la operación de la distribución del riego y el fertirriego. El análisis del sistema típico permitió encontrar que no se constituían circuitos cerrados para el flujo y que este era unidireccional. Características que determinaron representar el modelo del sistema típico mediante un grafo acíclico dirigido. La representación gráfica se describe mediante nodos y aristas dirigidas, para el sistema típico, los nodos se relacionaron con las bifurcaciones de la tubería, las válvulas, el punto de inyección de agua y de fertilizante, las aristas con las distancias de la tubería entre los nodos. La representación gráfica facilitó observar globalmente esta relación. La representación simbólica del grafo se expresó conceptualmente mediante un conjunto de pares ordenados con la relación existente entre el conjunto de nodos y el de aristas. La representación matricial permitió expresar el sistema típico con los parámetros de la red en forma matemática mediante

---

una matriz de dimensión  $n \times 3$ . Se definieron como parámetros de la red hidráulica, las longitudes de la tubería, la velocidad del fluido, el volumen por unidad de longitud y el tiempo que utiliza un volumen para ir de un nodo al siguiente o en salir de la tubería. Los anteriores parámetros se utilizaron para definir y calcular un “lote” (o *batches*). Este se definió como la unidad básica empleada para expresar volumen de riego o de fertirriego que ocupa una longitud  $L$  de la tubería y tarda un tiempo  $t$  en salir de la tubería. Con las anteriores definiciones la matriz se configura de la siguiente forma: los dos primeros elementos de los  $n$  vectores fila correspondían a nodos consecutivos y el tercer elemento al número de “lotes” entre estos dos nodos. La matriz de nodos y “lotes” obtenida presentaba la ventaja de poder ser declarada fácilmente en lenguaje de programación de Matlab.

La función de aptitud, se conformo con el algoritmo *ruta* y el algoritmo *eval*. Implementados mediante un programa en el lenguaje de Matlab. Con el fin de permitir una reproducibilidad de los resultados, su uso en trabajos futuros y modificaciones a su planteamiento se elaboro un pseudocódigo detallado para cada algoritmo. El insumo principal para los algoritmos fue la matriz de nodos y “lotes”, que permitiría al algoritmo *ruta* entregar el camino para llegar a una válvula del cultivo y al algoritmo *eval*, desplazar los lotes por la red, y luego determinar el error generado con respecto a una lamina deseada y el desperdicio ocasionado. Para calcular el error y el desperdicio a minimizar a cada solución generada por el algoritmo evolutivo, se construyeron las correspondientes funciones con sus restricciones. Una solución consistió en un vector de números reales que al ser decodificado por la función de aptitud entregaba un arreglo de dos vectores: el primero era una palabra binaria, donde, un cero (0) correspondía a un “lote” de riego y un uno (1) correspondía a un “lote” de fertirriego. El segundo, un vector con las válvulas del cultivo, estando asociadas a uno o varios bits de la palabra binaria. A las válvulas les fue asignado un número que las identificaba y les daba ubicación en el cultivo.

La tercera estrategia consistió en seleccionar una metodología de optimización que basada en algoritmos evolutivos que generara las posibles soluciones. En esta dirección se realizo la búsqueda de información sobre estas algoritmos [6, 7, 8, 9, 10, 11, 12, 13]. Se encontró que el algoritmo evolutivo búsqueda *Cuckoo* [13] inspirado en el comportamiento reproductivo de las aves *Cuckoo* Ani y Guira, presenta una metodología

basada en una población [14], de fácil entendimiento, con un número reducido de parámetros de ajuste y la evidencia en otros trabajos que entrega buenas soluciones. [15, 16]. A lo anterior se le agrega que esta implementado en Matlab, en una versión libre y robusta construida por el autor del algoritmo [17].

La cuarta estrategia consistió en validar la solución encontrada mediante otras técnicas de optimización evolutiva. Con este fin se utilizó el algoritmo genético (GA) [7, 11] y la optimización por enjambre de partículas (PSO) [19,26]. Estos algoritmos se seleccionaron debido a que han sido bastante empleados en la solución de problemas de optimización entregando buenas soluciones.

La búsqueda de la solución se restringió a un cultivo con solo cuatro válvulas de riego debido a las especificaciones técnicas y agronómicas para el cultivo típico seleccionado como referencia para este trabajo. Como nuevo se propuso la ubicación de válvulas de desperdicio al final de los ramales, con el fin de controlar el desalojo de riego o fertirriego fuera del cultivo. La solución se aborda como un problema que pertenece a la clase NP-completo o de complejidad polinomial no determinista [5], es decir el tiempo requerido para solucionar el problema aumenta exponencialmente con respecto al aumento de las variables que definen la cantidad de combinaciones de las soluciones. El número de combinaciones de lotes, o sea soluciones, podría ser grande si ocurre cualquiera de la siguientes situaciones o la conjunción entre ellas: disminución del tamaño de la unidad básica de riego y fertirriego que es el "lote", aumento de la lámina de riego a programar, el incremento la longitud de la tubería, aumento de la cantidad de válvulas del cultivo. Pero para las características del cultivo típico tratado en este trabajo, la lámina a programar y el número de válvulas, es moderado. El problema se resuelve como clase P, es decir "rápidamente" [5]. La evaluación de un cultivo que genere un problema NP – completo, se propone como trabajos futuros.

Los resultados de este trabajo muestran que se puede llegar a disponer de una herramienta versátil, con la que los técnicos encargados de fertirriego ahorren tiempo y esfuerzos en la programación y distribución de una lámina de riego. Los costos operacionales pueden verse sensiblemente reducidos con el mejoramiento de los sistemas de distribución del riego y del fertirriego, debido al ahorro en el bombeo de agua

y la eliminación del desperdicio de fertilizante. La entrega oportuna en cantidad y calidad proporciona un aumento en la eficiencia de la aplicación del riego y el fertirriego mediante el incremento del coeficiente de uniformidad [1, 29, 50, 51], impactando sobre el uso eficiente de los recursos como el agua y los nutrientes y por lo tanto en el aumento de la productividad y la calidad de las cosechas, que en el marco de una economía de mercado como es la estrategia dada para la agricultura Colombiana [34], se debe de proteger a los agricultores del deterioro de las rentas que ocasiona un uso ineficiente del fertirriego.

## Estado del arte

A través del tiempo se han desarrollado una gran cantidad de sistemas para la determinación, control y automatización del riego que permiten un consumo de agua más reducido y los nutrientes necesarios para las plántulas. Se han dedicado esfuerzos a lograr el momento oportuno y determinar la cantidad de riego necesario, mediante la utilización de las tecnologías de la información. Estos sistemas han sido de gran utilidad y han funcionado en sistemas productivos con estandarización de sus procesos.

Sin embargo, ha sido complejo adecuar estos sistemas a condiciones particulares de los cultivos, como es la heterogeneidad en la siembra. Son diversas las situaciones de producción que se presentan en un cultivo, donde el desarrollo de estas tecnologías de riego no han llegado o no han dado solución [20]. El momento oportuno y la cantidad en el riego son factores que se ven afectados, debido a que muchas veces no se dispone de una estrategia fácil y versátil para distribuir y entregar el riego y fertirriego y más cuando se dispone de una sola red hidráulica radial o de lazo abierto.

Al respecto y en la línea de las soluciones tecnológicas al riego, Ed-dahhak, Guerbaoui y otros [21], implementan un sistema de control de riego mediante lógica difusa, para un cultivo en un invernadero con riego por goteo, con la capacidad de elaborar registros históricos de la actividad del riego. Para ello utiliza la medición de variables como la humedad del suelo y la temperatura del invernadero para decidir la dosis de nutriente que hay que depositar en las plantas y controlar el crecimiento del cultivo. En este trabajo Ed-dahhak, Guerbaoui y otros [21] resaltan la importancia de realizar el riego varias veces al día y concluyen que con las reglas de la lógica difusa es posible controlar eficientemente el riego y mantener la humedad del suelo en un nivel apto sin que demande grandes cantidades de agua. Sin embargo, aunque Ed-dahhak, Guerbaoui y otros [21] mencionan la importancia del riego frecuente y proponen un método de control del riego, en este trabajo no hacen mención, si con la estrategia de control utilizada se pueda enviar varios insumos al cultivo simultáneamente, cuando se presentan varias fenologías en el cultivo o cambios en las órdenes de producción.

Jiménez y Martínez [22] presentan una red ramificada de lazo abierto para la distribución de agua y fertirriego simultáneamente para 380 parcelas en una superficie de 116 hectáreas. Desarrollaron una metodología basada en algoritmos genéticos (GA) y modelos hidráulicos para una red presurizadas para reducir al mínimo la cantidad de energía utilizada, por metro cubico de agua bombeada, mediante el agrupamiento de los puntos de suministro en sectores eficientes energéticamente. El trabajo presento un modelo matemático sectorizado de la red, de ocho ecuaciones, cada ecuación correspondía a un hidrante seleccionado a largo de la red, elaborándose con ellas una función objetivo que permitiera encontrar el mínimo error entre la presión estimada y la medida para determinar más exactamente el tiempo de viaje de los insumos. Como resultado, en los sectores de riego se podrían establecer la reducción al mínimo de la cantidad de energía utilizada por metro cubico de agua bombeada.

Este trabajo de Jiménez y Martínez [22] es interesante porque abordan el envío de varios insumos al mismo tiempo por la red, pero no se logro dar una solución completa a esta condición de operación. Precisan que el sistema necesitaba que uno solo de los insumos estuviera fluyendo por la tubería, debido a que no era posible garantizarles a los usuarios que necesitaban riego, que solamente este insumo llegará después de una jornada de fertirrigación. Jiménez y Martínez [22] concluyen que la única forma de gestionar la fertirrigación central en las redes presurizadas es sectorizando las demandas en turnos programados, que los modelos hidráulicos en conjunto con los algoritmos genéticos son herramientas para mejorar la gestión y el diseño de la fertirrigación.

Un diseño que involucra necesidades de riego diferentes es el que presenta Morari y Giardini [23] proponen un sistema de control que realice el fertirriego del jardín botánico de Padua, Italia. La característica de este sistema es realizar el fertirriego para vegetación heterogénea. La automatización consideró cuatro partes:

- Una red de riego con una unidad electrónica de control que consiste en un micro computador.
- Sistema de monitoreo de la humedad y profundidad del agua en el suelo.
- Administración del software para monitoreo y programación del riego.
- Un sistema de adquisición de datos que conecte los sensores al computador de riego.

Debido a la fenología de las plantas, Morari y Giardini [23] proponen una estrategia de riego distribuyendo el área del jardín en seis macro sectores, clasificados según la evapotranspiración, el tipo de cubierta (techo) y la necesidad de agua de la planta. De acuerdo a las anteriores necesidades, se podían programarse frecuencias muy definidas de riego para cada válvula mediante el intervalo de tiempo en que estaría activa o el volumen de agua que dejaría salir. Los problemas a la estrategia de Morari y Giardini [23] surgieron se debieron a la dificultad de tomar en cuenta la variabilidad del comportamiento de las diferentes plantas. Morari y Giardini [23] concluyen que desde el punto de vista operativo y técnico el sistema automatizado funcionó adecuadamente, permitiendo que con la información procedente de los sensores de medición de humedad se pudiera poner fin al riego. Sin embargo, el artículo Morari y Giardini [23] no mencionan varios aspectos importantes en la gestión del riego, como es si la estrategia de control permite enviar agua y fertilizante al mismo, como es el tipo de red hidráulica utilizada para la distribución de los insumos, después de cada ciclo de riego que hacer si queda fertilizante en la red hidráulica.

Las dos anteriores investigaciones se han acercado al problema de enviar insumos diferentes simultáneamente a través de la red hidráulica para el riego en cultivos con heterogeneidad fenológica en las plántulas. Pero se encuentra que la mayor parte de las investigaciones y desarrollos en estas tecnologías de riego están orientadas a encontrar las dosis de riego y fertirriego y la frecuencia para suministrarla cuando las características del cultivo son homogéneas.

Al respecto Zhao y Guan [24] realizaron un estudio sobre la precisión del riego automatizado en el control de la fisiología de las plantas y el ahorro de agua. Como resultado Zhao y Guan [24] desarrollaron un sistema de monitoreo y de control para diferentes sistemas de riego como el riego por goteo, por aspersión y el microrriego mediante sensores con transmisión inalámbrica que funcionaba en tiempo real registrando los eventos fisiológicos de las plantas y del contenido del agua del suelo. La información obtenida del cultivo se llevaba a una base de datos, siendo utilizada en análisis estadísticos en la generación de informes y acciones sobre el riego. Zhao y Guan [24] concluyen que un diseño flexible de software y hardware hacen la operación del

sistema de riego confiable y con notables efectos en el ahorro de agua. Que la eficiencia de este sistema sería un gran aporte a la grave situación de escasez de agua en muchas zonas de china [24].

En la misma dirección Castro, Águila y otros [25] desarrollaron un sistema de riego automatizado en tiempo real utilizando el lenguaje de programación Edlog para determinar el momento oportuno y la cantidad de riego, mediante tres estrategias: midiendo el flujo de agua en el suelo (lisímetro), determinando la cantidad volumétrica de agua en el suelo y calculando la evapotranspiración, todas monitoreadas por medio de tecnologías de información (TI). Castro, Águila y otros [25] construyeron para la parte experimental una cama de siembra con dimensiones de 68.8 m de largo por 1.4 m. El sistema de control implementado podía activar bombas y válvulas según las condiciones y necesidades del cultivo. Castro, Águila y otros [25] concluyen que la integración de información del hardware y el software interactuaron de acuerdo a lo programado dentro del sistema de riego, lo que permitió el correcto manejo y control, de las tres estrategias implementadas, además que el control del riego se puede realizar de manera automática en tiempo real mediante el uso de diversas tecnologías de información.

Salas y Urrestarazu [4] argumentan que los métodos de fertirriego deben de responder a dos cuestiones fundamentales: cuanta cantidad a aplicar y cuando regar (frecuencia de riego). Salas y Urrestarazu [4] hacen una clasificación de las metodologías utilizadas para realizar el fertirriego, la mayoría actúan sobre la frecuencia del riego, sobresaliendo los métodos asociados a las plántulas. Entre estos se encuentra: el nivel de transpiración que se mide en las variaciones del tallo; el método del sustrato que utiliza un tensiómetro; y el método de la balanza que utiliza el peso de una unidad de cultivo. Para determinar la cantidad de fertirriego predominan las metodologías basadas en la medición del clima y la radiación solar. En los métodos descritos y analizados por Salas y Urrestarazu [4] se pone atención en la cantidad y en la oportunidad del fertirriego que se entrega a la plántula. Sin embargo no incluyen en este análisis la incidencia de las dificultades en el transporte del riego y del fertirriego por las redes hidráulicas que utilizan los diferentes sistemas de riego.

La efectividad de los sistemas automáticos de riego depende de su diseño, de la selección de los equipos y del control que se ejerce sobre ellos. En este sentido Duran, Puig-Bargue's y otros [17] proponen un sistema de supervisión y de control para monitorear el microrriego y controlar cuatro unidades de filtración y cada una de ellas alimentando veinticuatro laterales de riego. El sistema de control y adquisición de datos SCADA se dedica exclusivamente a activar el riego y a medir el flujo y la presión en el cabezal de bombeo determinando las caídas de presiones en la tubería y con ello prevenir la obstrucción de las microválvulas (goteros) ejecutando el lavado de los filtros. Con esta herramienta que proponen Duran, Puig-Bargue's y otros [17] es rápido detectar fallas y anomalías en los equipos y útil para evaluar el rendimiento del sistema. Pero el sistema de control no presenta una solución a la distribución del riego.

La uniformidad en el riego es un índice para expresar la efectividad de riego. El sistema de red hidráulica por el cual se dirigen los insumos a las plántulas, es un factor que influye en la uniformidad. Si la red es de lazo abierto la uniformidad se ve afectada por que los insumos no tendrá la oportunidad y la precisión en la cantidad entregada. Una red con doble tubería sería la solución. Pereira, Calejo y otros [26] proponen una metodología que optimice la selección de los diámetros de tubería para riego discontinuo mediante la evaluación de los regímenes de flujo durante un periodo de tiempo, y a través de un método de simulación iterativo de los regímenes generados por la demanda, se pueda calcular la fiabilidad y la equidad del sistema. Pero en la descripción y análisis de los elementos utilizados para elaborar el modelo no se hace referencia a las condiciones del cultivo, tampoco menciona si es necesario definir la topología de la red para la cual se realiza el estudio [26].

Se infiere de los trabajos y referencias consultadas que las estrategias y los dispositivos desarrollados se han orientado fundamentalmente a proponer e implementar herramientas para determinar las cantidades de agua y de nutrientes necesarios para un cultivo. Para hacer llegar estos insumos al cultivo se utilizan programadores y controladores de riego, en este sentido la distribución de una lámina ha sido tarea del personal administrador del fertirriego. Se concluye en esta consulta de trabajos previos, que para el cálculo de una lámina de riego y del fertirriego hay suficientes metodologías, pero no se ha profundizado ni presentado una solución para distribuir una la lámina en un

cultivo cuando el estado fenológico de las plántulas no es uniforme y las características topológicas que presentan las redes hidráulicas para el riego son de lazo abierto.

Los sistemas automáticos basados en el esquema de riego por volumen y/o por tiempo presentan una buena adaptabilidad cuando tienen red doble de tubería, una para riego y otra para fertirriego. La distribución de una lámina de riego con esta característica de la red hidráulica no es problemática. En este caso las condiciones de uniformidad del cultivo (el mismo tipo de plántula, contemporaneidad en la siembra, terrenos sin inclinación) o sectores de este, no influirán en la distribución del riego y el fertirriego. Un escenario como el que se acaba de describir permite que la distribución de los insumos programados a las plántulas sea oportuna, en cantidad y en tiempo (uniformidad en el riego) y quien realice la programación de la distribución de una lámina de riego no tendrá que hacer con anterioridad laboriosos cálculos y planear complejas estrategias para llevar a cabo eficientemente su distribución.

Sin embargo, la adaptabilidad de los sistemas automáticos se pierde cuando los cultivos presentan condiciones problemáticas como son áreas de plántulas con el mismo o diferentes tiempos de siembra, distintos requerimientos de riego y fertirriego, áreas sin plántulas, ordenes de producción imprevistas y una sola red hidráulica de lazo abierto para el riego y el fertirriego. Para esta situación, obtener la lámina y de forma oportuna programar en el controlador/programador de riego su distribución es una labor muy difícil.

Mediante un equipo programador/controlador de riego se realiza la distribución de una lámina al programarle previamente una secuencia de activación de las válvulas del cultivo y del instante en que se debe activar la inyección de fertilizante. Es evidente que se debe disponer con anterioridad de esta información y para obtenerla es necesario evaluar con anterioridad mediante prueba de escritorio, estrategias de secuencias de apertura y cierre de válvulas y del momento de inyección de fertilizante, hasta obtener una buena solución. Son laboriosos los cálculos que se deben realizar, implican un conocimiento de la red hidráulica, de los caudales máximos de la red, de los volúmenes programados de riego y fertirriego, del lugar en el cultivo donde se deben entregar y del tiempo disponible para llevarse a cabo una jornada de riego y fertirriego.

En este trabajo se plasma una solución al problema de la distribución de una lámina de riego con la siguiente situación problemática: un monocultivo ubicado en un terreno sin inclinación que presenta una sola red hidráulica de lazo abierto para el riego, con áreas que tienen plántulas con distintos tiempos de siembra, diferentes requerimientos de riego y fertirriego, y con un sistema automatizado para el control y programación del riego. La optimización respecto a la cantidad a entregar de riego y de fertirriego para un conjunto de plántulas es muy difícil de hacer, debido a la variedad de tiempos en la siembra, al tiempo límite para depositarlo completamente, a la red hidráulica de lazo abierto para conducirlo y porque además de cumplir con los volúmenes y tiempo de entrega, se debe conseguir la mínima pérdida de insumos por desechos en la tubería y en el cultivo.

# Objetivos

## Objetivo general

Proponer una solución mediante algoritmos evolutivos para la distribución más adecuada del riego y fertirriego programado a un grupo de plantas en un cultivo de riego por goteo con mejor comportamiento en cuanto al desperdicio y el tiempo de entrega.

## Objetivos específicos

- Caracterizar un sistema típico de riego por goteo para un monocultivo, con loteo regular en terreno sin pendiente.
- Modelar el sistema típico y determinar la función de aptitud que permita la optimización del sistema.
- Proponer una metodología de optimización aplicando técnicas de computación evolutiva.
- Validar los resultados de la metodología propuesta comparándolos con una programación convencional.

# 1. Marco teórico

## 1.1 Sistemas de riego

El objetivo de los sistemas de riego es poner a disposición de los cultivos el agua necesaria para que cubran sus necesidades. El objetivo del fertirriego es suministrar en el agua, los nutrientes necesarios al cultivo, al ritmo y en la relación adecuada a cada etapa de desarrollo de la planta. Los sistemas de riego pueden clasificarse en tres grandes grupos [27, 28].

### Riego por superficie o gravedad

Con estos métodos de riego se suele mojar la totalidad del terreno y el agua es distribuida realizando en el terreno surcos ver figura 1a, o mediante tablares, una ilustración de este sistema se muestra en la figura 1b. Estos métodos fundamentalmente se utilizan para aplicar agua al cultivo [27]

**Figura 1-1:** Riegos superficiales



<http://ocw.us.es/ingenieria-agroforestal/hidraulica-y-riegos/>

a. Riego por surcos

b. Riego por tablares

### Riego por aspersión

El agua es conducida por tuberías presurizadas. Al llegar a los emisores (aspersores) produce gotas que mojan todo el terreno de forma similar a como lo haría la lluvia. A medida que se haga necesario aumentar el caudal y el alcance o superficie a mojar las

presiones deben ser mayores. Mediante este método pueden ser aplicados fertilizantes o pesticidas en los volúmenes de riego. El suministro de agua se lleva a cabo a través de una sola tubería principal que recorre el cultivo. Los sistemas de aspersión se clasifican de la siguiente forma:

- Estacionarios: a su vez clasificados como fijos, se ilustran en la Figura 1-2a.
- Móviles: se ilustran en Figura 1-2b.

**Figura 1-2:** Riego por aspersión



<http://www.traxco.es/blog/produccion-agricola/produccion-agricola-riego-y-cultivo>. Tomada en noviembre 2015

a. Aspersión fija

b. Aspersión móvil

### **Riego localizado**

En este tipo de riego se humedece sólo la parte del suelo que es próximo a la plántula. El agua llega a las plantas a través de tuberías que contienen microválvulas que funcionan con baja presión. A través de los goteros se realizan pequeñas aportaciones de agua, de manera continua y frecuente. Existen varios sistemas de riego localizado [29,30]: cintas de exudación (ver Figura 1-3a), riego subterráneo (ver Figura 1-3b), microaspersión (ver Figura 1-3c). El más ampliamente usado es el riego por goteo (ver Figura 3-1d) A través del riego localizado, en el agua pueden adicionarse sustancias nutritivas, dándosele el nombre de fertirriego. También es usual también aplicar sustancias plaguicidas.

**Figura 1-3:** Riego localizado



[http://www.poritex\\_04.jpg](http://www.poritex_04.jpg). Tomado en noviembre 2015

a. Cintas de exudación



<https://encrypted-tbn2.gstatic.com/images>. Tomado en noviembre 2015

b. Micro aspersión



<http://cdn.gardena.com> Tomado en noviembre 2015

c. Riego subterráneo



<https://files.wordpress.com/2012/05/01062012601.jpg> Tomado en noviembre 2015

d. Riego por goteo

En el riego localizado se utiliza tuberías en diferentes configuraciones para el transporte y distribución del riego) y/o del fertirriego, las más comunes [27,31] son las siguientes:

#### **Red radial abierta o en lazo abierto**

Es una tubería principal que recorre el cultivo, tiene derivaciones o ramales laterales que cubre todas las zonas. Se ilustra en la Figura 1-4. Este tipo de red se conoce como red de lazo abierto y es la más común y utilizada en cultivos.

#### **Red de tubería en bucle o cerradas**

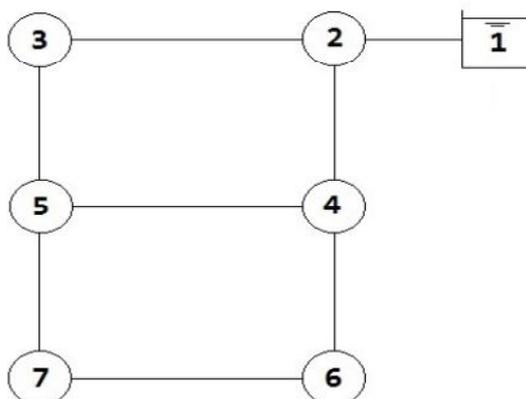
Emplea una sola tubería de distribución pero con laterales que terminan en un bucle cerrado (Figura 1-5), es muy utilizada para regular la presión de la red.

**Figura 1-4:** Tubería abierta o en lazo abierto



Fuente: <http://ingemecanica.com/tutorialsemanal/objetos/figutut207/fig1tut207.jpg>. Tomado en noviembre 2015

**Figura 1-5:** Red de tubería en bucle



Fuente: Alperovits and Shamir\_

### Red de tubería en paralelo

Se utilizan dos tuberías principales, no es muy común debido a que los costos iniciales de inversión son altos, ver ilustración de este método en la Figura 1-6. Una tubería se emplea para riego y la otra para fertirriego. Mediante este sistema se puede tener agua disponible en cualquier instante y lugar del cultivo, sin intervenir con el fertirriego, permitiendo realizar otras actividades como refresques, preparación de camas, limpieza etc. Con la red de dos tuberías, se gestiona más eficientemente la aplicación y distribución del fertirriego en cuanto a la oportunidad en tiempo y en cantidad. Sin embargo, el desperdicio podría ser alto. Si después de finalizar una jornada de fertirriego el fertilizante permanece mucho tiempo en la tubería, se genera precipitados y gránulos,

entonces, cuando nuevamente se empuje fertilizante las cintas de goteo se obstruirán, incrementando las tareas de mantenimiento. Esta situación hace que el fertilizante se deba desalojar de la tubería al finalizar el fertirriego, a un lugar donde normalmente no es aprovechado ocasionando desperdicio. Pero si se quiere depositar en el cultivo para que no se desperdicie totalmente se debe empujar con agua y cuidar que se distribuya sin causar perjuicio. Otra situación desventajosa que puede darse, es si la fórmula de fertilizante cambia y no es compatible con la anterior. Igualmente se deberá proceder a desalojar el fertilizante donde más convenga.

Este trabajo se centrará en la distribución del riego y del fertirriego por una red hidráulica de lazo abierto para un cultivo en un terreno plano que utilice el riego por goteo.

**Figura 1-6:** Red de dos tuberías: riego y fertirriego



## 1.2 Componentes de un sistema de riego por goteo

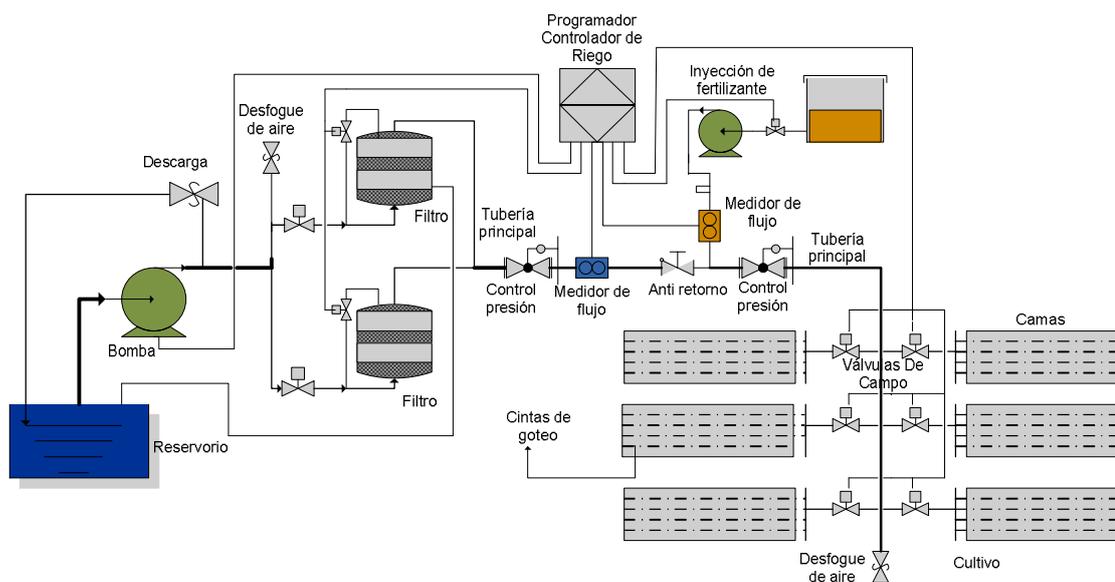
El esquema de un sistema de riego de este tipo, puede apreciarse en la Figura 1-7. Los subsistemas que lo conforman son los siguientes [29, 31].

### **Sistema de bombeo de agua o cabezal de bombeo**

Mediante este sistema se envía el agua al cultivo a la presión y calidad necesaria para garantizar la operación de los goteros y la no obstrucción por partículas. Este subsistema consta de los siguientes componentes [29,31]:

- Una bomba centrífuga la cual empuja un caudal  $Q$  a una presión  $P$  (normalmente 70 PSI). El volumen de agua es obtenido de un reservorio o de una fuente corriente (rio).
- Sistema de filtros, para eliminar las partículas solidas o materia orgánica que pueden obstruir las microválvulas o goteros.
- Sistema de válvulas de regulación y de control: son válvulas con las siguientes funciones: válvula de desfogue del aire contenido en la tubería, válvula de regulación de la presión en la línea de conducción, válvulas para el control de retrolavado de los filtros, válvula antiretorno de flujo.
- Hidrómetro: mide el caudal de agua enviada hacia el cultivo.

**Figura 1-7:** Esquema de un cultivo con un sistema de riego por goteo



### Sistema de inyección y dosificación de fertilizante

A través de este subsistema se inyecta a la tubería principal y se mezcla con el agua los nutrientes y/o productos fitosanitarios para las plántulas del cultivo. Está ubicado en la caseta del sistema de bombeo o en otro punto crítico que puede ser un punto equidistante del cultivo. Lo conforman los siguientes dispositivos [27, 29, 52]:

- Tanques almacenadores de fertilizante: contienen los productos que serán inyectados a la red principal de agua.
- Bomba centrífuga: succiona agua de la tubería principal y eleva su presión.
- Dosificador venturi: a través de él y la bomba se succiona el fertilizante de un contenedor e inyecta a la red principal de agua.
- Medidor de caudal o flujo: Es un caudalímetro que cada vez que pasa un cierto volumen de fertilizante envía un pulso o tren de pulsos al controlador de fertirriego.
- Válvula de cierre y apertura rápida para control de salida de fertilizante desde el recipiente contenedor.

### **Controlador/programador de riego**

Realiza las siguientes funciones [32, 33, 52]:

- Recibir la señales de los instrumentos de campo, como son el hidrómetro (agua) y del caudalímetro (Fertilizante), electroconductividad y pH
- Enviar la señal para la apertura y cierre de la válvula controladora de fertilizante según la relación programada fertilizante respecto agua.
- Permitir y alojar la programación de volúmenes, tiempos, ciclos de riego y fertirriego.
- Enviar la señal para la apertura y cierre de las válvulas de campo de acuerdo a la programación de caudal o tiempo asignado.
- Monitor de pH y conductividad. Realiza la función de monitorear la solubilidad del fertilizante con el agua. Se mide antes del que la red de fertilización conecte a la red principal.
- Permite y almacena la programación de retrolavado para los filtros

### **Válvulas Volumétricas o de riego**

Son un elemento final de control [52], las activa durante un determinado tiempo una orden del controlador/programador de riego. Su función es que al activarse deja pasar hacia las camas que contienen las plántulas, el insumo que esté presente en su entrada.

### **Funcionamiento del sistema de inyección de fertilizante**

En la Figura 1-7 puede observarse uno de los métodos de inyección de fertilizante a la tubería principal. Se realiza colocando en paralelo con la tubería principal una bomba seguida de un inyector venturi [52]. La función de la bomba es elevar la presión del agua tomada de la red, con el fin de producir una diferencia de presiones en el inyector venturi, generándose una presión negativa o de succión en el centro del inyector ocasionando que se aspire el fertilizante del recipiente. El fertilizante se mezcla con el agua que impulsa la bomba a una presión mayor que la de la tubería principal y de esta forma entra a la red. En el tramo de tubería entre el venturi y el depósito de fertilizante se colocan el medidor de flujo de fertilizante y una válvula que controla el paso de fertilizante. Cuando la válvula está abierta, el medidor de flujo envía una señal (un pulso o varios pulsos) al controlador / programador de riego, pasar un determinado caudal de fertilizante. La tasa de fertilizante en el agua de riego (volumen / tiempo) o la concentración (ppm ò masa / volumen), es fijada en el controlador. El controlador mediante la información que envía el hidrómetro y el medidor de flujo de fertilizante, calcula cuando debe enviar la señal de apertura y de cierre de la válvula de fertilizante [52]. Es decir, conocidas estas proporciones, el fertilizante, será inyectado a la red principal cada vez que pase cierto volumen de agua. Puede inferirse entonces que la bomba impulsora de agua siempre estará en funcionamiento y cuando solo se necesite enviar fertirriego, la válvula que deja pasar de fertilizante se activara. De acuerdo a la descripción anterior, estos insumos se desplazan por la tubería en forma de “lotes” discretos. “lotes” de riego (0) o “lote” de fertirriego (1).

### **1.3 Estrategia empírica para la distribución de una lámina de riego**

La programación de la distribución de una lámina de riego-fertirriego, en un sistema automatizado depende de las situaciones problemáticas ya definidas para este trabajo. Lo anterior implica plantear diferentes estrategias empíricas que no tienen un procedimiento estandarizado. La efectividad de la estrategia que se implemente depende de la experiencia e imaginación de quien la lleve a cabo y para implementarla muy posiblemente debe de realizar modificaciones sobre la marcha y de carácter temporal al sistema hidráulico. Por ejemplo, prescindir de sectores del cultivo realizando rutas

alternas (*bypass*), y/o alternar el controlador de modo automático a manual. La automatización en estas condiciones es ineficiente y termina siendo un fracaso y ocasionando altos costos económicos y de tiempo en la actividad del riego y fertirriego.

A continuación se describe una estrategia para realizar el envío de una lámina de riego utilizando válvulas para desalojar a un sumidero el desperdicio de riego y fertirriego generado.

Para comenzar a planear cualquier estrategia para la distribución de una lámina de riego en cultivo que tenga un sistema automatizado, mínimamente debe cumplirse las siguientes condiciones:

- Garantizar la presión necesaria en la red hidráulica para que los caudales de los goteros estén en el rango de trabajo.
- Conocer la lámina de riego que se va a distribuir.
- La tubería debe estar llena de solo agua (riego)
- Solo una válvula de campo estará activa al tiempo.
- La forma como se entrega el riego y fertirriego es mediante una secuencia de apertura y cierre de válvulas, cada válvula cubre un área sembrada de plántulas. Esta secuencia de apertura y cierre de las válvulas ira indicando la ruta que sigue el riego o fertirriego por la red hidráulica.
- La distribución del riego y fertirriego de una jornada, se realiza en un número de ciclos, que al finalizar, deben haber entregado toda la lámina programada. Los ciclos es la frecuencia de entrega de una cantidad mínima o máxima de riego o fertirriego, que ha sido previamente estudiada para que la absorción del alimento por las plántulas sea eficientemente y además la necesidad de mantener los suelos en cierta condición, para evitar la acumulación de sales alrededor del bulbo [28, 29].
- Conocer la topología de la red hidráulica y sus parámetros: diámetros interior y exterior, longitudes a bifurcaciones y a cada válvula desde el punto de inyección de fertilizante, volumen que aloja por metro la tubería.
- Conocer los caudales máximos de las válvulas de riego,
- Definir el sistema de inyección utilizado por la unidad de fertilización.
- Al comienzo y final de una jornada de riego, se debe realizar un acondicionamiento de campo con un volumen de agua asignado a cada válvula para ello.

La estrategia a describir puede llevarse a cabo utilizando el agua que contiene la tubería para hacer acondicionamiento de campo (volumen de agua para garantizar un porcentaje de humedad en el suelo y facilitar la absorción de nutrientes) [1, 28, 29], activando las válvulas que van a regar para que la desalojen y empujando (inyectando) fertilizante al mismo tiempo hacia las mismas válvulas, o mediante las válvulas desperdicio desalojar del agua presente en la tubería e inyectar fertilizante hasta que esta quede llena.

Para utilizar el agua en la tubería como acondicionamiento de campo, primero se debe calcular el volumen de agua presente en la tubería hasta la válvula más lejana que va a regar fertirriego. Luego, como se conoce el volumen de agua asignado a cada válvula para el acondicionamiento de campo, se calcula el tiempo total de desalojo de este caudal. Este tiempo divide entre las válvulas más lejanas que desalojaran fertirriego hacia las plántulas y es el tiempo durante el cual estarán activas. La ejecución se lleva a cabo de la siguiente forma: de las válvulas seleccionadas se activa, la menos lejana, al mismo tiempo se comienza a enviar (inyectar) fertirriego. Al terminar de desalojar el agua de acondicionamiento a través de estas válvulas, el fertilizante estará en toda la tubería desde la inyección de fertilizante hasta la válvula más lejana. Con este estado que presenta el sistema, se está en condiciones de comenzar con el fertirriego, activando las válvulas según la programación de la lámina y se comienza el fertirriego activando la válvula más cercana al punto de inyección de fertilizante. Luego de terminado el fertirriego, para dejar la tubería llena únicamente de agua, se procede a desalojar el fertilizante que quede en la tubería a través de las válvulas de desperdicio.

Esta estrategia tiene el inconveniente de depender de la forma en que está distribuida la red hidráulica. Esto hace que pueda quedar mucho fertilizante en la tubería al final del fertirriego, este no se debe evacuar al cultivo por que traería sobredosis a las plántulas, ocasionando daño por toxicidad, efectos salinos en la tierra y por consiguiente disminución de la absorción del fertilizante y finalmente baja producción.

## 1.4 Optimización

En las aplicaciones de ingeniería y en la industria, siempre se trata de optimizar, por ejemplo maximizar la eficiencia en la producción, el beneficio, reducir a un mínimo el tiempo utilizado en un proceso, los costos de producción, la energía utilizada, etc.

Para implementar computacionalmente un proceso de optimización, deben de tenerse en cuenta los siguientes elementos: un modelo de lo que se quiere optimizar, un algoritmo de optimización y un procesador numérico que realice la tarea de cálculo. La mayoría de modelos tienen factores que los hacen muy complicados, situación que ocasiona en muchos casos, altos costos computacionales en tiempo. Pero ante un modelo apropiado los costes computacionales estarán determinados por los algoritmos de optimización empleados y la solución numérica utilizada para la simulación [30].

La incertidumbre en los sistemas reales siempre está presente, lo que ocasiona que las perturbaciones estocásticas que los afectan, tengan una incidencia negativa en la respuesta a los estímulos que se les aplica. Por lo tanto las soluciones encontradas a su operación, control y predicción, deben de ser óptimas pero también robustas, esto quiere decir que se encuentre dentro de lo que se desea, pero que también se responda acertadamente a situaciones inesperadas.

Los algoritmos de optimización son las herramientas y técnicas para resolver problemas de optimización con la intención de encontrar su óptimo, aunque este no siempre se pueda obtener. Soluciones subóptimas o buenas soluciones robustas son menudo la opción en tales casos [36].

Los problemas de optimización pueden formularse de muchas maneras, por ejemplo empleando técnicas de máxima verosimilitud por ejemplo mínimos cuadrados o método simplex [19], que corresponden a sistema lineales. Los sistemas no lineales pueden escribirse como:

Minimizar

$$(\mathbf{x}), (i = 1, 2, \dots, M) \tag{1.1}$$

Sujeto a las siguientes restricciones

$$h_j(x) = 0, (j = 1, 2, \dots, J) \quad (1.2)$$

$$g_k(x) \leq 0, (k = 1, 2, \dots, K) \quad (1.3)$$

Donde  $f_i$ ,  $h_j$  y  $g_k$  en general son funciones no lineales,  $x$  corresponde a las variables o vector de diseño, estas pueden ser de característica continua o discreta,  $f_i$  es llamada la función objetivo o de costo. Si el problema de interés es de maximización, la función objetivo puede ser escrita reemplazando  $f_i(x)$  por  $-f_i(x)$ . Si  $M > 1$  la optimización es multiobjetivo [31].

Un algoritmo es un procedimiento que genera unas salidas determinadas a partir de un conjunto de entradas. Un algoritmo de optimización genera la solución  $x^{t+1}$  a un problema dado  $S$ , a partir de la solución  $x^t$ , en  $n$  iteraciones o en un tiempo  $t$ .

Las soluciones generadas pueden describirse de la siguiente forma.

$$x^{t+1} = \mathbf{A}(x^t, \mathbf{p}(t)) \quad (1.4)$$

Donde  $A$  es un vector de soluciones de  $d$ -dimensión donde el algoritmo realiza una correspondencia no lineal a partir de una solución  $x^t$  a una nueva solución  $x^{t+1}$ . El comportamiento o sintonización del algoritmo estará dado por  $k$  parámetros que pueden depender del tiempo, es decir,

$$\mathbf{p}(t) = (p_1, \dots, p_k) \quad (1.5)$$

Mediante un conjunto de reglas o ecuaciones matemáticas llevadas a un determinado número de iteraciones las soluciones pasan por estados que evolucionan, dando lugar o alcanzando estados autorganizados (concepto de atractor) [33], que corresponderán a algunos óptimos en el espacio de búsqueda, es decir el sistema converge a un óptimo.

Para encontrar la solución óptima  $x_*$  a un problema de optimización dado  $S$ , con un número infinito de estados, es frecuente seleccionar algunos estados  $\varphi$  deseados de todos los estados  $\psi$ , de acuerdo con algún criterio predefinido  $D$  de parada del algoritmo utilizado. La selección de los estados del sistema en el espacio de estudio se lleva a cabo mediante la ejecución del algoritmo de optimización [32, 33].

De otro modo.

---

$$S(\boldsymbol{\psi}) \xrightarrow{A(t)} S(\boldsymbol{\varphi}, \mathbf{x}_*) \quad (1.6)$$

Un algoritmo que no necesite estados iniciales del problema, es probable que sea más versátil para tratar muchos tipos de problemas. Sin embargo con el fin de ser eficaz en la selección de los estados o soluciones de manera eficiente, la información que va generando el proceso de búsqueda debe utilizarse para mejorar este mismo proceso. De hecho el mecanismo de selección más simple y a la vez más utilizado para elegir o mantener la mejor solución encontrada al momento, es la estrategia evolutiva de la "supervivencia del más apto" [30].

Estos algoritmos que se construyen con base a los teoremas del no-free-lunch [34], y expresan que el rendimiento de un algoritmo depende del tipo de problema  $S$  que resuelve y además, si el óptimo es alcanzable en un determinado número de iteraciones, dependerá del algoritmo utilizado.

Indudablemente un algoritmo de optimización eficiente es muy importante para asegurar que las soluciones óptimas sean alcanzables. Son muchos los algoritmos de optimización en la literatura pero ningún algoritmo a demostrado ser adecuado para todos los problemas [34].

Un algoritmo de optimización se llama determinístico si funciona sin ningún carácter aleatorio y siempre convergerá a la misma solución final si se parte del mismo punto inicial. Algoritmos de este tipo son Ascenso a la colina y el simplex cuesta abajo [34, 35]. Por otro lado, si hay algo de aleatoriedad en el algoritmo, este suele llegar a un punto diferente cada vez que se ejecute, a pesar de que empezamos con el mismo punto inicial. Los algoritmos metaheurísticos basados en poblaciones y de inspiración en la naturaleza o bioinspirados, utilizan la aleatoriedad cada que se realiza su ejecución y en el interior de los diferentes componentes del algoritmo. Estos son muy diversos y los más utilizados para la optimización son: los algoritmos genéticos, el recocido simulado, algoritmo de colonia de abejas y de hormigas, optimización por enjambre de partículas, la búsqueda armónica, algoritmo de la luciérnaga y la búsqueda cuckoo [35, 36].

Para dar una solución al problema de la distribución de riego y el fertirriego óptimamente, y de manera que presente mayores ventajas que la realizada intuitiva y empíricamente por un programador humano, se utilizan en este trabajo los algoritmos evolutivos basados en teorías de origen biológico, como son los Algoritmos Genéticos (GA), Optimización por Enjambre de Partículas (PSO) y la búsqueda cuckoo (CS). Estos algoritmos se han aplicado a la solución de problemas de ruteo en diferentes aplicaciones de la ingeniería. Por ejemplo, para hallar la ruta óptima en redes de telecomunicaciones que transmiten o distribuyen datos como son las redes de telefonía e Internet desde los remitentes hasta los destinatarios [7]. Los algoritmos evolutivos son extremadamente aptos, ya que han encontrado una enorme variedad de aplicaciones exitosas en diversos campos. La solución a la distribución del riego y fertirriego es un tipo de problema para el que la inteligencia computacional puede ofrecer una solución adecuada.

## **1.5 Metaheurísticas**

Un algoritmo optimización debe realizar un procedimiento de búsqueda que encuentre soluciones que puedan considerarse buenas y se logren en un tiempo razonable. Para un algoritmo que utilice una búsqueda estocástica se utiliza dos términos que son casi indistintos, uno es la heurística y el otro la metaheurística. Los dos hacen referencia a asignaciones al azar y búsqueda local. La heurística es un modo por ensayo y error para producir soluciones aceptables para un problema complejo en un tiempo razonablemente práctico [8].

La aleatorización proporciona una buena manera de alejarse de búsqueda local a la búsqueda en la escala global. Por lo tanto, casi todos los algoritmos metaheurísticos tienen la intención de ser adecuados para la optimización global. En resumen, una metaheurística es una estrategia de gran refinamiento que usa diferentes métodos heurísticos explorando en espacios de gran tamaño que puedan contener soluciones viables, es decir, óptimas o cuasi óptimas. Pero es claro que las técnicas metaheurísticas son algoritmos no exactos [37].

Dos componentes importantes de cualquier algoritmo metaheurístico son: la intensificación y la diversificación. La diversificación es generar diversas soluciones a

partir de la exploración de un espacio de búsqueda global, mientras que la intensificación busca alrededor de una región local mediante la explotación de la información para obtener una buena solución que se encuentra en esta región. La combinación de estos dos componentes por lo general asegura que la optimalidad global es alcanzable.

Los algoritmos metaheurísticos se clasifican de diversas formas. Las más empleadas son las que se basan en si la técnica utiliza un único punto del espacio de búsqueda o lo hace sobre una población. La primera se conoce como metaheurísticas basadas en trayectorias y las segundas metaheurísticas basadas en población. Estas últimas son fundamentalmente de origen en la naturaleza, y representan una metáfora biológica (algoritmos bioinspirados). En esta técnica la población está conformada por un conjunto de individuos los que a su vez son posibles soluciones. El resultado y eficiencia dependerá de la forma como se utiliza la población en cada iteración [9].

Los algoritmos evolutivos son algoritmos basados en poblaciones, donde los individuos tienen la capacidad de evolucionar y de adaptarse a los cambios del entorno. La forma de implementar este algoritmo es generando una población de forma aleatoria, cada individuo tiene un peso o bondad de acuerdo a una función que se conoce como función de aptitud "fitness" [10]. Esta es la información que el algoritmo utiliza para llevar a cabo la búsqueda. Hay fases en este procedimiento que deben de llevarse a cabo: la fase de selección de individuos, la explotación o búsqueda intensiva de una región local, y la fase de reproducción de individuos y búsqueda de nuevas regiones.

Los algoritmos genéticos GA [11], constituyen una de las técnicas más reconocidas, luego a esta estrategia le siguió en importancia el Algoritmos Basados en enjambres de Partículas o Particle Swarm Optimization (PSO) [12] y más recientemente el algoritmo Cuckoo Sears (CS) [13].

Las metaheurísticas basadas en trayectoria parten de un punto inicial y van actualizando la solución presente mediante la exploración del vecindario, formando una trayectoria. La búsqueda finaliza cuando se alcanza un número máximo de iteraciones, se encuentra una solución con una calidad aceptable, o se detecta un estancamiento del proceso.

A esta estrategia pertenecen los algoritmos como es el enfriamiento Simulado ES [38]. La Búsqueda Tabú (BT), una de las estrategias más utilizadas en optimización [39].

Las metaheurísticas poblacionales que se utilizaron en este trabajo son: Algoritmos Genéticos (*Genetic Algorithms (GA)*) y otros más recientes como Optimización de Enjambre de Partículas (*Particle Swarm Optimization (PSO)*) y la Búsqueda del *Cuckoo (Cuckoo Search (CS))*. Son varias las razones por las que estas metaheurísticas evolutivas se han seleccionado. La primera es que los algoritmos no necesitan de conocer la naturaleza del problema para generar soluciones [30], una segunda es que la función de aptitud construida se puede utilizar sin modificaciones en los tres algoritmos GA, PSO y CS. Una tercera es que son algoritmos ya consolidados que han demostrado entregar buenas soluciones a problemas de diversa índole [7, 36].

### **1.5.1 Algoritmo genético GA**

Los algoritmos genéticos (GA), fueron inventados en 1975 por John Holland, de la Universidad de Michigan. Los GA son, simplificando, algoritmos de optimización, es decir, tratan de encontrar la mejor solución a un problema dado entre un conjunto de soluciones posibles. Los mecanismos de los que se valen los AG para llevar a cabo esa búsqueda pueden verse como una metáfora de los procesos de evolución biológica.

Los GA son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los GA son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas [5].

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un

gran número de descendientes. Por el contrario individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagaran en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes "superindividuos", cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los GA son una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el AG encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

En el caso de que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al GA, tanto en rapidez como en eficacia. El gran campo de aplicación de los GA se relaciona con aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas hibridándolas con los GA.

Una secuencia del funcionamiento general de un algoritmo genético se muestra a continuación:

Algoritmo Genético:

1. Comienzo
2. Generar una población inicial.
3. Calcular la aptitud de los individuos de la población generada
4. Iterar hasta un criterio de parada.
4. Evaluar cada individuo de la población.
5. Seleccionar los progenitores o mejores individuos de la población evaluada.
6. Aplicar el operador de cruce a los individuos seleccionados.
7. Aplicar elitismo y calcular la aptitud de los nuevos individuos generados

8. Incluir la nueva descendencia para formar una nueva generación.
9. Aplicar nuevamente los pasos 5 a 8
10. Fin

Una característica importante es que los GA son intrínsecamente paralelos. La mayoría de los otros algoritmos son en serie y sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, y si la solución que descubren resulta subóptima, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo. Debido a que los GA tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez. Si un camino resulta ser un callejón sin salida, pueden eliminarlo fácilmente y continuar el trabajo con otros más prometedores, dándoles una mayor probabilidad en cada ejecución de encontrar la solución [5].

### **1.5.2 Optimización por enjambre de partículas PSO**

El algoritmo *PSO* es una técnica de optimización inteligente y se define como un algoritmo estocástico de optimización basado en población. Aquí los individuos usan su propia experiencia y en el grado de confianza en sus vecinos, para desplazarse en el espacio de búsqueda. Es básicamente paralelo. Muchos algoritmos tradicionales aplican secuencialmente y pueden explorar el espacio de solución únicamente en una dirección a la vez. *PSO* aumenta la probabilidad de encontrar la solución óptima ya que explora el espacio de la solución en múltiples direcciones simultáneamente. Si un camino no conduce al óptimo, el algoritmo sencillamente elimina ese camino y continúa buscando mejores soluciones.

Debido al paralelismo, los algoritmos inteligentes de optimización son adecuados para problemas con gran número de variables, donde el espacio de la solución es de gran tamaño para realizar una búsqueda exhaustiva en un tiempo razonable. Otra fortaleza de los algoritmos de optimización inteligente es su buen desempeño en problemas cuyo espacio de solución presenta múltiples mínimos locales.

Este método, desarrollado por James Kennedy y Russell Eberhart en 1995 [12], se basa en los movimientos de grandes grupos de insectos o de bandadas de peces. En estos

grupos, se estudian los movimientos de cada individuo en busca de un objetivo con la ventaja de que existe siempre alguna forma de comunicación entre los individuos. De esta forma, cada uno de ellos se mueve estimulado por su propia costumbre de los mejores lugares en los que ha estado anteriormente. O bien, se mueve hacia las mejores zonas reveladas por algún otro miembro del enjambre. También está presente una componente de inercia del movimiento anterior que amortigüe las velocidades de las partículas.

Cada posible solución del problema que se desea resolver estará constituida por un conjunto de valores de las variables a optimizar. En este algoritmo, cada solución se asocia con las coordenadas de un punto o partícula y, mediante la función objetivo, se calculará la bondad de cada solución. El algoritmo comenzará con una población de partículas que irán moviéndose en busca de mejores soluciones hasta llegar a la óptima. El procedimiento de implementación del algoritmo *PSO* se muestra en el diagrama de flujo de la Figura 1-8.

Cada partícula comunica las buenas posiciones a las demás y dinámicamente ajustan su propia posición y su velocidad con base en las buenas posiciones. La velocidad se ajusta con el comportamiento histórico de las partículas. De esta forma, las partículas tienden a dirigirse hacia un mejor espacio de búsqueda en el proceso de minimización de la función objetivo [12]. Este procedimiento de búsqueda se describe por la ecuación (1.7).

$$V^{k+1} = wv_i^k + c_1rand_1(pbest_i - x_i^k) + c_2rand_2(gbest - x_i^k) \quad (1.7)$$

$$i_i^{k+1} = x_i^k + v_i^{k+1}$$

En (1.7)  $c_1$  y  $c_2$  son constantes positivas, definidas como coeficientes de aceleración;  $w$  es el factor inercial;  $rand_1$  y  $rand_2$  son dos números aleatorios (con distribución de probabilidad uniforme) en el rango  $[0,1]$ ;  $x_i$  representa la  $i^{ésima}$  partícula y  $pbest_i$  la mejor posición previa de  $x_i$ ;  $gbest$  es la posición de la mejor partícula de toda la población; y  $v_i$  es la razón de cambio de la posición (velocidad) de la partícula  $x_i$ . Los cambios de velocidad en (1.7) se componen de tres partes: *Confianza personal*, *Confianza social*, *Inercia*. De esta forma se obtiene una velocidad que tiende a acercar la partícula a  $pbest$  y a  $gbest$ .

**Figura 1-8:** Diagrama de flujo algoritmo *PSO* [40, 41].



Los tres componentes del movimiento de cada partícula se definen de la siguiente manera [8,29]:

- *Confianza personal.* Cada una de las partículas recordará el mejor punto por el que ha pasado hasta el momento y se establecerá una componente de su velocidad como la diferencia entre el mejor punto personal y el punto actual.
- *Confianza social.* Todas las partículas tendrán conocimiento del mejor punto que haya alcanzado cualquiera de las demás partículas del enjambre. Por tanto, la otra componente de la velocidad será la diferencia entre este mejor punto global y la situación actual de cada partícula.
- *Inercia.* Una componente de inercia del movimiento anterior de la partícula.

Considerando estas tres componentes, cada partícula se moverá de un punto a otro nuevo y la variación de estos factores puede balancear la búsqueda global y local durante el proceso de optimización.

Las constantes de aceleración  $c_1$  y  $c_2$  sirven para dos propósitos en el algoritmo. En el primero, ellas controlan la influencia relativa hacia  $gbest$  y  $pbest_i$  respectivamente, en el segundo, los dos coeficientes de aceleración cercanos a cero (0) producirán una búsqueda fina en una región, mientras coeficientes cercanos a uno (1) permitirán a la partícula la posibilidad de sobrepasar al  $gbest$  y al  $pbest_i$ , resultando una búsqueda amplia.

### 1.5.3 Búsqueda *cuckoo* (CS)

La búsqueda cuckoo (CS), es un algoritmo metaheurístico inspirado en la naturaleza, como una técnica de optimización inteligente. Desarrollado en 2009 por Xin-Ella Yang y Suash Deb [37] se basa la actitud parasita y agresiva que presentan algunas especies de aves cucos como las Ani y la Guira para su reproducción. La cual consiste en poner un huevo en los nidos de otras aves en el momento en que estas lo han hecho, con el fin de que estas sean las que lo incuben. La Tapera cucú hembra, ha evolucionado a lograr imitar el color y la forma del huevo al de algunas especies seleccionadas, para disminuir la probabilidad de que el huevo sea descubierto, pues si la ave hospedera lo descubre abandona definitivamente el nido. En general, el huevo huésped eclosiona primero y el polluelo aun ciego, y por instinto desaloja del nido al huevo del ave residente aumentando la participación en comida que proporciona el ave residente. Si la eclosión es del huevo residente, y el huésped también lo logra. Comienza la lucha por la sobrevivencia, el polluelo cucú ha logrado imitar el llamado del polluelo huésped para demandar más alimento [42].

El algoritmo de búsqueda *cuckoo* usa las siguientes tres reglas ideales:

- Cada cuco elige un nido al azar pone un huevo a la vez;
- Los nidos con huevos (soluciones) de alta calidad se llevarán a las generaciones siguientes;
- El número de nidos es fijo, y el huevo puesto por un cuco tiene una probabilidad de ser descubierto por el pájaro anfitrión con una probabilidad de  $p_a \in (0,1)$ . En ese caso, el pájaro anfitrión puede o bien deshacerse del huevo, o simplemente abandonar el nido y construir uno nuevo. Este valor de probabilidad es una fracción

de los nidos que serán reemplazados por unos nuevos con nuevas soluciones que son tomadas al azar.

En el algoritmo CS, un huevo representa una solución, y cada nido podrá contener solo una. Una solución no muy buena es reemplazada, por otra potencialmente mejor. Sin embargo, un algoritmo más complejo puede desarrollarse colocando varias soluciones en un nido.

La exploración de las soluciones en la CS, se realiza mediante un paseo aleatorio conocido como los vuelos de Lévy. Esta exploración consta de dos etapas. Una que selecciona en un plano bidimensional, la dirección de la búsqueda, para la cual utiliza una distribución aleatoria uniforme. La otra etapa es determinar el paso o salto realizado para llegar a una solución candidata. Para determinar este salto hay algunas formas de realizarse, pero desde el punto de vista de la implementación, la más utilizada es el algoritmo de Mantegna [43] por su sencillez y eficiencia, (Ver ecuación (8.1)).

$$s = \frac{u}{|v|^{1/\beta}} \quad \text{Donde:} \quad 1 < \beta < 2 \quad (8.1)$$

$u$  y  $v$  son dos distribuciones normales con  $u = N(0, \sigma_u^2)$  y  $v = N(0, \sigma_v^2)$ .

El cálculo de la varianza es muy importante para esta distribución y se obtiene de la siguiente forma, (ver ecuación (9.1))

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \text{se} \pi(\pi\beta/2)}{\Gamma[(1+\beta)/2] \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_u = 1 \quad (9.1)$$

El salto  $s$  debe tener un valor inicial  $s_0$ , tomando un valor en el intervalo de 0.1 a 1.

Para los vuelos de Lévy y para la implementación del algoritmo Mantegna [31], es determinante dar un valor de la varianza de los vuelos, debido a que después de un buen número de saltos, estos deben tender a una distribución estable [30, 37].

La búsqueda de *cuckoo* tiene dos posibilidades: la búsqueda local y búsqueda global, controlados por una probabilidad de conmutación o descubrimiento, mediante el parámetro  $p_a$ . La búsqueda local es muy intensa utilizando una fracción de tiempo  $p_a$  de las interacciones, mientras que la búsqueda global tarda aproximadamente  $1-p_a$  del tiempo total de búsqueda. Esto implica que para la optimización, CS puede explorar el espacio de búsqueda de manera más eficiente en la escala global, y no quedarse estancado óptimos locales demostrando tener propiedades de convergencia globales, a diferencia de los algoritmos de proceso de Gauss estándar [37]. A continuación se describe el Pseudocódigo del algoritmo CS [13].

1. Comienzo
2. Definir la función objetivo
3. Definir la población inicial con el numero de nidos  $n$
4. *Mientras* no se cumpla el criterio de parada o no se cumpla el máximo número de iteraciones
5. Obtenga una solución aleatoriamente mediante un vuelo de Lévy.
6. Evaluar la calidad o aptitud de esta solución  $F_i$
7. Seleccione un nido  $F_j$  aleatoriamente,
8. *Si*  $F_i > F_j$
9.     reemplazar por una nueva solución
10. *Fin Si*
11. Abandonar una fracción  $P_a$  de nidos
12.     reemplazarlos por unos nuevos.
12. Guardar el nido con la mejor solución.
13. Clasificar las soluciones y encontrar la mejor vigente.
14. *Fin Mientras*
15. Procesar resultados y mostrar.
16. Fin

Fuente (Yang, Deb)

## 1.6 Grafos y dígrafos

Desde mediados del siglo XVII se han propuesto soluciones de problemas utilizando representaciones de gráficas como fue la aplicada al problema de los siete puentes de Königsberg [44]. A partir de esta solución muchos problemas de ingeniería pudieron plantearse mediante una sucesión de puntos y líneas (gráficas) para encontrarles una solución. Algunas áreas de interés que las han utilizado son: en robótica para diseño y cálculo de rutas óptimas, en redes de datos [15], en ciencias de la computación [16] y en

redes hidráulicas [14]. Es así como los grafos una herramienta muy poderosa para definir sistemas expertos utilizando sus teorías.

La idea comienza con una representación geométrica simple, el dígrafo. Un dígrafo es un grafo dirigido y está definido en la teoría de grafos. Hay dos elementos básicos en los grafos, los nodos o vértices (puntos) y las aristas (líneas). Estos se utilizan para representar un conjunto de variables proposicionales (los nodos) y una relación de dependencia entre ellas (las aristas) [45].

### 1.6.1 Grafos dirigidos y no dirigidos

Según el orden o la relación que tengan con el objeto, las aristas pueden ser dirigidas o no dirigidas, y dos nodos no necesariamente estarán unidos por una sola arista o una arista puede unir un nodo consigo mismo. Un Grafo  $G$  dirigido o no dirigido, consiste en dos conjuntos: un conjunto  $V$  de vértices o nodos y un conjunto  $E$  de aristas. En un grafo  $G$  no dirigido cada arista  $e \in E$  está asociada a un par de vértices  $u$  y  $w$ . Se escribe:

$$e = (u, w) \text{ o } e = (w, u), \quad (10.1)$$

Donde  $e$  denota una arista única entre  $u$  y  $w$  en una gráfica no dirigida pero no denota un par ordenado. En un grafo  $G$  dirigido cada arista  $e \in E$  está asociada a un par ordenado de vértices  $u$  y  $w$  se escribe:

$$e = (u, w), \quad (11.1)$$

que denota una arista única de  $u$  a  $w$ .

Un grafo dirigido o no, con vértices  $V$  y aristas  $E$  se escribe:

$$G = (V, E). \quad (12.1)$$

Se supone que los conjuntos  $V$  y  $E$  son finitos y  $V$  no es vacío. Para los grafos dirigidos  $G_d$  y de los no dirigidos  $G_{nd}$  el conjunto de vértices  $V$  es:

$$V = \{v_1, v_2, v_3, v_4\}, \quad (13.1)$$

y el conjunto de aristas  $E$  es:

$$E = \{e_1, e_2, e_3, e_4\}, \quad (14.1)$$

El grafo dirigido  $G_d$  (Figura 1-9a) se expresa de la siguiente forma:

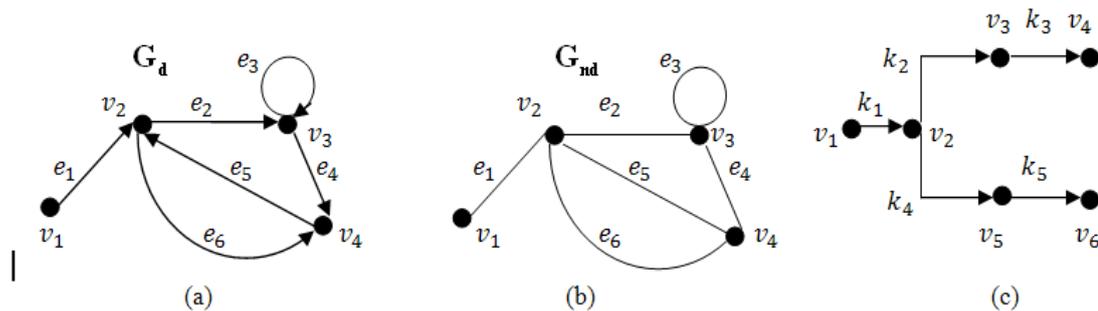
$$G_d(V \mathfrak{R} E) = \{e_1(v_1, v_2), e_2(v_2, v_3), e_3(v_3, v_3), e_4(v_3, v_4), e_5(v_4, v_2), e_6(v_2, v_4)\} \quad (15.1)$$

El grafo no dirigido  $G_{nd}$  (Figura. 1-9b) se expresa:

$$G_{nd}(V \mathfrak{R} E) = \{e_1(v_2, v_1), e_2(v_3, v_2), e_3(v_3, v_3), e_4(v_3, v_4), e_5(v_4, v_2), e_6(v_4, v_2)\} \quad (16.1)$$

Puede observarse que para un grafo dirigido el orden del par de nodos que define la arista es decisivo, para un grafo no dirigido el orden no tiene relevancia.

**Figura 1-9:** Representación gráfica de un grafo. (a) Grafo dirigido o dígrafo, (b) Grafo no dirigido (c) Grafo acíclico dirigido



## 1.6.2 Definiciones en los grafos

Hay muchos tratados sobre grafos, pero en este trabajo nos centraremos en las definiciones y en sus características de manera que permita abordar el análisis y modelado de la red de tubería única por la cual se distribuye el riego y fertirriego en un cultivo que utiliza riego localizado o por goteo.

Ya se ha definido que un grafo lo constituyen los nodos o vértices y las aristas que interconectan estos vértices, y que estos podrían ser dirigidos o no. Un grafo con números en las aristas se le llama grafo ponderado y el peso de la arista ponderada es  $k$ , este peso representa un parámetro del objeto que se quiere representar mediante el grafo. Una arista que incide en un mismo nodo se le llama lazo, si dos o más aristas se asocian

con el mismo par de nodos se les llama aristas paralelas [45]. Un grafo que no tiene aristas paralelas ni lazos se conoce como grafo simple.

Dado un grafo  $G = (V, E)$  y un nodo  $v_i$  el conjunto de nodos adyacentes a  $v_i$  es el conjunto de nodos que son directamente alcanzables desde  $v_i$ . Una trayectoria es una sucesión de nodos cada uno adyacente al siguiente, de modo que al comenzar en un nodo y viajar a través de una arista elegida, nunca se vuelve a utilizar la misma arista dos veces.

Un ciclo es una trayectoria de longitud diferente de cero que comienza en un nodo y finaliza en el mismo sin pasar más de una vez por el mismo nodo. Un grafo dirigido se denomina cíclico si al menos contiene un ciclo, en caso contrario se denomina grafo acíclico dirigido, ver Figura 1-9c. Los grafos acíclicos que tiene como máximo un nodo de descendencia o padre, se les llama arboles simples de lo contrario poliárboles. En un grafo acíclico dirigido simple en donde las aristas son ponderadas, suele llamársele red, la longitud de una trayectoria es la suma de los pesos de las aristas de la trayectoria. Para la red hidráulica a representar estas definiciones se ajustan perfectamente.

### 1.6.3 Representación de los grafos

Hay tres formas de representar un grafo, dos de ellas ya utilizadas en este trabajo, cada una de ellas necesaria en su contexto. La primera es la *forma simbólica* donde los nodos y aristas se expresa como un par  $(V, E)$ , aquí  $V$  es un conjunto de nodos y  $E$  un conjunto de Aristas. Esta representación es útil para expresar relaciones y propiedades en los grafos y la construcción de algoritmos para su representación. La segunda es la *gráfica*, conforma el grafo por medio de un diagrama que interconecta un conjunto de nodos, uno para cada variable y un conjunto de líneas o flechas que corresponde cada arista del conjunto. Esta representación es particularmente útil porque permite que se visualice las propiedades y las características topológicas del grafo, aunque a veces puede resultar un diagrama complicado. La tercera es una representación más formal del grafo, es decir, *numérica*. Ésta a través de cierto tipo de matrices permite obtener las características de los grafos por simples manipulaciones algebraicas. Las matrices que se obtiene para los dígrafos son: la matriz  $A$  de adyacencia, está compuesta por valores enteros, en donde los índices de las filas y columnas son la identificación o etiqueta del nodo  $v_i$ . En esta

matriz de dimensión  $n \times n$  un elemento  $a_{ij}$  representa las aristas o su propiedad que la relaciona con los nodos  $i$  y  $j$ . En la Figura 1-10 se muestran la matriz de adyacencia para cada uno de los grafos mostrados en la Figura 1-9. Mediante esta matriz se obtiene el número de trayectorias y la longitud de las trayectorias, los grados internos y externos de un vértice en un grafo dirigido. La representación *gráfica* y la *numérica* se utilizarán para representar el sistema de la red hidráulica y sus características.

**Figura 1-10:** Representación Matricial de un grafo. (a) Grafo dirigido o dígrafo, (b) Grafo no dirigido, (c) Grafo acíclico dirigido ponderado.

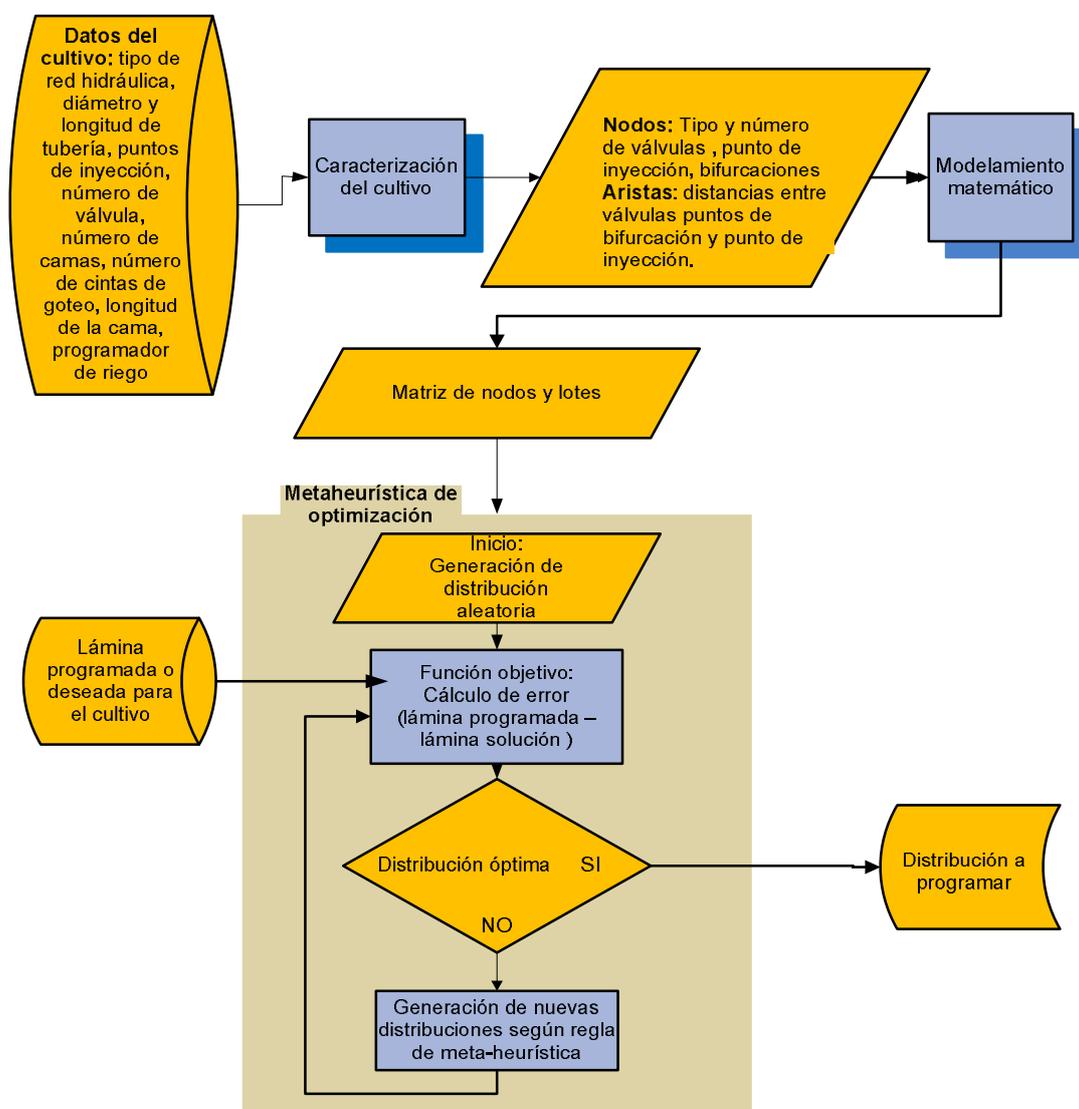
$$\begin{array}{ccc}
 A_d = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} \begin{pmatrix} v_1 & v_2 & v_3 & v_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} ; A_{nd} = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} \begin{pmatrix} v_1 & v_2 & v_3 & v_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} ; A_{adp} = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} \begin{pmatrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ 0 & k_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_2 & 0 & k_4 & 0 \\ 0 & 0 & 0 & k_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_5 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 \text{(a)} & \text{(b)} & \text{(c)}
 \end{array}$$

## 2. Marco experimental

### 2.1 Metodología Propuesta.

A partir de los objetivos específicos propuestos, se desarrolló una metodología con cuatro procesos bien definidos que permitieron alcanzar el objetivo de este trabajo. Un diagrama de flujo (Figura 2-11) muestra la secuencia en que se llevaron a cabo estos procesos.

**Figura 2-11:** Diagrama de flujo de la metodología utilizada

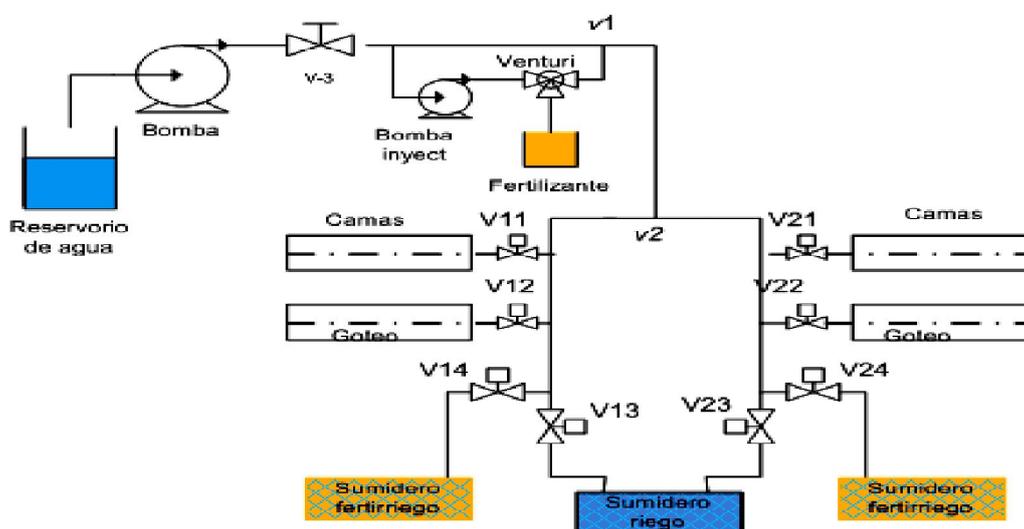


A continuación se hace un desarrollo de cada uno de estos procesos.

### 2.1.1 Características y parámetros del sistema típico de riego objeto de estudio

La solución al problema de la distribución de una lámina de riego asignada a un cultivo de plántulas mediante algoritmos evolutivos se realizó tomando como referencia el sistema de riego por goteo con red hidráulica en lazo abierto que se presenta simplificado en la Figura 2-12. Se pone de manifiesto que la red hidráulica mostrada en la Figura 2-12 cumple con los requerimientos de presión en todo el circuito hidráulico. Es decir, las especificaciones de las tuberías y los componentes del sistema dados en el anexo D, aseguran que las caídas de presión están en las tolerancias necesarias para que el funcionamiento de los goteros este en el valor especificado por el fabricante para cualquier punto del cultivo y pueda garantizarse su caudal (litros/hora). Sin embargo, en cada válvula de riego hay un regulador de presión que permite ajustar la presión a la cual trabajan los goteros.

**Figura 2-12:** Esquema simplificado de un cultivo con riego por goteo



Los parámetros que fueron necesarios para caracterizar la red de la Figura 2-12 y que permitieron elaborar un modelo representativo de ella, se enuncian y describen a continuación.

- **Numero de válvulas:** se seleccionaron cuatro válvulas para riego, identificadas como V11, V12 y V21, V22, y cuatro válvulas para el desperdicio identificadas como V13, V14 y V23 V24. Cada válvula se asignó como un nodo en la red. Este número obedeció a que era suficiente para obtener una solución al problema de la distribución de una lámina. Un número mayor de válvulas conduce a plantear un problema de mayor complejidad. Sin embargo el sistema hidráulico impone unas restricciones a la cantidad de válvulas a utilizar, una solución para esta particularidad puede abordarse a partir de los resultados obtenidos en este trabajo. Desde el punto de vista de diseño, la cantidad de válvulas se obtiene aplicando una metodología estandarizada a su cálculo [13, 14, 15]. Este tiene en cuenta los siguientes parámetros del cultivo: la cantidad de camas y sus dimensiones, el número de plántulas por cama y el volumen de riego o fertirriego para cada plántula. El cálculo del caudal máximo que maneja cada válvula y su dimensión se realizan con los datos del Anexo D.
- **Direccionalidad de flujo:** el líquido que se desplaza por la tubería tiene solo una dirección y es desde la inyección hasta la salida por las válvulas de riego y desperdicio, no hay circuitos de retorno en la red hidráulica.
- **Punto de bifurcación de la tubería (V2):** son los puntos donde la tubería se divide en dos ramales. Se considero como nodo un de la red.
- **Punto de inyección de agua y de fertilizante (V1):** se considera el nodo origen, en este punto se inyecta la tubería principal el fertilizante y cuando no se hace es agua la que se está inyectándose a la red.
- **Válvula general del sistema (V3):** su función es controlar la salida de agua al cultivo, cuando el riego está en funcionamiento se encuentra abierta. Es común que tenga incorporada la función de medir el flujo de agua (hidrómetro).

A continuación se hace una relación entre los puntos de interés de la red hidráulica y los nodos que se tomaran para el grafo que representara el cultivo.

- Punto de inyección de fertilizante a la red principal V1, se asigno como el nodo  $v_1$ ,
- Punto de Bifurcación de la tubería V2, se asigno como el nodo  $v_2$ .
- Válvulas de riego V11, V12 y V21, V22, se asignaron como los nodos  $v_3, v_4, v_5, v_6$  respectivamente.
- Válvulas de desfogue de desperdicio como V13-V14 y V23-V24, se asignaron como los nodos  $v_7$  y  $v_8$  respectivamente.
- Longitudes de la tubería: las longitudes entre los nodos de la tubería se describen en la Tabla 2-1.

**Tabla 2-1:** Longitudes entre nodos

Nodos	Longitud en metros (m)	Trayectoria	Longitud trayectoria (m)
$v_1$ al $v_2$	352	1	352
$v_2$ al $v_3$	118	2	178
$v_3$ al $v_4$	30		
$v_4$ al $v_5$	30		
$v_2$ al $v_6$	92	3	152
$v_6$ al $v_7$	30		
$v_7$ al $v_8$	30		

- Función de los sumideros: el sumidero de recolección de fertirriego recoge el fertirriego que ha sido causado como desperdicio en la tubería durante la distribución de una lámina o en el lavado de la tubería. El sumidero de recolección de agua actúa como sistema de recolección del agua generada por desperdicio durante la distribución de una lámina conduciéndola al reservorio.

▪ **Calculo del volumen por unidad de longitud de la tubería.**

Para los siguientes cálculos, se utilizan los valores dados en el anexo D. Inicialmente se calcula el volumen de líquido en la tubería por unidad de longitud, en este caso por metro. Se utiliza la siguiente expresión:

$$V_{ul} = \pi \frac{D_{ext} - 2*ET}{4} * 100 \quad (2.1)$$

donde:

$V_{ul}$  = Volumen por unidad de longitud,

$D_{ext}$  = Diámetro exterior,

$ET$  = Espesor de la tubería.

$$V_{ul} = 8.507 \text{ l} * \text{m}$$

▪ **Calculo de la velocidad del fluido por la tubería**

Para obtener este parámetro se necesita conocer el caudal de la válvula y un volumen de líquido en un tramo de la tubería. La expresión para obtener el caudal de la válvula por hora se muestra en la ecuación 2.2. Los valores para cada una de las variables de la ecuación son dados en las especificaciones del sistema y se encuentran en el anexo D.

$$Q_{V/h} = \frac{N_{lgc} * l_c}{d_{gl}} * Q_{g/h} * N_c \quad (2.2)$$

donde:

$Q_{V/h}$  = caudal de cada válvula por hora.

$N_{lgc}$  = número de líneas con goteros por cama = 4

$l_c$  = longitud de cada cama = 33m

$d_{gl}$  = distancia entre goteros de la línea = 0,15 m

$Q_{g/h}$  = caudal gotero/ hora = 1 l/h

$N_c$  = número de camas = 32

El caudal de la válvula es de

$$Q_{v/h} = 28160 \text{ l/h}$$

Luego se calcula el tiempo que tarda para evacuarse un volumen de un tramo de la tubería por una de las válvulas. La expresión que calcula este tiempo es una relación entre el volumen que hay en un tramo de la tubería y el caudal de la válvula. El volumen del tramo (ver ecuación 2.2) se calcula con su longitud y el volumen por unidad de longitud obtenido con la ecuación 2.1

$$V_{TT} = l_t * V_{ul} , \quad (2.2)$$

donde:

$V_{TT}$  = Volumen en un tramo de tubería.

$l_t$  = longitud de un tramo entre dos válvulas consecutivas=30m

$V_{ul}$  = Volumen por unidad de longitud =8.507 l \* m

$$V_{TT} = 255.21 \text{ l}$$

Ahora, el tiempo se puede calcular con la ecuación 2.3

$$t_{ev} = \frac{V_{TT}}{Q_{v/h}} * 60 \quad (2.3)$$

donde :

$t_{ev}$  = El tiempo de evacuación del líquido

$V_{TT}$  = Volumen en un tramo de tubería. El volumen tomado es el que hay entre dos válvulas consecutivas: 255.21 l

$Q_{v/h}$  = Caudal de cada válvula por hora: 28160 l/h

$$t_{ev} = 0,543 \text{ min}$$

La velocidad del fluido en la tubería se calcula con la ecuación 2.4.

$$v_{IT} = \frac{l_{tt}}{t_{ev}} \quad (2.4)$$

donde:

$v_{IT}$  = velocidad del líquido en la tubería.

$l_{tt}$  = longitud del tramo

$t_{ev}$  = Tiempo de evacuación del líquido en el tramo

$$v_{IT} = 55.24 \text{ m/s}$$

La velocidad promedio que presenta el líquido en la tubería está dentro del rango especificado para un flujo laminar [31].

De este objetivo se obtuvieron las siguientes variables para el cultivo: número de válvulas de riego y de desperdicio, número de nodos. Se definieron los valores de las distancias entre nodos, el volumen por unidad de longitud que almacena la tubería y la velocidad del fluido por la tubería. Parámetros que se utilizaran para conformar la matriz de nodos y lotes en la siguiente sección.

### **2.1.2 Modelamiento del cultivo típico con riego por goteo obtención de la función de aptitud.**

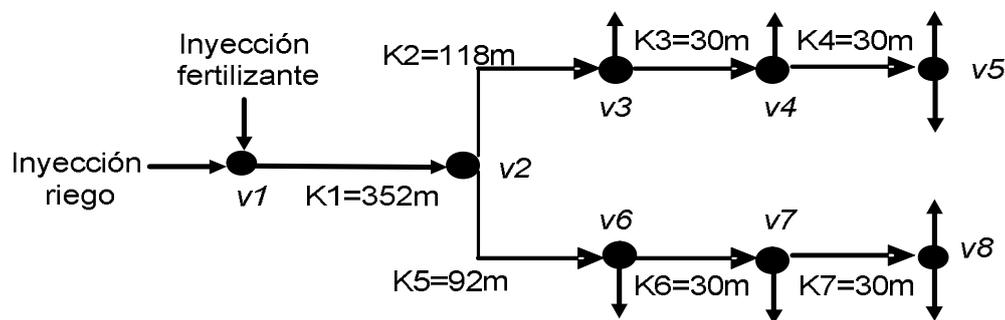
Para modelar el cultivo típico se utilizó la información obtenida en el numeral 2.1.2, concluyéndose que el tipo de grafo que permite representar la red hidráulica del cultivo es un acíclico dirigido, debido a que el flujo es en una sola dirección y no hay una trayectoria de realimentación o retorno del líquido por circuitos de la red hidráulica. A continuación se obtiene un modelo que permite reconstruir la red hidráulica a partir de sus parámetros.

#### **▪ Representación mediante un grafo de la red hidráulica de lazo abierto**

Como se ha mencionado en este trabajo, se necesita de una expresión que permita reconstruir la red hidráulica y recorrerla por sus puntos críticos (nodos), en especial aquellos a donde es necesario hacer llegar riego y/o fertirriego, además de que exprese la capacidad de almacenamiento de líquido entre estos puntos. Para obtenerla se tuvo como punto de partida el cultivo objeto de estudio de la Figura 2-12 y la correspondencia expresadas en la sección 2.1.1 entre los elementos de la red hidráulica y los de un grafo. La expresión obtenida es el grafo que se muestra en la Figura 2-13. En este los  $v_i$  son

los nodos y los pesos  $k_i$  de las aristas indican la distancia que hay entre nodos. Las longitudes de la tubería que está después de las válvulas de riego corresponden a las cintas de goteo, estas no se tuvieron en cuenta en el grafo debido a que son muy cortas y el tiempo que tomaría el líquido para desplazarse por ella mientras sale a las plántulas es muy pequeño.

**Figura 2-13:** Grafo representativo del cultivo donde los nodos son las válvulas y las aristas las distancias



#### ▪ Matriz de distancias del grafo

El grafo de la Figura 2-13 se representa numéricamente mediante la matriz **A** de adyacencia, definida en el numeral 1.6.3, conformando la matriz de pesos o de distancias  $M_d$ . Esta se muestra en la Figura 2-14.

**Figura 2-14:** Matriz de pesos o de distancias

$$M_d = \begin{bmatrix} 0 & 352 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 118 & 0 & 0 & 92 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 \end{bmatrix}$$

#### ▪ Matriz de nodos y lotes

Se tomo la longitud de la tubería entre dos válvulas consecutivas, la cual permanece constante en todo el cultivo, como unidad básica de almacenamiento de líquido o unidad básica de volumen. En la sección 2.1.1 se calculo el volumen por unidad de longitud para

la tubería. Definir esta unidad básica de volumen es útil debido a que podrá aumentar la posibilidad de que solo quede un insumo entre válvulas (riego o fertirriego) y por lo tanto, cuando se habrá una válvula para depositar un determinado insumo, no vaya a depositarse parte del otro.

Para el sistema de riego en estudio, la longitud entre válvulas es 30 m, si la matriz  $M_d$  de distancias se divide por este escalar, se obtiene una matriz de unidades básicas de longitud, o de unidades básicas de volumen de líquido,  $M_{ub}$ . A esta unidad básica la llamaremos “lote” (*Batch*). En la Figura 2-15 se muestra la matriz obtenida.

**Figura 2-15:** Matriz de unidades básicas de longitud o volumen (“Lotes”)

$$M_{ub} = \begin{pmatrix} 0 & 11.73 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.93 & 0 & 0 & 3.06 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

La matriz  $M_{ub}$  obtenida presenta la mayoría de entradas en ceros, el tiempo de procesamiento computacional se incrementa, debido a que cada elemento de la matriz se debe buscar y acceder. Por lo tanto, es más conveniente expresarla en una forma más compacta, procediendo de la siguiente forma: cada entrada no nula se representa por un vector fila de tres entradas, las dos primeras entradas del vector fila recogen la ubicación del valor no nulo de la matriz de unidades básicas (ver Figura 2-15). En la tercera entrada del vector fila, se asigna la cantidad no nula de la matriz de unidades básicas. La matriz resultante  $M_{nl}$  se muestra en la Figura 2-16. El comando de Matlab ***sparse***, realiza la operación compactar una matriz con valores no nulos dispersos. Puede apreciarse (ver Figura 2-16) que la matriz  $M_{nl}$  queda expresada directamente con el número de lotes que hay entre nodos consecutivos, lo que permite recorrer y reconstruir el grafo y por lo tanto la red hidráulica.

**Figura 2-16:** Matriz de nodos y lotes

$$M_{nl} = \begin{pmatrix} 1 & 2 & 11.73 \\ 2 & 3 & 3.93 \\ 3 & 4 & 1 \\ 4 & 5 & 1 \\ 2 & 6 & 3.06 \\ 6 & 7 & 1 \\ 7 & 8 & 1 \end{pmatrix}$$

La matriz  $M_{nl}$  de la Figura 2-16 es el insumo que le define a los algoritmos *eval* y *ruta* y a los algoritmos evolutivos *GA*, *PSO* y *CS* el sistema de riego por goteo.

▪ **Equivalencia entre un lote de riego o fertirriego y un volumen correspondiente**

A continuación se calcula la equivalencia de un lote de riego o fertirriego con la siguiente ecuación:

$$L_{r-fr} = D_{VV} * D_{VV} \quad (2.4)$$

donde :

$L_{r-fr}$  = Lote de riego o fertirriego

$D_{VV}$  = Distancia entre dos válvulas consecutivas

$D_{VV}$  = Volumen por unidad de longitud.

$L_{r-fr}$  = 255 litros

Los resultados que en este trabajo serán arrojados por la solución estarán dados en un número de lotes, por lo tanto la interpretación se deberá hacer con la equivalencia entre un lote y un volumen de riego o fertirriego, donde 1 lote es a 255 litros.

El tiempo para evacuar un lote por una válvula, se calcula mediante la expresión.

$$t_{Etr-fr} = L_{r-fr}/Q_{V/h} \quad (2.5)$$

Donde:

$t_{Etr-fr}$  = tiempo para evacuar un lote de riego o fertirriego a través de una válvula

$Q_{V/h}$  = Caudal máximo de la válvula de riego por hora. Se obtuvo con la ecuación (2.2)

$L_{r-fr}$  = lote de fertirriego en unidades de litros

$t_{Elr-fr}$  = 0,45 min o 27 s

#### ▪ **Función de aptitud para evaluar la distribución del riego fertirriego**

Se programo con buenos resultados una función de aptitud en Matlab capaz de optimizar la distribución de una lámina de riego. La función de aptitud, se conformo con el algoritmo **ruta** (algoritmo 2-1) y el algoritmo **eval** (algoritmo 2-2). Sin embargo con el fin de permitir una reproducibilidad de los resultados, su uso en trabajos futuros y modificaciones a su planteamiento se elaboro un pseudocódigo detallado para cada función.

Los insumos o entradas para las funciones son seis (6): la matriz  $M_{nl}$  de nodos y “lotes”, un vector de las válvulas de riego del cultivo, un vector de las válvulas de desperdicio del cultivo, un vector solución generado y entregado por el algoritmo evolutivo (AE) y dos vectores con la programación de la lámina de riego que se desea distribuir: uno para riego y el otro para fertirriego.

A continuación se describe como procede la función de aptitud. El algoritmo **ruta**, a partir de la matriz  $M_{nl}$ , entrega el camino o ruta para llegar a una válvula del cultivo. El algoritmo **eval** utiliza esta ruta para desplazar los lotes por la red hasta las válvulas. Para determinar el error (ver ecuación 2.6) y el desperdicio (ver ecuación 2.8) en el riego y en el fertirriego, el algoritmo (algoritmo 2-2) hace una comparación (resta) de cada una de las soluciones que genera el algoritmo evolutivo EA, con respecto a la lámina que se desea distribuir. Se aclara, que una solución arrojada por el EA, consiste en un vector de números reales que al ser decodificado por un procedimiento del algoritmo 2-2, se obtiene un arreglo de dos vectores: uno con una secuencia de válvulas y el otro con una secuencia de “lotes”. El algoritmo 2-2 distribuye los “lotes” hacia las válvulas. Al finalizar la distribución el algoritmo 2-2 hace la resta de lo que llegado a cada válvula con lo programado en la lámina para la misma válvula, obteniéndose el error que se desea minimizar. Cada error calculado por la función de aptitud se envía al EA, este genera otra distribución que es enviada a la función de aptitud realizando para ella una nueva evaluación. Este ciclo se continúa repitiendo hasta que se cumpla su condición de

parada, que es el número de iteraciones programado, al final el EA entrega el mínimo error que cálculo. La función que calcula el error se muestra en la ecuación (2.6).

▪ **Calculo del error en Riego y Fertirriego**

$$\text{Min } E_{tRF} = \left( R_d - \sum_{L=1}^{L_{max}} (L_{RV}) \right)^2 + \left( F_d - \sum_{L=1}^{L_{max}} (L_{FV}) \right)^2 \quad \forall L_{RV} = 0 ; \forall L_{FV} = 1 \quad (2.6)$$

Siempre y cuando  $L_{RV} \wedge L_{FV} \in \{0,1\}$

$R_d \wedge F_d \in (\mathbb{Z} \in \mathbb{N})$

$L_{RV} \in$  al vector Riego [ $L_{RV(1,1)} \dots \dots L_{RV(1,L_{max})}$ ]

$L_{FV} \in$  al vector Fertirriego [ $L_{FV(1,1)} \dots \dots L_{FV(1,L_{max})}$ ]

donde:

$E_{tRF}$  = Error total en riego y fertirriego

$V$  = Número de válvulas

$L_{max}$  = Número de lotes máximo

$R_d$  = Riego (agua) programada

$F_d$  = Fertirriego programado

$L_{RV}$  = Lote de Riego que llega a válvula con programación para riego

$L_{FV}$  = Lote de Fertirriego que llega a válvula con programación para fertirriego

▪ **La minimización del error está condicionada a las siguientes restricciones:**

- El número máximo de lotes para evaluar la distribución de una lámina debe ser menor que la suma de lotes programados y de los que contenga la tubería.

$$L_{max} < L_{tub} + TL_{prog} \quad (2.7)$$

donde:

$L_{max}$  = Número máximo de lotes para evaluar la distribución de una lámina.

$L_{tub}$  = Cantidad máxima de lotes que aloja la tubería.

$TL_{prog}$  = Cantidad total de lotes programados en la lámina.

- Asegurar que el tiempo de entrega de la lámina no puede superar el tiempo máximo dado a una jornada de riego. Este tiempo de entrega depende del caudal de la válvula y del número de lotes máximos para distribuir una lámina.

$$\frac{\sum_{L=1}^{L_{max}}(L_{Re})}{Q_{V/h}} + \frac{\sum_{L=1}^{L_{max}}(L_{Fe})}{Q_{V/h}} \leq t_{max \text{ entreg lam}} \leq t_{max \text{ JFR}} \quad (2.8)$$

donde:

$Q_{V/h}$  = Caudal hora de la válvula de riego.

$t_{max \text{ entreg lam}}$  = Tiempo máximo para la entrega de una lámina de riego.

$t_{max \text{ JFR}}$  = Tiempo máximo dado a una jornada de riego.

### ▪ **Calculo del desperdicio en Riego**

La función que calcula el desperdicio en riego (agua) hace la diferencia entre el total de lotes del vector de riego estimado por el AE y el total de lotes del vector de riego que llega a las válvulas, es la siguiente:

$$Desp \text{ Riego} = \sum_{i=1}^k L_{VRE} - \sum_{L=1}^{L_{max}} L_{RV} \quad (2.10)$$

Donde:

$Desp \text{ Riego}$  = Desperdicio en riego

$L_{VRE}$  = Lote vector de riego estimado

$i$  = posición inicio del vector lote de riego estimado.

$k$  = última posición del vector de lote riego estimado

$L_{RV}$  = Lote de Riego que llega a válvula con programación para riego  
 $L_{max}$  = Número de lotes máximo

- **La minimización del desperdicio en riego está condicionada a la siguiente restricción:**

$$Desp\ Riego < L_{tub} \quad (2.11)$$

Donde:

$L_{tub}$  = Cantidad máxima de lotes que aloja la tubería.

- **Calculo del desperdicio en Fertirriego**

La función que calcula el desperdicio en fertirriego hace la diferencia entre el total de lotes del vector de fertirriego estimado por el AE y el total de lotes del vector de fertirriego que llega a las válvulas. Es la siguiente:

$$Desp\ Ferti = \sum_{i=1}^k L_{VFRE} - \sum_{L=1}^{L_{max}} L_{FRV}$$

(2.12)

$L_{VFRE}$  = Lote vector de riego estimado.

$i$  = posición inicio del vector lote de fertirriego estimado.

$k$  = última posición del vector de lote fertirriego estimado.

$L_{FRV}$  = Lote de fertirriego que llega a válvula con programación para fertirriego.

$L_{max}$  = Número de lotes máximo

- **La minimización del desperdicio en fertirriego está condicionada a la siguiente restricción:**

$$Desp\ Riego < L_{tub} \quad (2.13)$$

Donde:

$L_{tub}$  = Cantidad máxima de lotes que aloja la tubería.

Una descripción más detallada de la salida que aporta la solución es la siguiente. El algoritmo 2-2 decodifica el vector solución enviado por el EA, obteniéndose un arreglo de dos vectores fila: el primero es una palabra binaria, donde, un cero (0) corresponde a un "lote" de riego y un uno (1) corresponde a un "lote" de fertirriego. Este vector es el número de "lotes" que deben enviados para cumplir con la lámina deseada o programada con un mínimo de error y de desperdicio. El segundo, es un vector con las válvulas del cultivo, las válvulas pueden estar repetidas en el vector, a cada válvula se le asocia un bit de la palabra binaria.

Físicamente funcionaria de la siguiente forma: se activa la primera válvula durante el tiempo que se necesita para desalojar el lote de insumo que hay en su entrada. En ese mismo instante el bit ("lote") menos significativo de la palabra binaria se inyecta a la tubería durante el tiempo que la válvula esta activa. Estos eventos se repiten hasta concluir con la última válvula y el último bit ("lote") de la palabra binaria.

A continuación se describe el pseudocódigo del algoritmo **ruta**. (Algoritmo 2-1)

**Algoritmo 2-1:** Pseudocódigo algoritmo **ruta**

1. **Algoritmo ruta**
2. *Cargar MATRIZ % Matriz que es el modelo que representa la red hidráulica*
3. *Cargar valvula % Una de las n Válvulas de riego y desperdicio*
4. *uniones ← las dos primeras columnas de Matriz*
5. *raiz ← válvula.*
6. *recorrido ← [ ]*
7. *Procedimiento ruta.*
8.     *mientras válvula sea ~= 0*
9.         *fila ← buscar en la segunda columna de uniones, la fila donde esta raiz*
10.         *recorrido ← recorrido y fila % es la salida de la función*
11.         *raiz ← actualizar raiz con fila*
12.     *fin mientras*

13. Fin procedimiento

El pseudocódigo del algoritmo **eval** puede verse a continuación (Algoritmo 2-2).

**Algoritmo 2-2:** Pseudocódigo algoritmo **eval**

1. **Algoritmo eval**
2. Definir la matriz que representa el modelo de la red hidráulica **MATRIZ**
3. Cargar el vector solución  $x$  generado por el algoritmo de optimización
4. Definir el vector de riego deseado **aguad**
5. Definir el vector de fertirriego deseado **fertd**
6. Definir el vector de válvulas de riego **numval**
7. Definir el vector de válvulas de desperdicio de agua **valea**
8. Definir el vector de válvulas de desperdicio de fertirriego **valef**
9. Definir el vector de lotes de prueba para el envío de una lámina **lotes**
10. Definir el vector longitud palabra binaria **vexp**
11. Definir vector de desperdicio de agua **despa**
12. Definir vector de desperdicio de fertirriego **despf**
13. Procedimiento eval.
14.  $nvb \leftarrow$  Calcular el número de variables binarias con **lotes** /  $n$
15. **VRF**  $\leftarrow$  conformar el vector riego-fertirriego con los  $nvb$  primeros elementos de  $x$
16. **VRFB**  $\leftarrow$  Convertir cada elemento de VRF a binario.
17. **VVRD**  $\leftarrow$  Con los  $x - nvb$ , Conformar el vector de Válvulas de riego y desperdicio
18. **DIST**  $\leftarrow$  Crear matriz con los vectores VRFB y VVRD
19. Para contador  $\leftarrow$  desde 1 hasta longitud filas de MATRIZ
20. Lista {contador}  $\leftarrow$  Crear listas de longitud igual a cada valor de la tercer
21. Columna de MATRIZ.
22. Fin para.
23. **agua**  $\leftarrow$  Crear vector columna de ceros con longitud de primera columna de MATRIZ.
24. **fert**  $\leftarrow$  Crear vector columna de ceros con longitud de primera columna de MATRIZ.
25. Para contador  $\leftarrow$  desde 1 hasta longitud de la segunda fila de DIST
26. **valvula**  $\leftarrow$  llamar primera válvula de la fila dos de DIST
27. llamar la función **ruta**
28. Si la ruta de **valvula** está en ceros

29.	<i>desplazar un cero (0) hacia <b>agua</b></i>
30.	<i>Si No</i>
31.	<i>desplazar un uno (1) hacia <b>fert</b></i>
32.	<i>Fin Si</i>
33.	<i>Actualizar lista con <b>agua</b> y <b>fert</b></i>
34.	<i>Fin para.</i>
35.	<i><b>erroragua</b> ← Calcular error entre <b>aguad</b> y <b>agua</b> mediante una diferencia</i>
36.	<i><b>errorfert</b> ← Calcular error entre <b>fertd</b> y <b>fert</b> mediante una diferencia</i>
37.	<i><b>erroreval</b> ← Calcular error cuadrático total <b>erroragua</b><sup>2</sup> + <b>errorfert</b><sup>2</sup>.</i>
38.	<i><b>despa</b> ← Determinar desperdicio de agua consultando número de activaciones de <b>valea</b></i>
39.	<i><b>despf</b> ← Determinar desperdicio de fertirriego consultando número de activaciones de <b>valef</b></i>
40.	<i>Fin procedimiento</i>

### 2.1.3 Metodología de optimización aplicando técnica de computación evolutiva.

De acuerdo a la revisión bibliográfica, para alcanzar el tercer objetivo se seleccionó el algoritmo evolutivo *Cuckoo Search* (CS). Pudo determinarse que el algoritmo necesita de un número reducido de parámetros para ajustar, presenta buen equilibrio entre la exploración y la diversificación en la población sobre la cual se realiza la búsqueda de la solución y no necesita del conocimiento específico sobre el problema que intenta resolver. Tiene la ventaja además de que es software libre, está disponible y elaborado en código de Matlab [17]

De la salida arrojada por el CS, se esperaba una distribución que permitiera entregar una lámina de riego deseada con cero errores en la entrega de los insumos y un mínimo desperdicio.

Los parámetros del CS que se tuvieron en cuenta para observar el comportamiento de la respuesta fueron: la población o número de nidos ( $n$ ), la probabilidad de descubrimiento de una mala solución y ser remplazada por otra ( $p_a$ ), el número de iteraciones ( $N$ ), el índice  $\beta$  que define los valores del ancho de paso la varianza de vuelo de Lévy.

Para tener un espectro amplio de la respuesta que arroja cada solución y realizar un análisis de estas, cada simulación se realizó en un intervalo de 4 a 40 “lotes”. Se seleccionó la mejor respuesta de cada simulación con respecto a criterios definidos de error cero (0) y el desperdicio mínimo.

La respuesta a las simulaciones se obtuvo de forma gráfica pues permitía una forma comparativa más versátil para el análisis de cómo fue el comportamiento del error y del desperdicio para cada cadena de lotes en evaluación. El comportamiento del error de minimización para cada cadena de “lotes” también es mostrado gráficamente.

#### **2.1.4 Validación de los resultados**

El algoritmo CS por las ventajas que ya han sido mencionadas, fue el AE seleccionado para encontrar una solución óptima a la distribución de una lámina programada. Para validar los resultados obtenidos mediante el algoritmo CS, se utilizaron otras técnicas evolutivas como son los algoritmos genéticos (GA) y la optimización por enjambre de partículas (PSO). Metodologías con gran reconocimiento en la literatura y en las soluciones que han dado a diversidad de problemas [7, 8]. A diferencia del CS, estas presentan un mayor número de parámetros para ajustar, el alcance de su sintonía se limitó a que las respuestas que arrojaran estuvieran alrededor de las dadas por el algoritmo CS.

En la misma dirección que la seguida con el algoritmo CS, se realizaron 5 simulaciones para diferentes cantidades de población, y se ajustaron algunos parámetros que se detallan en la sección siguiente.

Las soluciones se presentaron en gráficas, indicando el comportamiento del error (en cantidades de lotes) para el riego y el fertirriego, el desperdicio ocasionado (en cantidades de lotes) y el error de minimización (número real). Se concluyó que el algoritmo CS entregó una solución adecuada a la distribución de una lámina.

## 2.2 Materiales y métodos

El problema de la distribución de una lámina de riego y fertirriego tuvo una solución computacional mediante el algoritmo de optimización CS y concretamente con la implementación en Matlab de este algoritmo realizado por el profesor Xin-She Yang en noviembre del 2008 y actualizado en febrero del 2013 [17]. Las soluciones entregadas por el CS se validaron sus mediante las técnicas evolutivas: Algoritmo genético del *Global optimization toolbox* de Matlab [46] y la herramienta para Matlab *Particle Swarm Optimization* de Trelea. Construido por Brian Birge en el año 2005 [47]. A continuación se detalla la utilización de estos algoritmos.

### 2.2.1 Algoritmo *Cuckoo Search* en Matlab

En comparación con otros algoritmos inspirados en la naturaleza, el algoritmo CS es fácil de entender y poner en práctica y tiene el número mínimo de parámetros para sintonizar. Este algoritmo para el ambiente de *Matlab* fue desarrollado por Xin-She Yang en noviembre del 2008 y actualizado en febrero del 2013. Presenta tres versiones para optimización sin restricciones:

- *Cuckoo\_search.m*: es para una tolerancia dada.
- *Cuckoo\_search\_new.m* : es para un número fijo de iteraciones, pero da la posibilidad de modificarse.
- *Cuckoo\_search\_spring.m*.: está limitada para la optimización del diseño de un resorte.

*Cuckoo\_search\_new.m*. Es la versión utilizada para las simulaciones en este trabajo. Este programa demo implementa una versión secuencial de algoritmo CS con el fin de que los vuelos Lévy y la generación de nuevas soluciones puedan utilizar métodos ligeramente diferentes. En este sentido, la versión *Cuckoo\_search\_new.m* empleada, generó los vuelos aleatorios de Lévy mediante el algoritmo de Mantegna [43]. Por lo tanto, el código que implementa difiere del propuesto en el libro *Nature-Inspired Metaheuristic Algorithms* [37], donde se expone el algoritmo básico. Como se ha venido mencionando, este algoritmo se caracteriza por tener pocos parámetros, a continuación se describen los que fueron utilizados.

***n*: número de nidos.**

Es la población. Cada nido contiene solamente un huevo, el cual es una posible solución. Se utilizó una cantidad de 25, que está por defecto en la versión del programa, y es el valor recomendado por el autor. Sin embargo, se realizó varias simulaciones modificando a un número mayor de 25 hasta 50 en saltos de 10, manteniendo el  $P_a$  y el número de iteraciones  $N$  en los valores dados por defecto. Se observó que los resultados no variaron significativamente.

 **$P_a$ : Probabilidad de descubrimiento de huevo huésped**

Este parámetro es utilizado para definir la conmutación o el salto que realiza el vuelo de Léavy. El autor recomienda utilizar el valor 0.25 y es el que utiliza por defecto el programa [13].

 **$N_{IterTotal}$ : Número de iteraciones**

Este parámetro se modificó para cada simulación. Se observó que la solución mejora a medida que este parámetro se incrementa. Sin embargo el tiempo de cómputo empeora. El número de iteraciones que se tomó por cada simulación fue 100, 200, 500, 1000, 1500.

**Índice  $\beta$**  : Este índice viene dado por defecto en su valor es 1.5. Como es un valor que afecta directamente los pasos de los vuelos, se varió en el intervalo que expresa la restricción de ecuación (8.1). Se observó que los mejores valores entregados por la solución del CS se obtenían para el valor dado por defecto.

**2.2.2 Algoritmo genético *Global Optimization toolbox* de Matlab**

*Global optimization toolbox* de *Matlab* [46] tiene implementado el algoritmo genético que minimiza una función objetivo. Esta es una de las herramientas de optimización utilizadas para validar la solución obtenida con el algoritmo CS al problema de la distribución del riego fertirriego. Este algoritmo es un método para resolver problemas de optimización con restricciones y sin ellas, mediante la analogía a la selección natural y la evolución biológica. En esencia lo que hace el algoritmo es cambiar continuamente una población constituida por un cierto número de individuos, que no son más que las posibles soluciones a un problema. En cada iteración, el algoritmo genético selecciona aleatoriamente los individuos de la población actual y los utiliza como padres para

producir las próximas generaciones. En el cruzamiento de estos padres, los hijos generados, con factores asignados de mutación, van produciendo generaciones, con individuos de cualidades óptimas, estos para las aplicaciones de optimización son las soluciones óptimas. El algoritmo genético es muy adecuado para resolver problemas de optimización que presenta un comportamiento discreto, pero no cuando la función objetivo es altamente no lineal o discontinua.

Los parámetros que se variaron en este algoritmo de Matlab fueron:

- Tamaño de la población: se tomo la siguiente cantidad de individuos: 300;500;1500;3000
- Cantidad de generaciones: 200
- La mutación. Para este parámetro se emplearon la función de probabilidad Gaussiana y la uniforme.
- La probabilidad de cruzamiento: 0,2 ; 0,4 ; 0,6 ; 0,8. Su modificación no afecto los resultados

### **2.2.3 Optimización por enjambre de partículas PSO, Trelea tipo2**

Este algoritmo fue implementado como la herramienta *Particle Swarm Optimization* para el entorno de Matlab por *Brian Birge* el 22 de abril de 2005. La versión utilizada para las simulaciones en este trabajo es la actualizada el 20 de marzo de 2006 [47].

La herramienta presenta dos funciones: "*parametrosps.m*" y "*pso\_Trelea\_vectorized.m*" En la primera se introducen los valores de los parámetros del algoritmo, y en la segunda está desarrollado el algoritmo, es en esta función donde se llama la función que se va a minimizar o maximizar, siempre y cuando tenga múltiples entradas y una sola salida. El algoritmo tiene dos criterios de parada, cuando se cumple el número máximo de iteraciones o el requerimiento del mínimo error establecido.

Esta versión permite implementar varios tipos de *PSO*:

- *PSO comun,*

- *Trelea* tipo 1,
- *Trelea* tipo 2
- *Clecr* clase 1

Cada tipo de *PSO* tiene una fórmula diferente que la actualiza la velocidad de las partículas.

Los parámetros de uso son declarados en la función "*parametrosps.m*" y la estructura de su de declaración es la siguiente:

$$[optOUT, tr, te] = PSOtype(funcname, D, mv, VarRange, minmax, PSOparams, \dots, plotfcn, PSOseedValue)$$

donde :

- *PSOtype*: es el tipo de *PSO* que se va a emplear

Las siguientes declaraciones corresponden a entradas necesarias:

- *funcname*: archivo de Matlab que contiene la función a optimizar, construida mediante el algoritmo **eval**.
- *D* = Número de dimensiones del problema, o número de variables de la función. Para el problema en este trabajo depende del número de "lotes" que se desea enviar para evaluar la distribución de una lámina.

Las siguientes declaraciones corresponden a entradas Opcionales, de lo contrario el programa las asume y les da los valores por defecto:

- *mv*: máxima velocidad de partícula, Su valor por defecto es  $mv = 4$ .
- *VarRange*: es una matriz de límites para cada variable de entrada (límite inferior y límite superior) de tamaño  $2 \times D$ . se tomaron positivos:
  - $minmax = 0$ , la función es minimizada (por defecto),
  - $minmax = 1$ , la función es maximizada.

Se utiliza el valor por defecto ya que se desea minimizar.

- La entrada *PSOparams* son los Parámetros de *PSO*:

- 
- P(1): Épocas para actualizar gráfico, por defecto=100, si es 0 = no gráfica.
- P(2): Máximo número de iteraciones (épocas) del ciclo, por defecto = 2000.
- P(3): Tamaño de la población, por defecto = 24.
- P(4): aceleración constante 1 (influencia del mejor valor personal  $pbest$ ), por defecto=2.
- P(5): aceleración constante 2 (influencia del mejor valor del enjambre(social)  $Gbest$ ), por defecto=2.
- P(6): Peso inicial de inercia, por defecto = 0.9.
- P(7): Peso final de inercia, por defecto = 0.4.
- P(8): Valor final de épocas cuando se asigna peso inercial, por defecto = 1500.
- P(9): Mínimo gradiente de error global, *if*  $abs(Gbest(i+1)-Gbest(i)) < gradient$  durante cierta longitud de épocas, termina la corrida, por defecto=1e-25.
- P(10): Épocas antes de que el criterio del error del gradiente termine de correr, por defecto = 150.
- P(11): error deseado, si su valor es NaN (Not a Number) entonces, minimizar o maximizar sin restricción, por defecto = NaN.
- P(12): Tipo de *PSO* para utilizar.
- 0 = *PSO* Común con inercia (por defecto).
- 1,2 = *PSO* tipo Trelea 1,2. En este trabajo se utilizó P(12)=2
- 3 = *PSO* restringido de Clerc, Tipo 1"
- P(13): *PSOseed*, por defecto = 0
- = 0 para posiciones iniciales aleatorias
- = 1 para partículas iniciales ingresadas por el usuario.
- *plotfcn*: nombre opcional de función encargada de graficar, por defecto '*goplotps*'. Se ingresó el nombre de la función de aptitud ***eval***.
  - *PSOseedValue*: Posición inicial de partículas, depende de P(13), el cual deber estar definido como 1 or 2, (no usado para P(13)=0). Este valor necesita ser de dimensiones  $n \times m$  donde  $n \leq$  Número de partículas, y  $m \leq$  Número de dimensiones. Si  $n < ps$  y/o  $m < D$ , entonces los valores restantes son escogidos aleatoriamente en *Varrange*.

La siguiente declaración corresponde a la salida:

- *[optOUT]*: salidas min/max asociadas con función, de la forma: [ *bestin1*; *bestin2*; ... *bestinD*; *bestOUT*].

Las siguientes declaraciones corresponden a las salidas opcionales:

- *tr*: Gbest en cada iteración, sigue el vuelo del enjambre.
- *te*: épocas alcanzadas, retorna como un vector de 1 hasta el número final de épocas.

Finalmente, el vector de parámetros quedo expresado en la siguiente forma.

```
Funcname = @(x)eval(x,MATRIZ,aguad,fertd,num_val,lotes,vexp);
```

```
[x,fval]=pso_trelea_vectorized(funcname,D,NaN,mat,0,[200,2000,300,2,+..2,0.9,0.4,1500,1e-25,150,NaN,2,0]);
```

## 2.3 Resultados

A continuación se presentan los resultados de las simulaciones realizadas para la lámina de riego y fertirriego mostrada en la Tabla 2-2. El procedimiento para obtener la solución de distribución más adecuada es el siguiente: se realizaron diferentes simulaciones para cada algoritmo de optimización, AG, PSO y CS. Se observó en todas las ejecuciones, que el intervalo de cadenas de “lotés”, que presentaba mayor interés se encontraba entre 20 a 30 cadenas. Sin embargo, para ampliar aun más el espectro, se realizaron todas las simulaciones en un intervalo entre 4 y 40 lotes.

**Tabla 2-2:** Programa de riego y fertirriego (Lámina deseada) para distribuir en el cultivo.

Válvula	V11	V12	V21	V22
Lotes de Riego	3	0	0	3
Lotes de Fertirriego	0	3	3	0

De las cinco (5) simulaciones realizadas para cada algoritmo evolutivo (ver anexo A, B y C) se seleccionó la gráfica con la mejor respuesta a partir de los siguientes criterios:

- Cero (0) error de “lotes” en riego y fertirriego,
- Cero (0) “lotes” de desperdicio en fertirriego,
- Desperdicio en riego menor que:

$$D_R < L_{AT} - L_{PR} \quad (2.6)$$

Donde:

$D_R$  = Desperdicio en riego

$L_{AT}$  = Cantidad de lotes que aloja la tubería.

$L_{PR}$  = Cantidad de lotes programados de riego.

Para cada mejor respuesta seleccionada se describe como debe de realizarse el envío o distribución de estos lotes, para que pueda cumplirse con la entrega de la lámina programada en la Tabla 2-2.

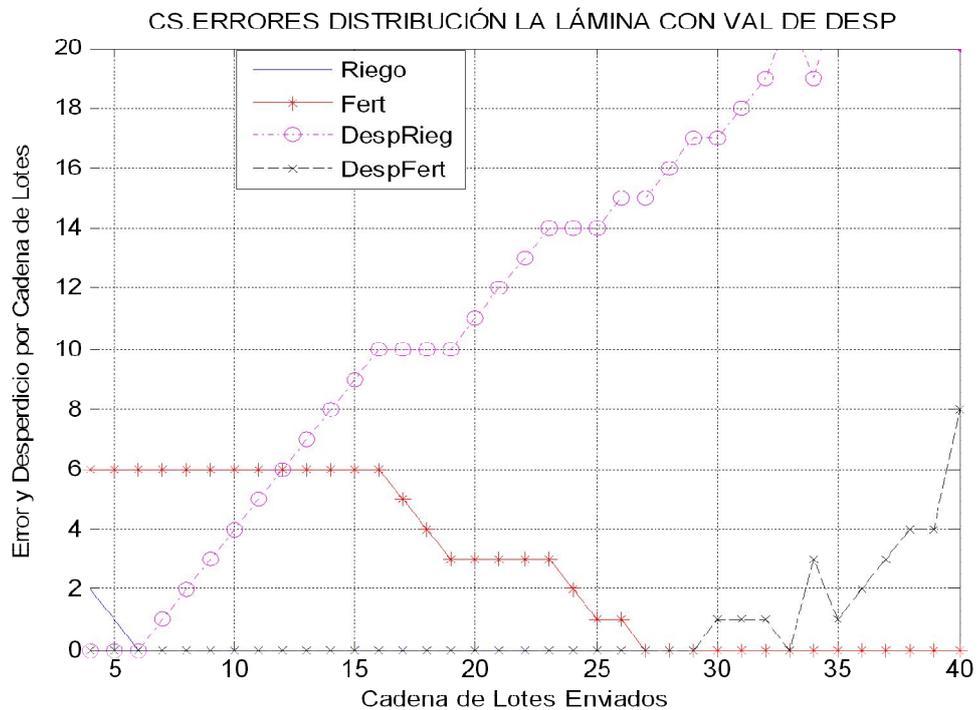
### 2.3.1 Simulación con el algoritmo CS

Para las siguientes iteraciones: 100, 300, 500, 1000, 1500 se realizaron 5 simulaciones con el programa del algoritmo búsqueda *Cuckoo* variando el número de nidos de 25 a 50. Se observó que no se presentaban modificaciones significativas en las respuestas, por lo tanto se realizaron de nuevo 5 simulaciones para cada iteración con un número fijo de 25 nidos (ver anexo C). De acuerdo a los criterios definidos en la sección 2.3 se seleccionó la mejor respuesta (ver Figura 2-17).

En la Figura 2-17 se muestra la gráfica del comportamiento del error en riego, en fertirriego y el desperdicio ocasionado de la simulación seleccionada. Los parámetros asignados para esta simulación fueron los siguientes:

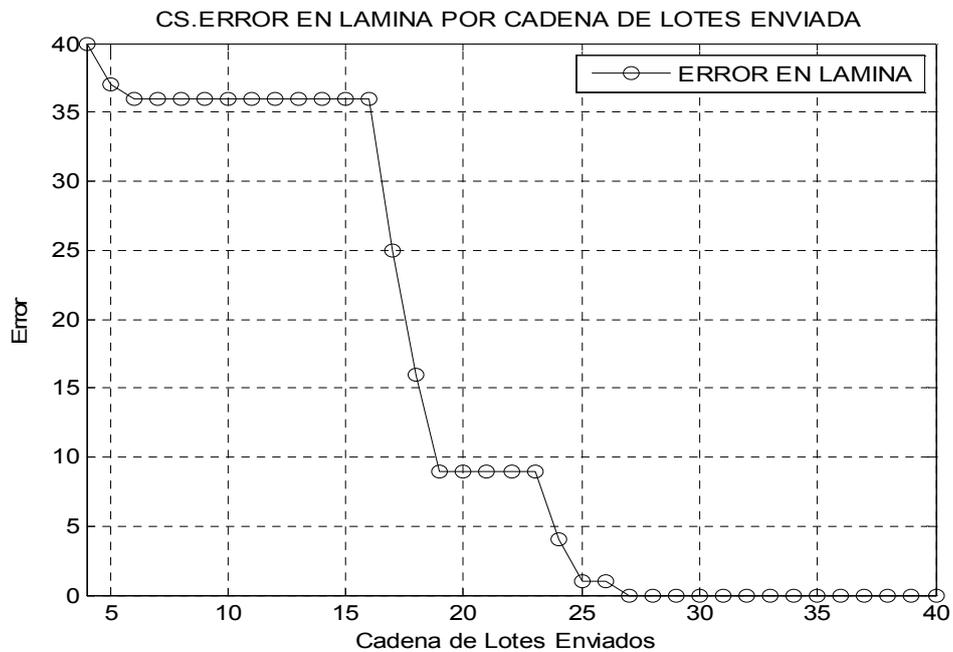
- Número de nidos: 25
- Probabilidad de descubrimiento: 0.25
- Número de iteraciones: 1500

**Figura 2-17:** Error y desperdicio en riego y en fertirriego con el CS



A continuación, en la Figura 2-18 se presenta el comportamiento del error que entrega el algoritmo *eval*.

**Figura 2-18:** Error calculado por el algoritmo *eval* para todas las cadenas de lotes



### ▪ Interpretación y análisis la simulación

En la figura 2-17 puede apreciarse los valores del error para el riego, el fertirriego y el desperdicio. Se observa un error de cero (0) para el riego y el fertirriego cuando se ha enviado una cadena de 27 lotes. Cumpliéndose con esta cantidad de lotes la lámina programada. También se observa la figura 2-17 que el desperdicio en fertirriego es cero (0) lotes y el desperdicio en el riego es de quince (15) lotes. Puede verse (Figura 2-17) que para un menor número de lotes el desperdicio en riego disminuye pero el desperdicio en fertirriego aumenta. En tres puntos, donde el número de lotes es de 28, 29, 33, se cumple las condiciones de cero (0) error en desperdicio de fertirriego, pero aumenta en uno (1), dos (2), y seis (6) lotes respectivamente, la cantidad de desperdicio de riego. Los valores superiores a 27 “lotes” no interesan debido a que el desperdicio de riego se incrementa y no se cumplen los criterios.

El algoritmo 2-2 a partir del vector solución  $x$  que entrega el CS, genera la matriz ***DISTR***. El tamaño de esta matriz para la cantidad de lotes seleccionados es de  $2 \times 27$ . La distribución de los lotes de riego y fertirriego para cumplir con la lámina programada con un mínimo desperdicio se realiza de la misma forma que para el GA y el PSO. En la Tabla 2-3, se muestra organizada esta matriz para su interpretación.

La información de la primera fila discrimina el tipo de insumo. La segunda fila le da nombre a los elementos de la tercera fila. La fila tres enumera el orden de la secuencia de activación de las válvulas y la inyección de fertilizante, esto sucede en el orden ascendente que se muestra. La fila cuatro, corresponde a los lotes de riego (0) y fertirriego (1). Estos son los insumos que se inyectarían en el punto correspondiente al nodo  $V_1$  de la Figura 2-12. La quinta fila, corresponde a las válvulas que se deben de activar para depositar los lotes de riego o fertirriego. Resumiendo, el evento uno consiste en abrir la válvula **V23**, y en el mismo instante inyectar fertilizante. El tiempo de inyección de riego o de fertilizante, es el mismo tiempo durante el cual la válvula esta activa. Este tiempo se calculó para un lote en la ecuación (2.5). De acuerdo a la descripción anterior se continúa con la interpretación de las filas cuatro y cinco, y el evento 2 siguiente, luego secuencialmente hasta el evento 27. Se pone nuevamente de manifiesto que la tubería antes de comenzar a distribuirse la lámina, debe contener solamente agua.

**Tabla 2-3:** Cadena de 27 lotes para realizar la distribución de la lámina con el CS.

RIEGO = 0 ; FERTIRRIEGO = 1																										
EVENTO																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	1	1	0	1	0	1	1	1
23	12	23	14	12	21	23	14	21	23	13	24	21	24	13	23	23	13	22	13	12	23	11	11	22	22	11
VALVULA																										

En la Tabla 2-4 se hace un resumen de los insumos programados, el error y el desperdicio. Puede verse que para cada válvula de riego se lista la cantidad de lotes que le llegan (Lotes acumul), los lotes que le han sido programados (Lotes Prog), y la diferencia entre los programados y los acumulados, que es el error (Err). Por ejemplo, para la válvula **V11**, se han programado 3 lotes de riego, han llegado a ella 3, por lo tanto el error es cero (0), la lámina se ha cumplido para esta válvula. Y lo mismo ocurre con la válvula **V21**. Las **V12** y **V22** programadas con fertirriego, también tiene el mismo resultado.

Para las válvulas de desperdicio se expresa la cantidad de lotes de riego y fertirriego que desalojan. De la Tabla 2-4, puede observarse que las válvulas **V13**, **V14**, **V23** y **V24** se han activado 4, 2, 7 y 2 veces respectivamente. Las aperturas de estas válvulas suman 15, que es el número de lotes de desperdicio de riego. Se verifica lo mostrado en la gráfica de la Figura 2-17. El error total en riego es cero (0) en fertirriego es cero (0), en desperdicio de fertirriego es cero (0) y en desperdicio de riego quince (15) lotes. Las válvulas 14 y 24 participan en el desalojo de desperdicio de riego, y nunca van a desalojar desperdicio de fertirriego para esta programación. En estas condiciones estas válvulas podrían desde el programador/controlador de riego, programarse como una sola válvula. Permitiendo de esta forma una mayor flexibilidad en la operación de la red.

**Tabla 2-4:** Resumen de la distribución de la lámina programada con el CS

Vál	V11			V12			V21			V22			V13	V14	V23	V24
Dato	Lot acu mul	Lote Prog	Err	Lote acu mul	Lote Pro g	Err	Lot acu mul	Lote prog	Err	Lote acu mul	Lote prog	Err	Lot de Desp	Lote de Desp	Lote de Desp	Lote de Desp
Fer- tirri- ego	0	0	0	3	3	0	3	3	0	0	0	0	0	0	0	0
Ri- Ego	3	3	0	0	0	0	0	0	0	3	3	0	4	2	7	2
ERROR TOTAL EN RIEGO.							0									
ERROR TOTAL EN FERTIRRIEGO							0									
DESPERDICIO TOTAL EN RIEGO							15									
DESPERDICIO TOTAL EN FERTIRRIEGO							0									

### 2.3.2 Simulación con el algoritmo genético (GA)

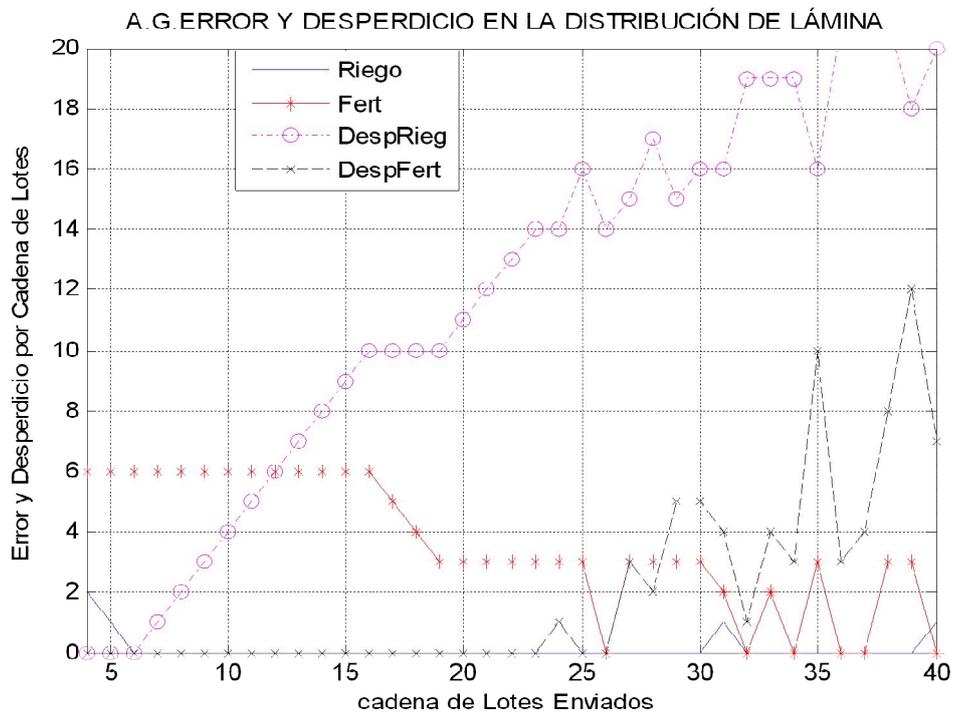
Para este algoritmo se realizaron 5 simulaciones para una de población de 100, 300, 500, 1000, 1500, 3000 individuos. Para cada población se varió el número de generaciones de 80 a 200. Se observó que no se presentaba modificaciones importantes en las respuestas. Por lo tanto se realizaron de nuevo las 5 simulaciones para cada población, con un número de 200 generaciones (Ver Anexo A). De estas simulaciones se seleccionó la mejor mediante los parámetros definidos en la sección 2.3

A continuación se muestra en la Figura 2-19 la gráfica con la mejor solución. Los parámetros del algoritmo para esta simulación fueron los siguientes:

- Tamaño de la población: 1000 individuos
- Número de generaciones: 200
- Función de distribución de mutación utilizada: uniforme 0.8.
- El Tiempo de ejecución que utilizó el algoritmo fue de 9313 segundos.

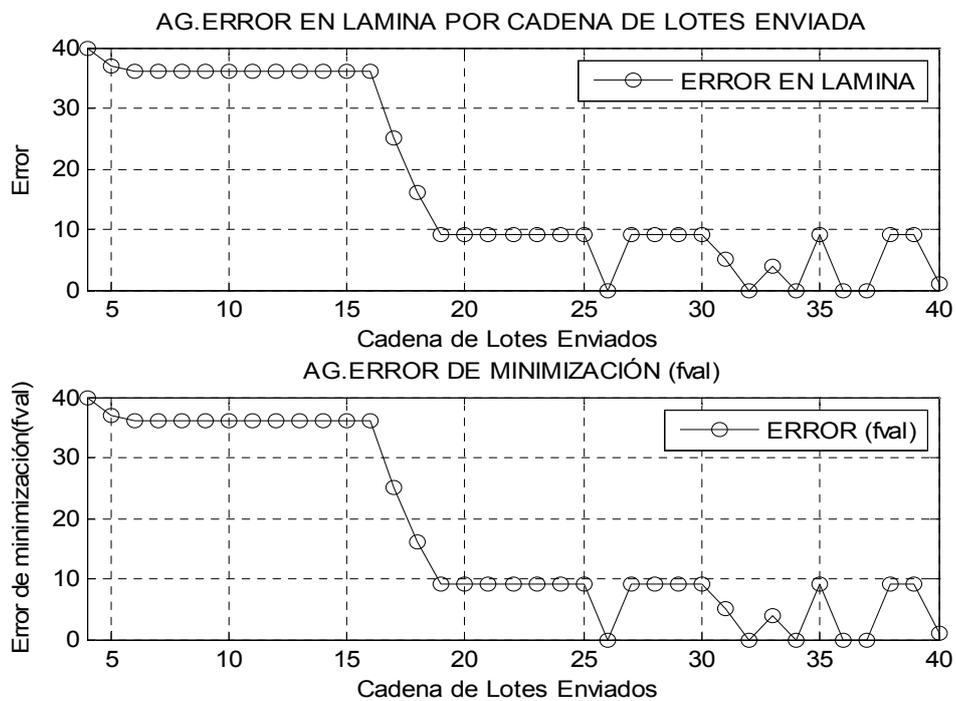
En la figura 2-20 se presenta el comportamiento del error que entrega el algoritmo 2-2 y el error de minimización, por cada cadena de lotes enviada.

**Figura 2-19:** Error y desperdicio en riego y en fertirriego con el GA.



**Figura 2-20:** Error en lámina y error de minimización según cadena de lotes enviados.

(a) Error en el riego y fertirriego. (b) Error resultado de la minimización.



---

### ▪ Interpretación y análisis de la simulación del algoritmo genético

En la Figura 2-19 puede verse el comportamiento que presenta el error para el riego, el fertirriego y el desperdicio; el cual va a un sumidero. Se observa un error de cero (0) en riego y fertirriego cuando se ha enviado una cadena de 26 lotes. Esto significa que con este número de lotes, la lámina programada se cumple. También se aprecia que el desperdicio en fertirriego es cero y que este valor únicamente se presenta para esta cantidad 26 lotes. El desperdicio que se ocasiona en el riego, al enviar la lámina es de catorce (14) lotes.

A continuación, en la Figura 2-20a. Se presenta el comportamiento del error que entrega el algoritmo 2-2. Para la cadena de 26 lotes el error es cero (0). En la figura 2-20b. Se muestra el comportamiento del error de minimización realizado por el GA para el intervalo de 4 a 40 cadenas de lotes. Puede observarse que este error es de cero para la cadena de 26 lotes. El algoritmo 2-2 a partir del vector solución  $x$  que entrega el algoritmo GA, genera la matriz ***DISTR***, el tamaño de esta matriz es de 2 X 26. En la primera fila se encuentra el tipo de insumo que representa cada lote. En la segunda fila las válvulas que deben ser activadas. En la Tabla 2-5 se amplía de una manera más detallada la información que entrega esta matriz.

A continuación se describe la información que presenta las filas 3, 4 y 5 de la Tabla 2-5. La fila tres enumera el orden de la secuencia de activación de las válvulas y la inyección de fertilizante, esto sucede en el orden ascendente que se muestra. La fila cuatro, corresponde a los lotes de riego (0) y fertirriego (1) que se van inyectando a la tubería en el nodo  $V_1$  según la Figura 2-12. La quinta fila, corresponde a las válvulas que se deben de activar para depositar los lotes de riego o fertirriego que tienen a la entrada. Según lo anterior, el evento uno consiste en abrir la válvula **V13** y en el mismo instante inyectar fertilizante. El tiempo de inyección de riego o de fertilizante es el mismo durante el cual la válvula esta activa. Este tiempo se calculó para un lote en la ecuación (2.5).

Los eventos se realizan con la misma lógica desde el segundo evento hasta el evento 26. Nuevamente se pone de manifiesto que la tubería antes de comenzar a distribuirse la lámina debe contener solamente agua.

**Tabla 2-5:** Cadena de 26 lotes para realizar la distribución de la lámina con el AG

RIEGO = 0    FERTIRRIEGO = 1																									
EVENTO																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	1	1	1	1	1	0	1	0	1	1	0	1	1	1	0	0	1	0	1	1	1	1	0	1	1
13	23	23	21	21	14	12	24	13	23	13	12	21	13	12	24	23	13	23	22	14	22	11	11	22	11
VÁLVULAS																									

En la tabla 2-6 se hace un resumen de los errores por válvula y el desperdicio. Puede observarse que para cada válvula de riego, se lista la cantidad de lotes que llegan (lot acumul). Los lotes que le han sido programados (Lote Prog), y la diferencia entre los lotes programados y los acumulados, que es el error (Err). Por ejemplo, para la válvula **V11**, se han programado 3 lotes de riego, han llegado a ella 3, por lo tanto el error es cero (0), y la lámina se ha cumplido para esta válvula. Y lo mismo ocurre con la válvula **V21**. Las **V12** y **V22** programadas con fertirriego, también tiene el mismo resultado.

Para las válvulas de desperdicio se muestra la cantidad de lotes de riego y fertirriego que desalojan. De la Tabla 2-6, puede observarse que las válvulas **V13**, **V14**, **V23** y **V24** se han activado 5, 2, 5 y 2, veces respectivamente. Las aperturas de estas válvulas suman 14. Este es el número de lotes de desperdicio de riego. Se verifica lo mostrado en la Figura 2-7. El error total en riego es cero (0) en fertirriego es cero (0), en desperdicio de fertirriego es cero (0) y en desperdicio de agua quince (14) lotes.

Se observa que al igual que para el CS, las válvulas 14 y 24 participan en el desalojo de desperdicio de riego, y nunca van a desalojar desperdicio de fertirriego para este número de lotes.

**Tabla 2-6:** Resumen de la distribución entregada por el AG

Vál	V11			V12			V21			V22			V13	V14	V23	V24
Dat	Lot acu mul	Lote Prog	Err	Lote acu mul	Lote Prog	Err	Lot acu mul	Lote prog	Err	Lote acu mul	Lote prog	Err	Lot de Des	Lote de Des	Lote de Des	Lote de Des
Fer tiri ego	0	0	0	3	3	0	3	3	0	0	0	0	0	0	0	0
Ri- Eg o	3	3	0	0	0	0	0	0	0	3	3	0	5	2	5	2
ERROR TOTAL EN RIEGO.									0							
ERROR TOTAL EN FERTIRRIEGO									0							
DESPERDICIO TOTAL EN RIEGO									14							
DESPERDICIO TOTAL EN FERTIRRIEGO									0							

### 2.3.3 Simulación con el algoritmo optimización por enjambre de partículas (PSO).

Para este algoritmo se realizaron 5 simulaciones para una población de 100, 300 500 1000 y 3000 individuos (ver el anexo B). Se varió el número de épocas y el tamaño de la población en un amplio espectro, pero el resultado de la simulación no mostro un cambio significativo en la respuesta.

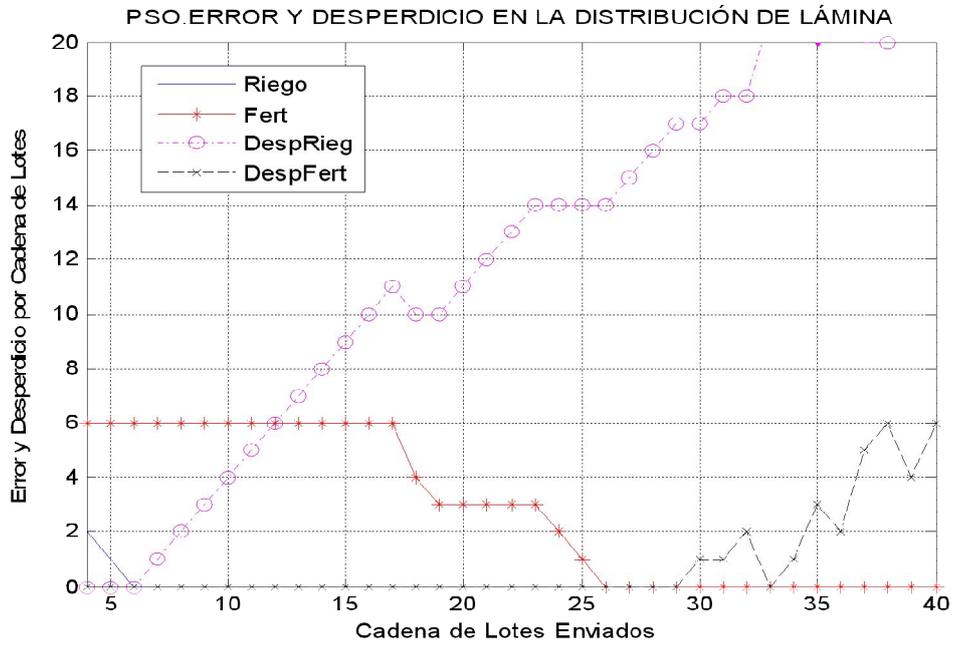
De las cinco simulaciones por población (ver anexo B) se seleccionó la mejor, ver Figura 2-21. Para esta selección se emplearon los criterios definidos en la sección 2.3. Cada respuesta de la simulación se presenta con tres gráficas. La primera, el error y el desperdicio en riego y fertirriego. La segunda, el error hallado por la función de aptitud para cada envío de lotes. La tercera, el comportamiento del error para cada época.

Para la simulación seleccionada se emplearon los siguientes parámetros:

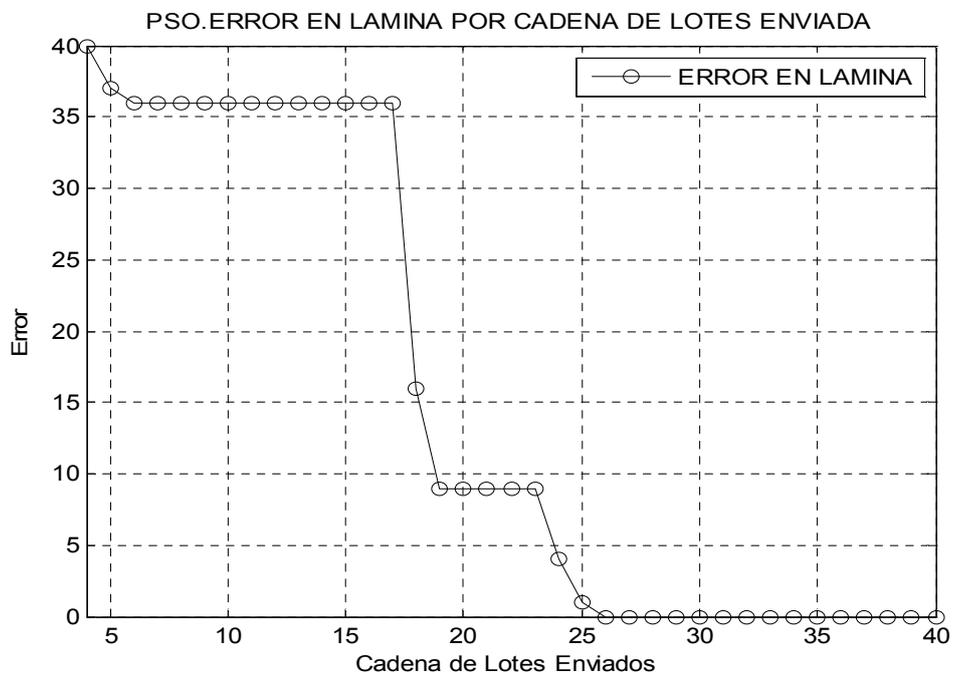
- Épocas: 200
- Tamaño de la población: 300 individuos.

El Tiempo de ejecución promedio que utilizo el algoritmo fue de 5300 segundos.

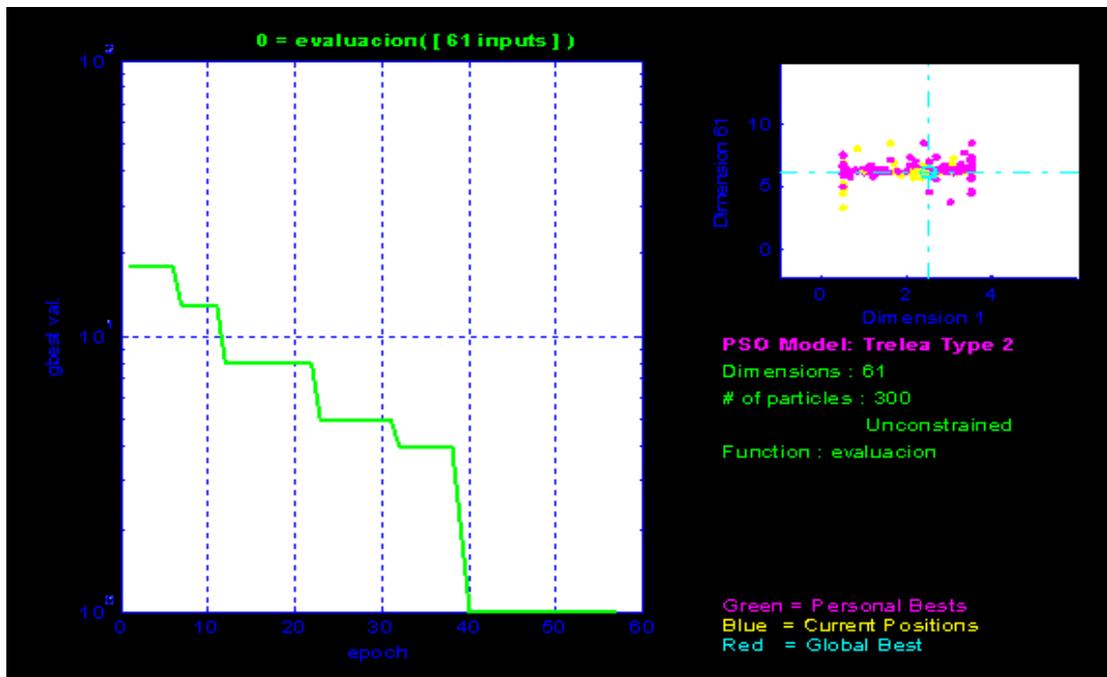
**Figura 2-21:** Error y desperdicio en riego y en fertirriego con el PSO



**Figura 2-22:** Error calculado por la función de aptitud para cada envió de lotes con el PSO



**Figura 2-23:** Tendencia del mejor valor para cada época.



#### ▪ Interpretación y análisis de la simulación con el PSO

En la Figura 2-21, puede verse los valores del error para el riego y el fertirriego y la cantidad de desperdicio. Se observa un error de cero (0) en riego y fertirriego cuando se ha enviado una cadena de 26 lotes. Con este número de lotes enviados, la lámina programada se cumple. También puede observarse que el desperdicio en fertirriego es cero (0) y el desperdicio en el riego es de catorce (14) lotes.

En la Figura 2-22. Se presenta el comportamiento del error que entrega el algoritmo 2-2, para la cadena de 26 lotes. El error es cero (0). Esto es cierto, por que con este número de lotes la lámina se cumple. Sin embargo también es cero para valores superiores a 26 lotes, en la Figura 2-21 puede observarse que para estos valores el desperdicio se incrementa.

En la Figura 2-23. Se muestra como es la tendencia del mejor valor para cada época. Se observa que los mejores valores comienzan en 40 épocas. Para una población es de 300 partículas la dimensión en que se mueven para la búsqueda las partículas es de 61.

El algoritmo 2-2 a partir del vector solución  $x$  que entrego el algoritmo *PSO*, genero la matriz ***DISTR***. El tamaño de esta matriz para la simulación seleccionada, es de 2 X 26. En la primera fila de la matriz ***DISTR*** se expresan los insumos. En la segunda fila la válvula que debe ser activada. En la Tabla 2-7 se amplia de manera más detallada la información que entrega esta matriz.

A continuación se describe la información que presenta las filas 3, 4 y 5 de la Tabla 2-7. La fila tres enumera el orden de la secuencia de activación de las válvulas y la inyección de fertilizante. La fila cuatro, corresponde a los lotes de riego (0) y fertirriego (1) que se van inyectando a la tubería en el nodo  $V_1$  según se muestra en la Figura 2-12. La quinta fila, corresponde a las válvulas que se deben de activar para depositar los lotes de riego o fertirriego que tienen a la entrada. Según lo anterior, el evento uno consiste en abrir la válvula **V13** y en el mismo instante inyectar fertilizante. El tiempo de inyección de riego o de fertilizante, es el mismo tiempo durante el cual la válvula esta activa. Este tiempo se calculó para un lote en la ecuación (2.5).

Los eventos se realizan con la misma lógica desde el segundo evento hasta el evento 26. Se pone nuevamente de manifiesto que la tubería antes de comenzar a distribuirse la lámina, debe contener solamente agua.

**Tabla 2-7:** Cadena de 26 lotes para realizar la distribución de la lámina de riego con el *PSO*

RIEGO = 0    FERTIRRIEGO = 1																									
EVENTO																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	1	1	0	0	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	0
13	13	13	24	14	23	21	21	23	13	23	13	12	12	12	14	11	11	23	23	11	21	23	22	22	22
VÁLVULAS																									

---

En la Tabla 2.8 se hace un resumen de los insumos programados: los que cumplen con la lámina y son depositados, los desalojados por las válvulas como desperdicio y los errores ocasionados. Se verifica lo mostrado en la Figura 2-21. El error total en riego es cero (0) en fertirriego es cero (0), en desperdicio de fertirriego es cero (0) y en desperdicio de agua catorce (14) lotes desalojados a través de las válvulas **V13, V14, V23, V24**.

En la Tabla 2-8. Puede observarse que para cada válvula de riego se lista la cantidad de lotes que llegan (Lotes acum), los lotes que le han sido programados (Lotes Prog), y la diferencia entre los programados y los acumulados, que es el error (Err). Por ejemplo, para la válvula **V11**, se han programado 3 lotes de riego, han llegado a ella 3, por lo tanto el error es cero (0), la lámina se ha cumplido para esta válvula. Lo mismo ocurre con la válvula **V21**. Las **V12** y **V22** programadas con fertirriego presentan el mismo resultado.

De la Tabla 2-8 puede observarse que las válvulas **V13, V14, V23** y **V24** se han activado 5, 2, 6 y 1, veces respectivamente. Las aperturas de estas válvulas suman 14, que es el número de lotes de desperdicio de riego. Se verifica lo mostrado en la figura 21. El error total en riego es cero (0) en fertirriego es cero (0), en desperdicio de fertirriego es cero (0) y en desperdicio de agua catorce (14) lotes. Se observa que las válvulas 14 y 24 participan en el desalojo de desperdicio de riego, como ocurrió con la solución del GA y del CS, pero no llegan a desalojar desperdicio de fertirriego para esta programación. En estas condiciones estas válvulas pueden desde el programador de riego, programarse como una sola válvula o como coválvulas. Permitiendo de esta forma una mayor flexibilidad en la operación de la red.

**Tabla 2-8:** Resumen de la distribución de la lámina de riego fertirriego programada con el *PSO*.

Vál	V11			V12			V21			V22			V13	V14	V23	V24
	Lot acu mu	Lote Pro g	Err	Lote acu mul	Lot Pro g	Err	Lot acu mul	Lote pro g	Err	Lote acu mul	Lote pro g	Err	Lot de Des	Lote de Des	Lote de Des	Lote de Des
Fer tiri ego	0	0	0	3	3	0	3	3	0	0	0	0	0	0	0	0
Ri Eg o	3	3	0	0	0	0	0	0	0	3	3	0	5	2	6	1
ERROR TOTAL EN RIEGO.									0							
ERROR TOTAL EN FERTIRRIEGO									0							
DESPERDICIO TOTAL EN RIEGO									14							
DESPERDICIO TOTAL EN FERTIRRIEGO									0							

## 2.4 Comparación de los resultados entre las estrategias evolutivas

En la Tabla 2-9 se realiza una comparación entre las diferentes estrategias evolutivas utilizadas para enviar la lámina programada. Esta comparación se realiza con respecto al número de lotes a enviar para cumplir con la lámina, a los errores en riego y fertirriego, al desperdicio generado y al tiempo utilizado para realizar una simulación.

En la tabla 2-9 puede verse que el algoritmo *GA* y *PSO* entregan una mejor solución al realizar la distribución con 26 lotes y el *CS* con 27, la diferencia es mínima y lo mismo ocurre para el desperdicio. Estos valores tan próximos indican que las técnicas de los algoritmos evolutivos son aptos encontrar un solución a la distribución de una lámina de riego programada. Según la ecuación (2.4), en términos de volumen de riego, 14 lotes representan  $3.57 \text{ m}^3$  de riego (agua), los cuales regresan al reservorio.

**Tabla 2-9:** Comparación de resultados entre las estrategias evolutivas *GA*, *PSO*, *CS*

Estrategia Evolutiva	Cantidad de lotes enviados	Error en riego	Error en fertirriego	Desperdicio en fertirriego	Desperdicio en riego	Tiempo de ejecución en segundos
Algoritmo Genético ( <i>GA</i> )	26	0	0	0	14	6500
Enjambre de Partículas ( <i>PSO</i> )	26	0	0	0	14	5700
Busqueda <i>Cuckoo</i> ( <i>CS</i> )	27	0	0	0	15	6100



## **3. Discusión y Conclusiones**

### **3.1 Aporte de la metodología propuesta**

El problema tratado en este trabajo acerca de cómo realizar la distribución de una lámina programada de riego y fertirriego mediante un sistema automático, es frecuente en muchos cultivos y su origen tiene mucho que ver con la red hidráulica utilizada. Es común encontrar que la prioridad generada por órdenes de producción imprevistas en los cultivos, ocasione que áreas con plántulas en siembra, crecimiento vegetativo o poscosecha, vea modificada su programación y desarrollo. Por ejemplo, tener que dejar áreas libres de siembra, cosechar antes de tiempo y preparar terreno para una nueva siembra, etc. Estas situaciones generan muchos contratiempos para hacer el riego y el fertirriego utilizando un controlador/programador de riego [2, 3]. Implementar una red que tenga doble tubería da una solución parcial al problema de la distribución, garantiza oportunidad y calidad en la lámina entregada, pero no lo hace para el desperdicio y para los costos que demandan su construcción y operación. Utilizar una red hidráulica de lazo abierto tiene una implementación menos costosa y por lo tanto es la más comúnmente utilizada, sin embargo, si se utiliza una estrategia empírica de programación, ocasiona también grandes desperdicios en agua y fertilizante y la estrategia de distribución se hace más complicada.

Durante el diseño de una red de lazo abierto es común no tener en cuenta si los cultivos tienen o no estandarizada la disposición y planificación de las áreas para la siembra ante las contingencias y órdenes de producción urgentes. Situación que se convierte más problemática cuando sea extendido y adoptado el concepto de aplicar dosis de riego y fertirriego frecuentemente, lo que genera un alto tráfico de estos insumos durante el día. A lo anterior se le suma la necesidad de tener la red disponible parte de la jornada de trabajo en el cultivo para distribuir agua solamente, la cual es utilizada en otras actividades en el cultivo como es la presiembra y la preparación del terreno [1]. El tiempo y la logística que se destina a la preparación y ejecución de una jornada de fertirriego es alto y costoso. La automatización del riego y fertirriego en estas condiciones es un fracaso y la falta de una estrategia que realice de una forma versátil y eficaz la distribución del riego y el fertirriego para las condiciones anteriormente mencionadas hace que los sistemas automáticos implementados se desmonten o que los que se

deseen implementar se descarten [50]. La falta de literatura disponible asociada al problema que se ha tratado en este trabajo aumenta las limitaciones para abordar una solución, sin embargo los cultivos que las han enfrentado han dado soluciones de contingencia de acuerdo a la situación particular. Eventos que por lo demás no terminan documentándose.

Con la metodología desarrollada en este trabajo se da una solución para la distribución del riego y el fertirriego automatizado en un cultivo, cuando se tiene una red hidráulica en lazo abierto las condiciones de operación son generadas por órdenes de producción imprevistas, que ocasionan variación en la programación en la siembra, en el crecimiento vegetativo y en la cosecha.

La gestión del riego y fertirriego mediante la implementación de esta metodología tiene el potencial de cambiar radicalmente las costumbres en la preparación de la logística para hacerlo, porque no se tendrán que realizar maniobras engorrosas que implican utilizar más tiempo en la labor, cambios temporales en la red hidráulica y por lo tanto aumento de contingencias en ella. El tiempo de programación será mucho menor debido que no es necesario realizar complicados cálculos, este dependerá del tiempo en que el sistema de cómputo se tarde y se obtendrá una estrategia de distribución inmediata y óptima. El desperdicio de fertilizante será mínimo, y permitirá que sean planeados intervalos de tiempo para que la red hidráulica esté disponible sólo para agua y por lo tanto puedan ejecutarse otras labores en el cultivo para los cuales ésta es imprescindible [50].

Además, el presente documento es una fuente de nueva información a disposición de la comunidad científica y técnica que se dedica a las actividades del riego y el fertirriego.

## **3.2 Conclusiones**

En esta tesis de maestría se demostró que las estrategias de optimización meta-heurísticas y en concreto los algoritmos genéticos, la optimización por enjambres de partículas y la búsqueda *cuckoo*, son herramientas con las que puede encontrarse una solución óptima al problema de distribuir una lámina programada de riego y fertirriego a un cultivo con cero error (0) y un desperdicio muy pequeño.

Mediante la teoría de grafos se puede modelar matemáticamente el cultivo, obteniendo una matriz de nodos y “lotes” que exprese el recorrido y los volúmenes de almacenamiento de insumos en la tubería en una red hidráulica de lazo abierto. Se concluye que esta matriz es el insumo principal para una función de aptitud que necesite reconstruir la red y a través de ella pueda evaluar las soluciones de distribución de una lámina que una meta-heurística proponga.

Es acertada la estrategia implementada para enviar una lámina compuesta por agua y fertilizante de dividirla en valores discretos (0 y 1), que se definieron como “lotes”. Planteándose un problema combinatorial de optimización que implicó buscar la secuencia óptima para el envío de estos lotes. Se concluyó que al obtenerse una combinación mínima de lotes y de válvulas a activar, se entregaría la lámina programada con un mínimo error y desperdicio.

Las herramientas utilizadas para darle aplicación a la estrategia de los “lotes” fueron los algoritmos genéticos, la optimización por enjambre de partículas y la búsqueda *cuckoo*, entregando excelentes resultados. Se concluye que los algoritmos evolutivos utilizados son una herramienta que puede utilizarse para encontrar las combinaciones adecuadas que ofrezcan soluciones a la distribución de una lámina de riego

Se construyó una función objetivo que tuvo éxito en la evaluación de la aptitud de las combinaciones enviadas por los algoritmos evolutivos, determinando el error y el desperdicio ocasionado.

Para validar esta metodología es necesario realizar varias simulaciones con cada algoritmo evolutivo, debido a que por su característica de búsqueda global, encuentra varios óptimos globales, todos ofreciendo soluciones válidas.

Se concluye que no puede asegurarse que una de las tres herramientas sea la mejor para la solución de problemas de distribución de riego, porque al comparar los valores entregados por las mejores simulaciones del CS el AG y el PSO se encontró que eran similares o con una diferencia mínima. Esto demuestra que la metodología utilizada es robusta y no depende de la herramienta utilizada.

La validación de la metodología propuesta contra una estrategia de distribución empírica (convencional) se descarta debido a que la solución empírica no provee un procedimiento metódico para las situaciones problemáticas para un cultivo planteadas en este trabajo y los resultados varían necesariamente con la experticia del programador. Se concluye que el objetivo de encontrar una solución a la distribución de una lámina mediante técnicas evolutivas no pretende superar la solución encontrada por un programador empírico experimentado que por sí misma no es apropiada, sino la de proponer una solución metodológica a este problema que permita su automatización.

### **3.3 Trabajo futuro**

Se sugiere profundizar en el tema con nuevos trabajos que apliquen esta metodología, considerando variaciones en la red y en el cultivo, como es la de ampliar a un mayor número de válvulas y enviar un mayor número de "lotes". Esto ocasiona un problema combinatorial de mucha mayor complejidad.

Es necesario incluir en nuevos trabajos otras restricciones, como es la de minimizar la cantidad de fertirriego que pueda quedar en la tubería después de que se ha enviado la totalidad de la lámina.

También pueden ser aplicados otros algoritmos meta-heurísticos a la solución del problema, tales como las colonias de hormigas o diferentes variaciones de las meta-heurísticas usadas.

Finalmente, un producto final que puede obtenerse a partir de lo que se los resultados de este trabajo, es el de incorporar en controladores/programadores de riego, una función que determine cómo se debe realizar la distribución de una lámina de riego y fertirriego y permita su ejecución automatizada.

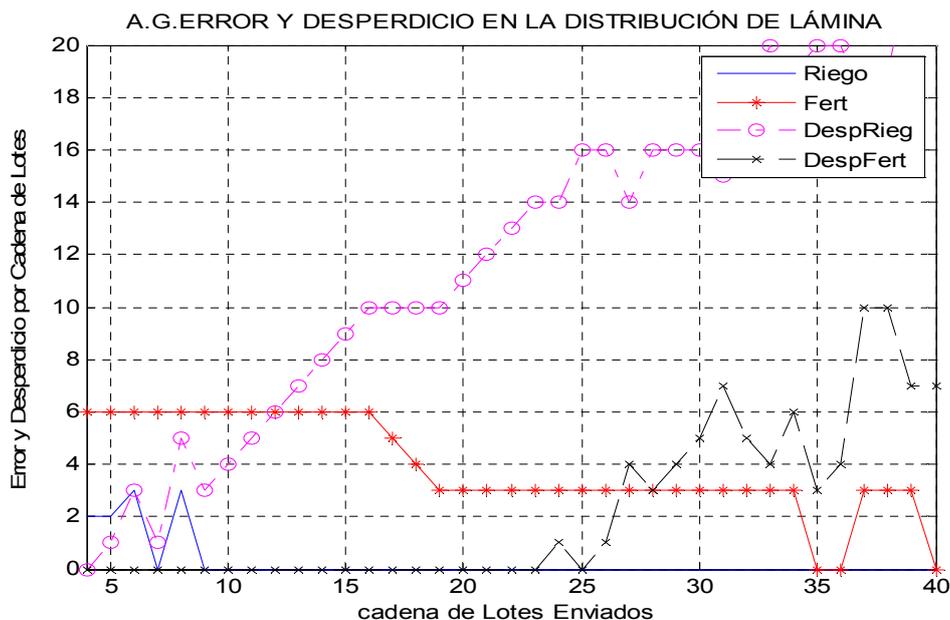


## A. Anexo: Simulaciones realizadas con el Algoritmo Genético.

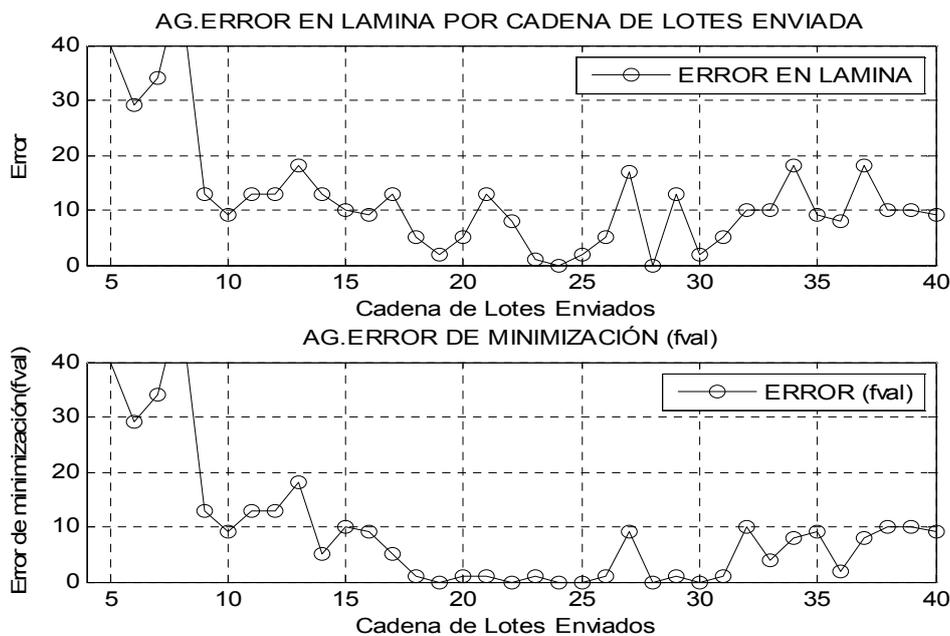
Las siguientes gráficas corresponden a las cinco simulaciones seleccionadas del conjunto de 25 realizadas con el algoritmo *ruta* (Algoritmo 2-1), el algoritmo *eval* (Algoritmo 2-2) y el algoritmo genético, para diferentes valores de población y generaciones.

- Los parámetros para realizar la simulación de la figura A-24 y A-25 son los siguientes:
- Población: 300 individuos
- Generaciones: 200

**Figura A-24:** Simulación 1 AG con Población 300 individuos; Generaciones 200

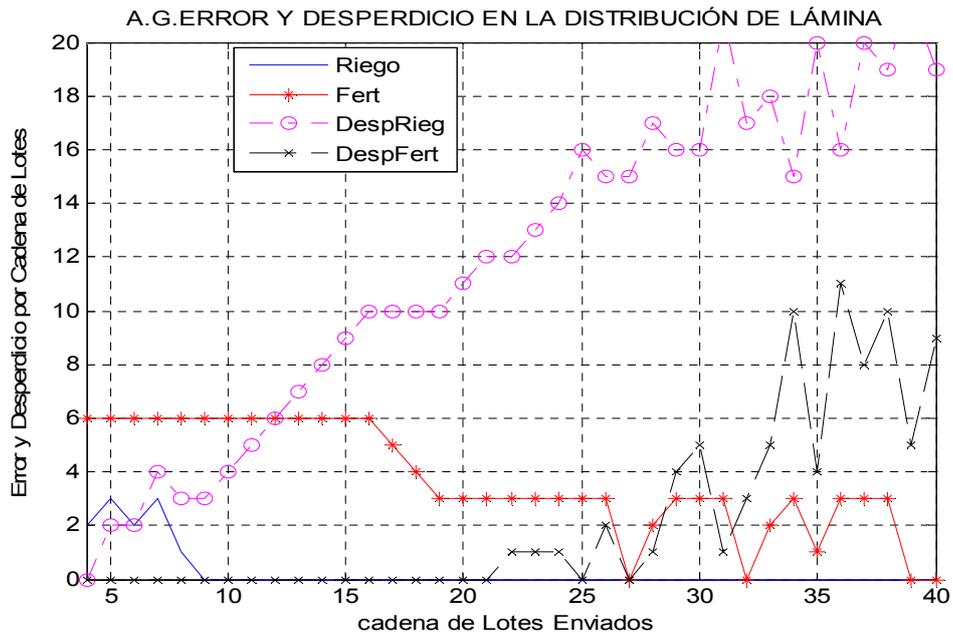


**Figura A-25:** AG Errores lámina y error algoritmo *eval*

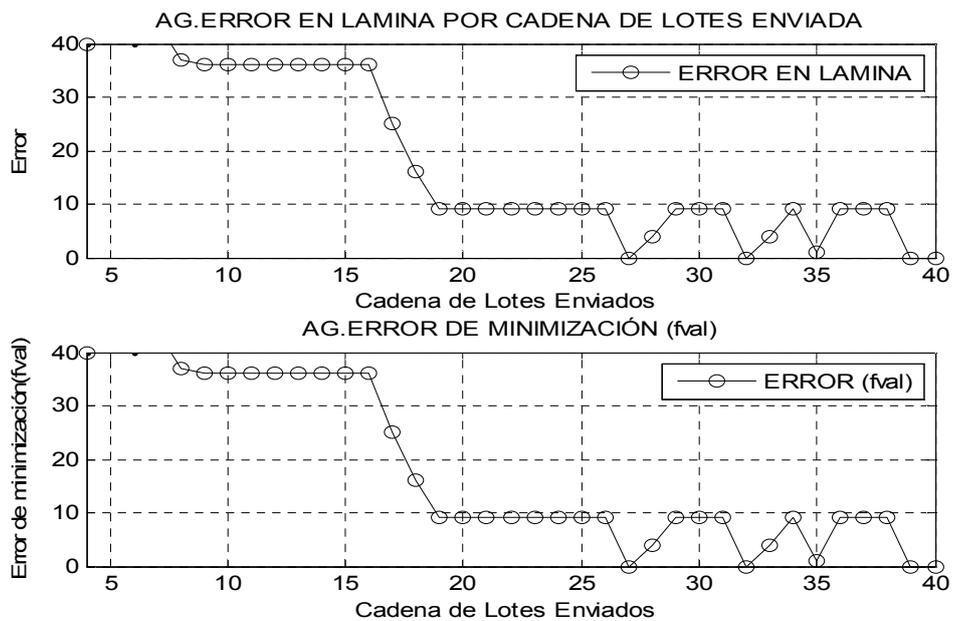


- Los parámetros para realizar la simulación de la figura A-26 y A-27 son los siguientes:
- Generaciones = 200;
- Población = 500.

**Figura A-26:** Simulación 2. GA con Población 500 individuos; Generaciones 200

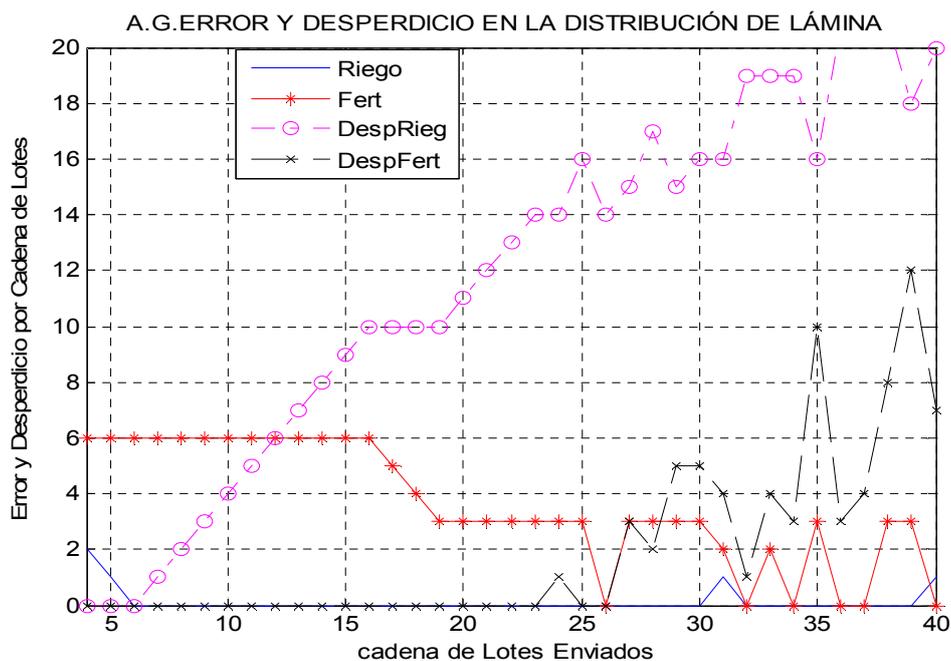


**Figura A-27:** A. G Errores lámina y error *eval*

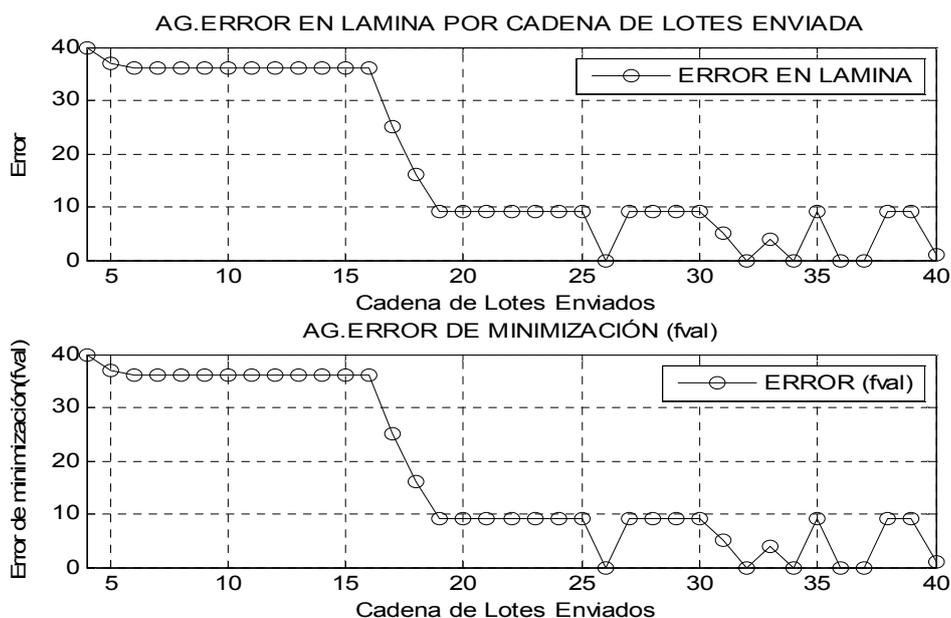


- Los parámetros para realizar la simulación de la figura A-28 y A-29 son los siguientes:
- Generaciones =200;
- Población =1000 individuos;

**Figura A-28:** Simulación 3. AG con Población 1000 individuos; Generaciones 200

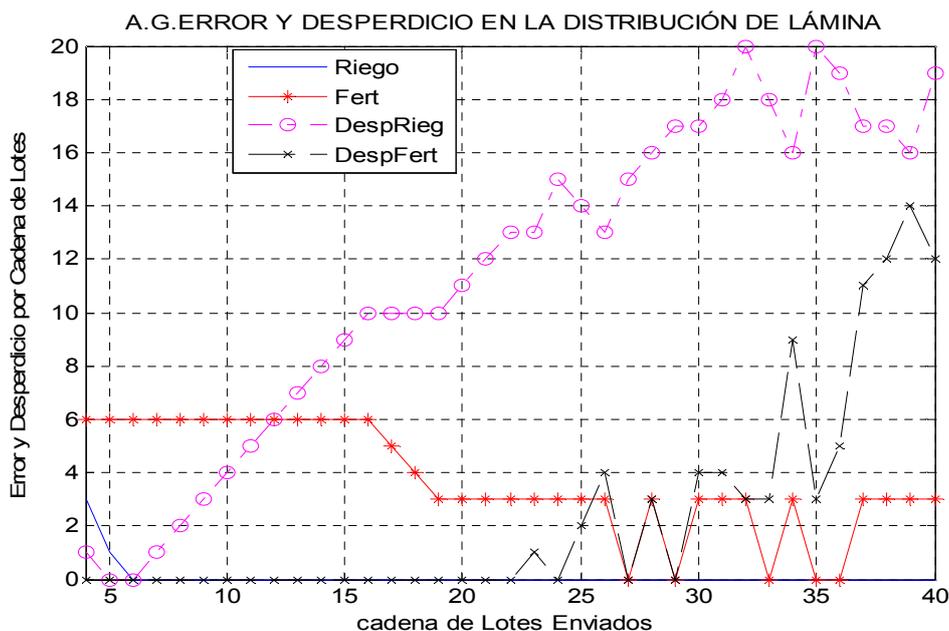


**Figura A-29:** A. G Errores lámina y error *eval*

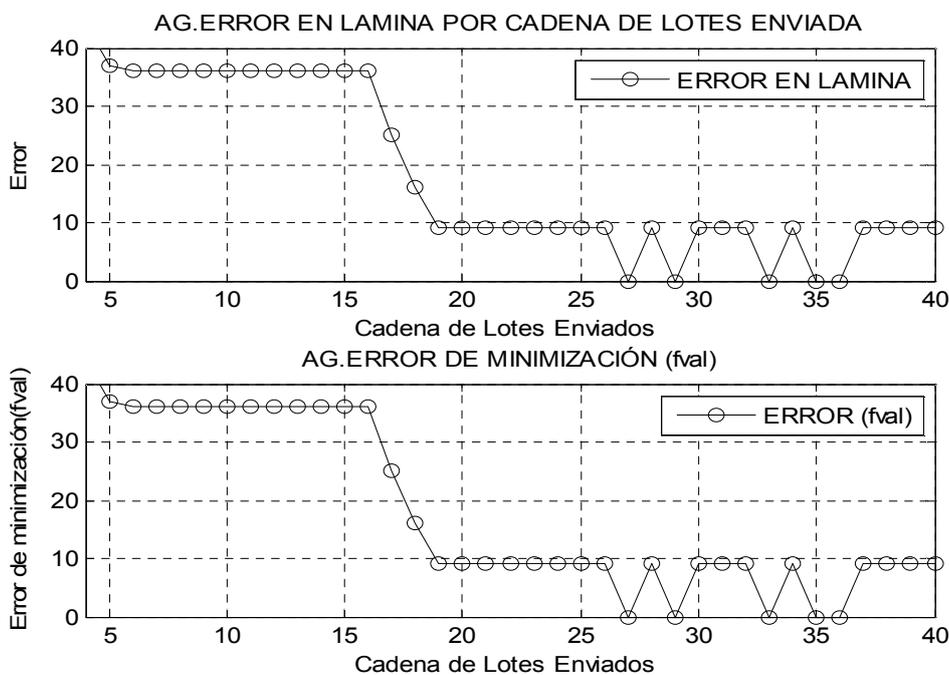


- Los parámetros para realizar la simulación de la figura A-30 y A-31 son los siguientes:
- Generaciones =200;
- Población =1500;

**Figura A-30:** Simulación AG. Población 1500 individuos, Generaciones 200

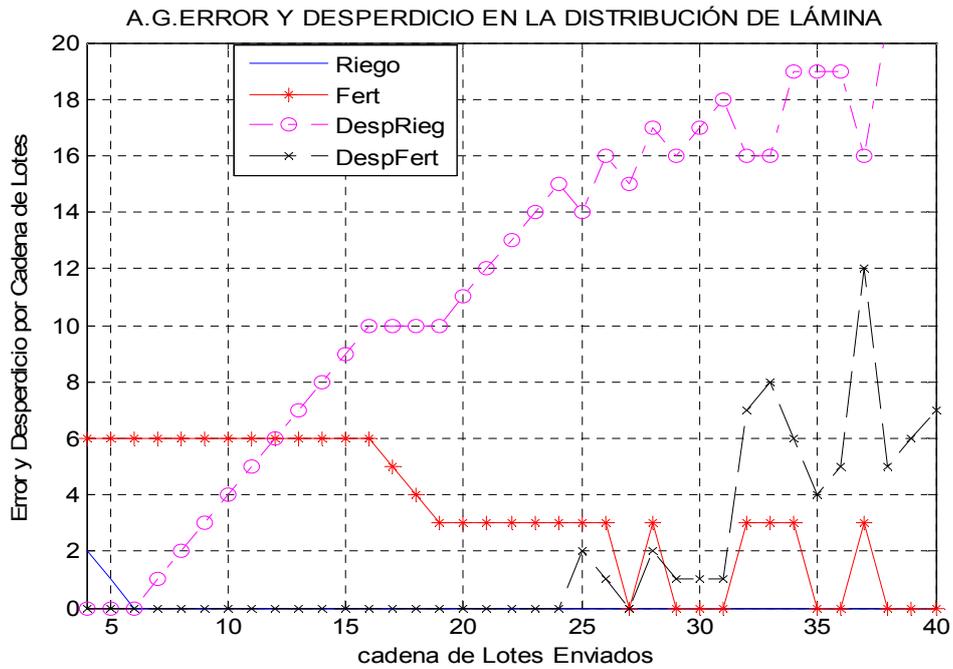


**Figura A-31:** AG Errores lámina y error *eval*

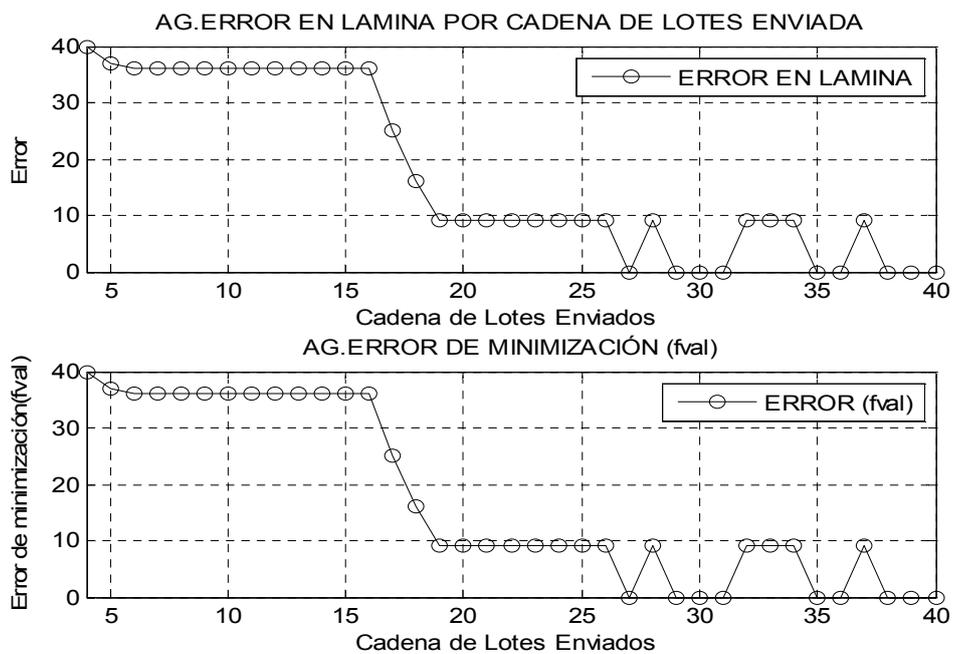


- Los parámetros para realizar la simulación de la figura A-32 y A-33 son los siguientes:
- Generaciones=200;
- Población=3000;

**Figura A-32:** Simulación 5. GA con Población 3000 individuos; Generaciones 200



**Figura A-33:** AG Errores lámina y error *eval*



## B. Anexo: simulaciones con el PSO

Las siguientes gráficas corresponden a las cinco simulaciones seleccionadas del conjunto de 25 realizadas con el algoritmo *ruta* (Algoritmo 2-1), el algoritmo *eval* (Algoritmo 2-2) y el algoritmo *PSO*, para diferentes valores de épocas y población

Los parámetros utilizados para simular las gráficas de la Figura B-34, Figura B-35 son los siguientes:

- Épocas: 200
- Población: 100

**Figura B-34:** Error y desperdicio *PSO* Épocas: 200; Población: 100

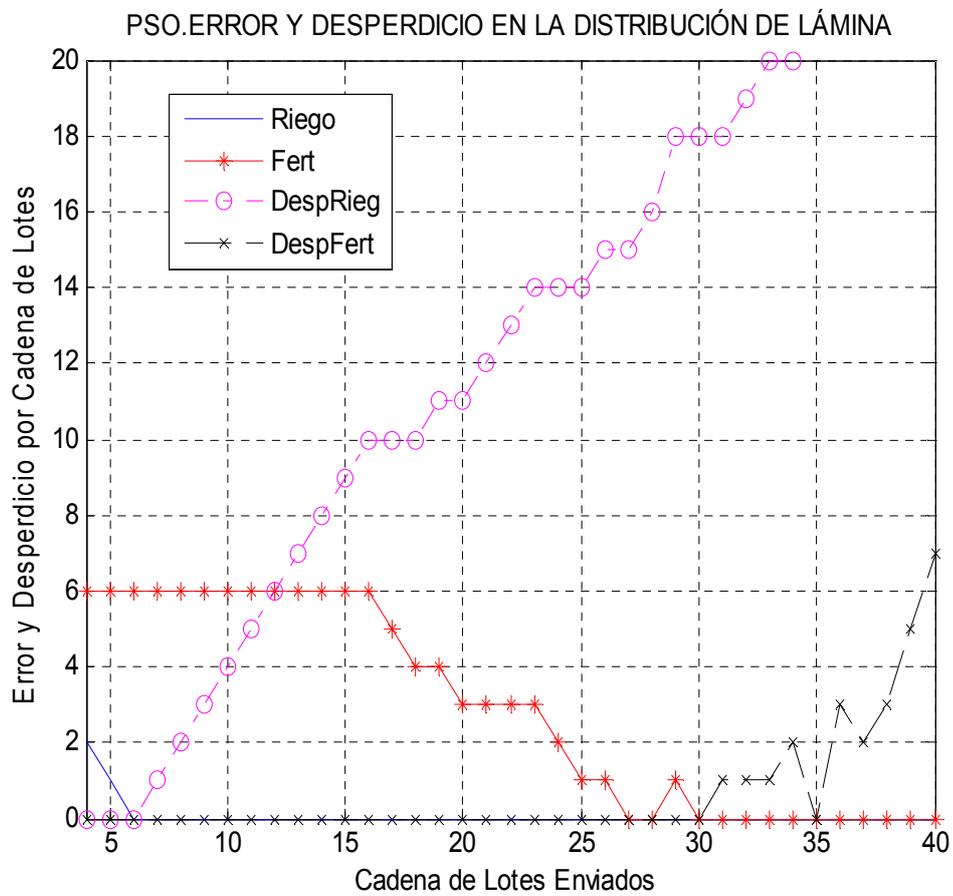


Figura B35: Error en riego y fertirriego

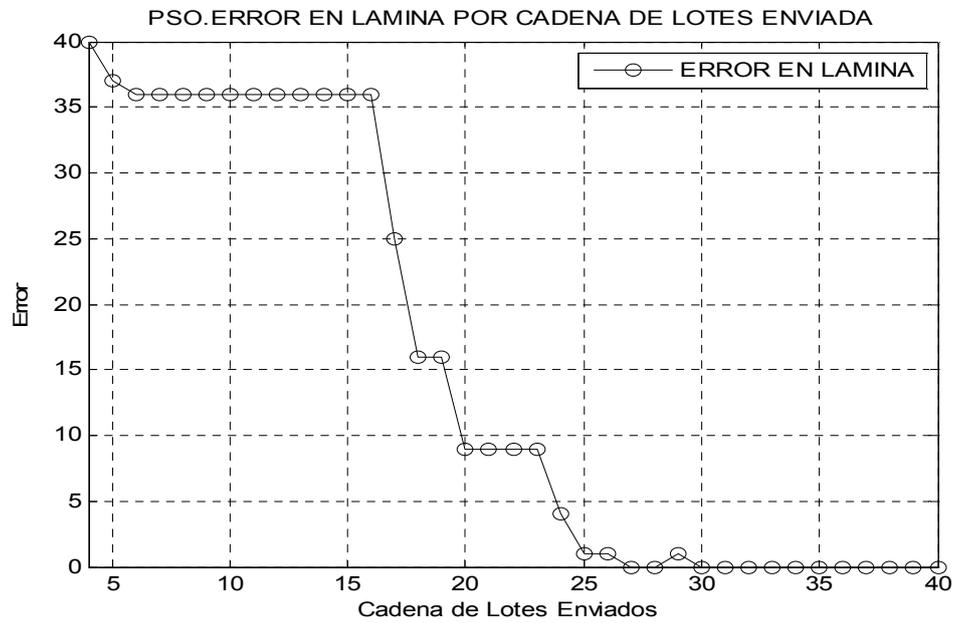
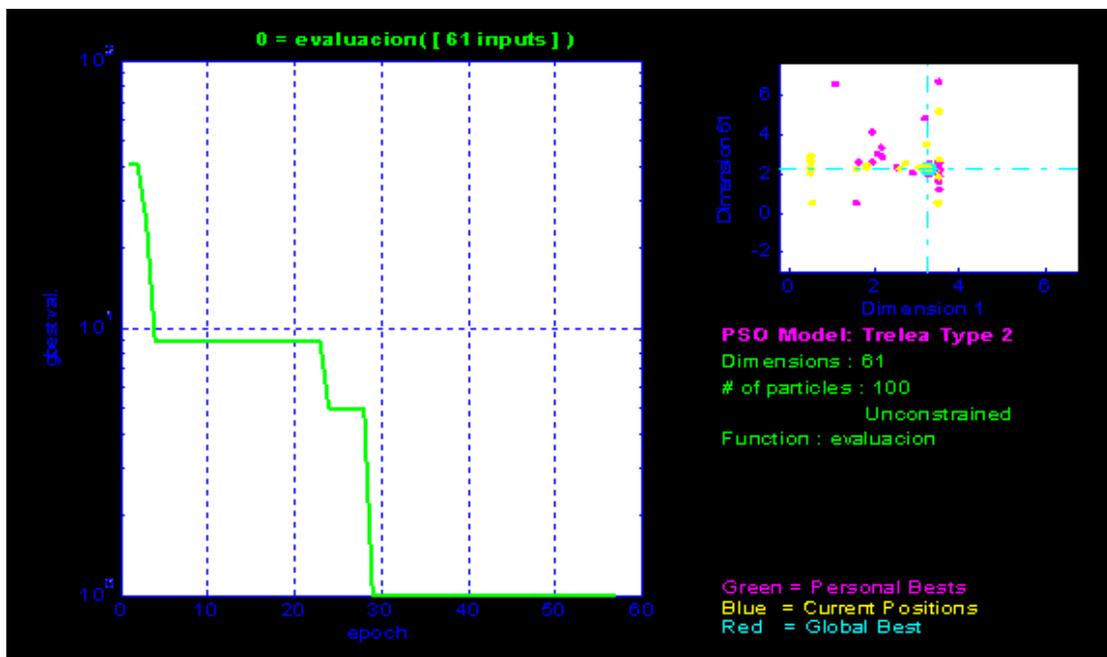


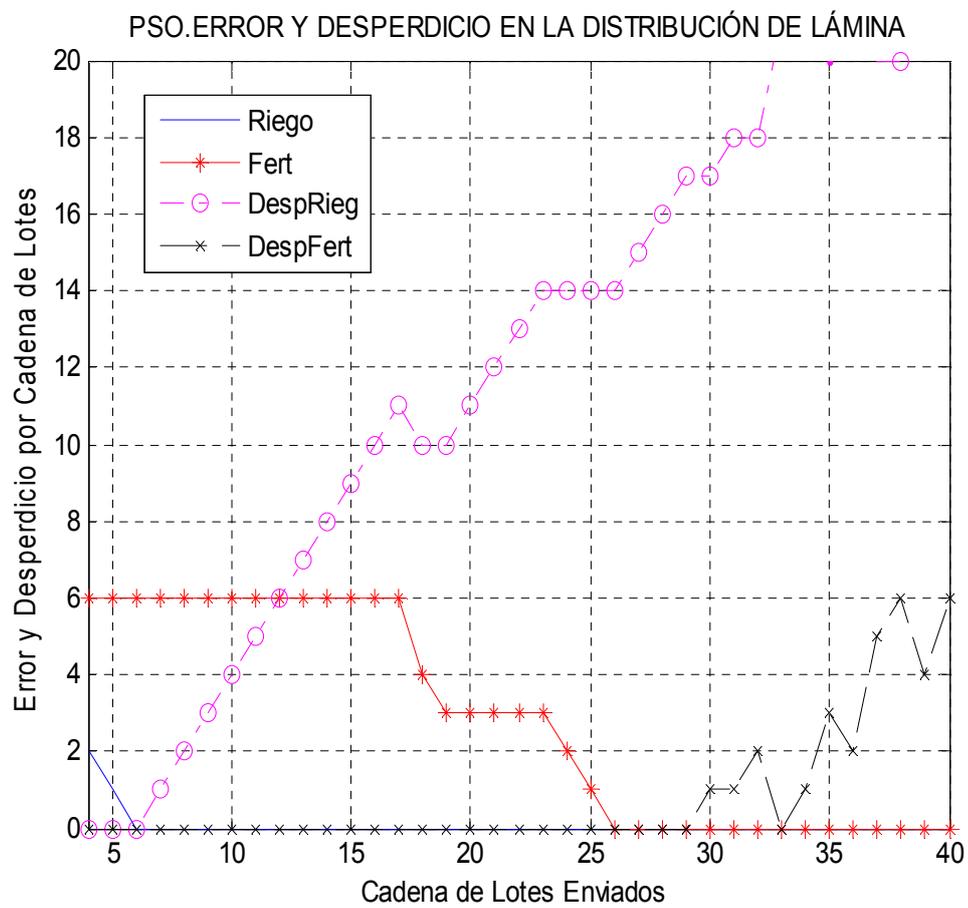
Figura B-36: Error de minimización PSO Épocas: 200; Población: 100



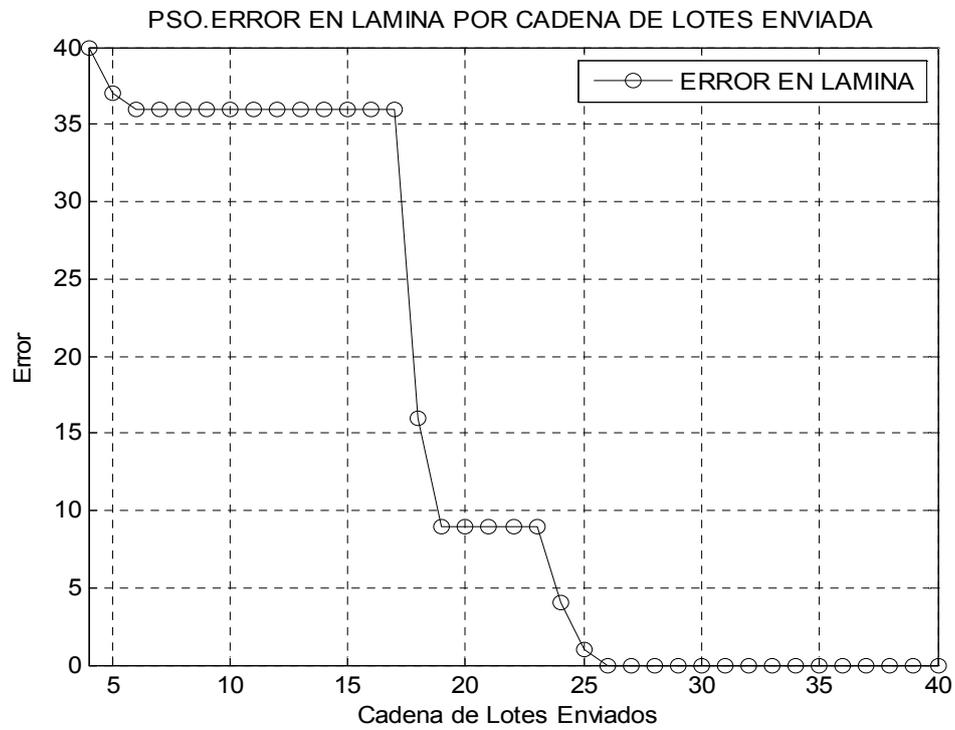
Los parámetros utilizados para simular las gráficas de la Figura B-37, Figura B-38 y la Figura B-6 son los siguientes:

- Épocas: 200.
- Población: 300

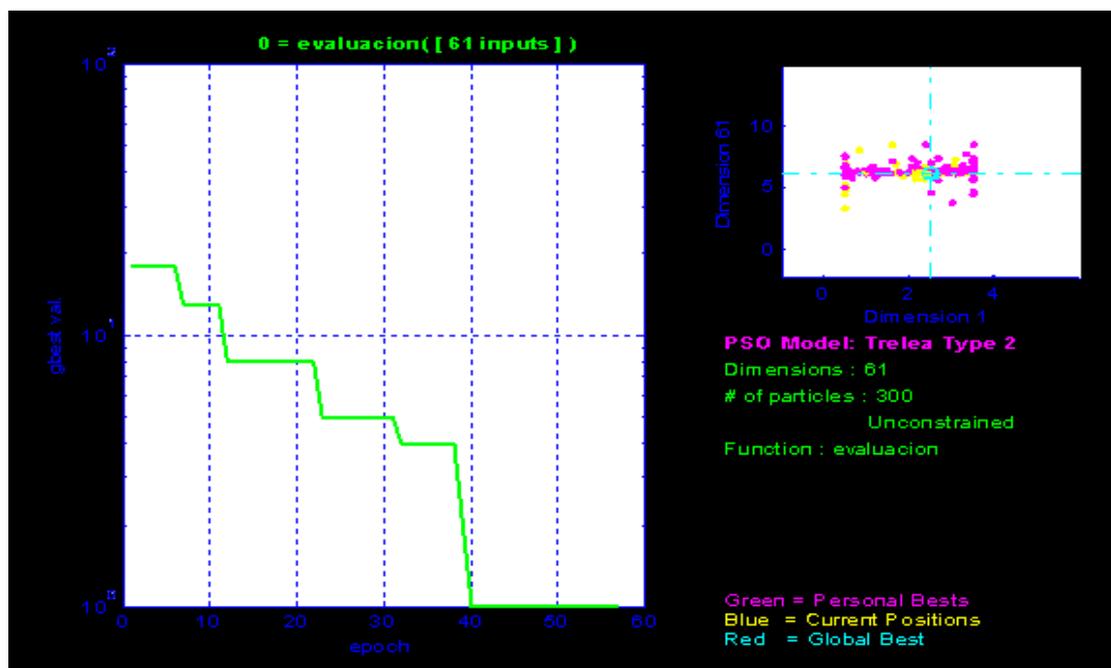
**Figura B-37:** Error y desperdicio Épocas: 200. Población: 300



**Figura B-38:** Error en riego y fertirriego



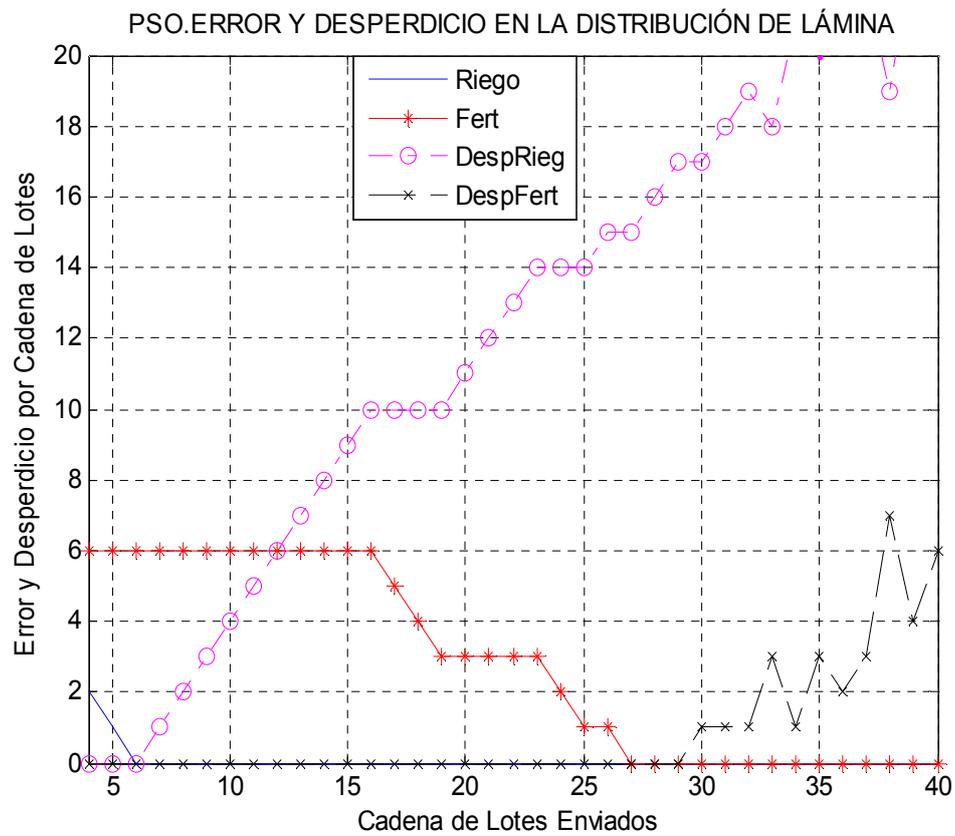
**Figura B-39:** Comportamiento del error de minimización PSO



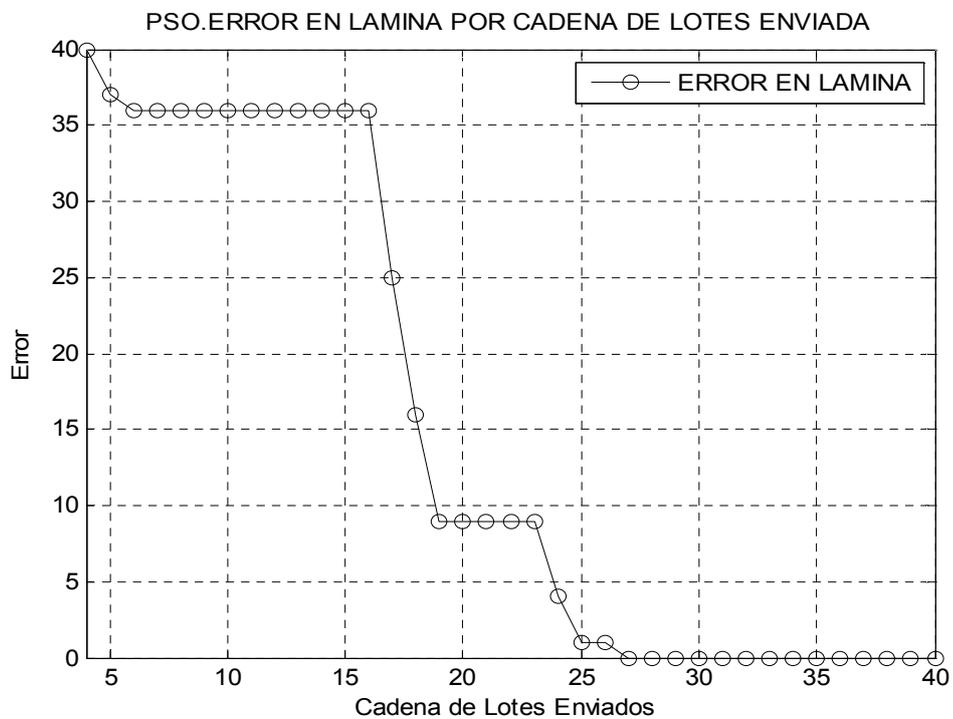
Los parámetros utilizados para simular las gráficas de la Figura B-40, Figura B-50 y la Figura B-51 son los siguientes:

- Épocas = 200
- Población = 500

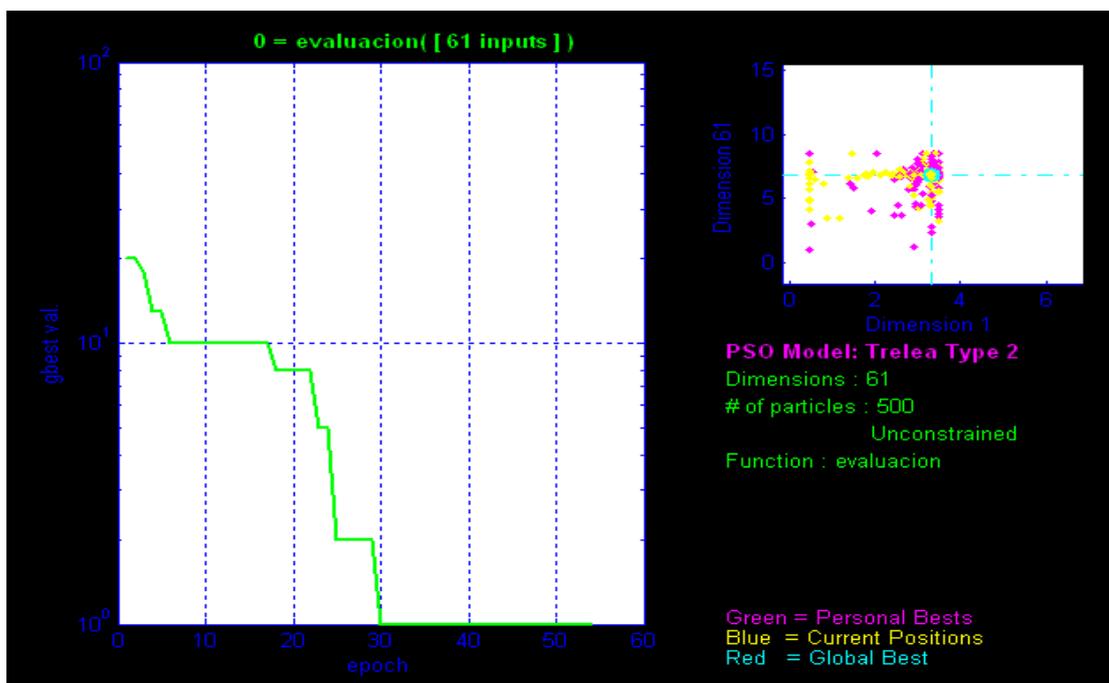
**Figura B-40:** Error y desperdicio Épocas 200; población 500



**Figura B-41:** Erro en riego y fertiriego Épocas: 200; Población: 100



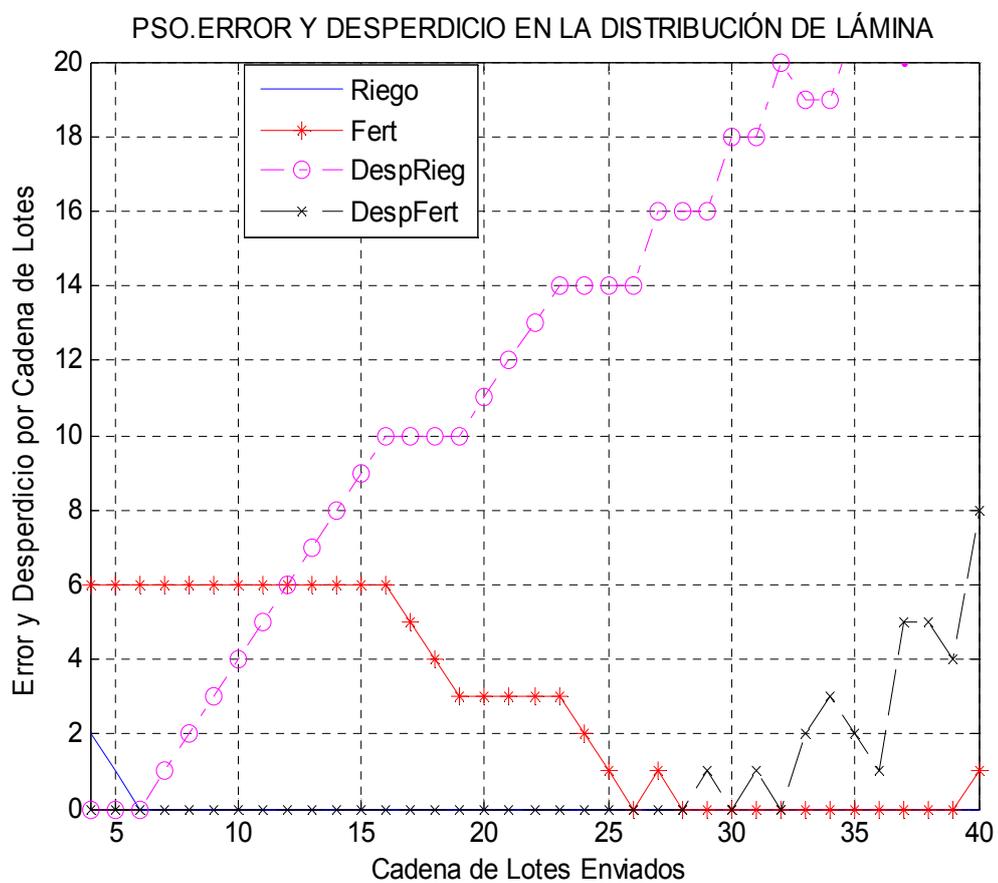
**Figura B-42:** Error de minimización Épocas: 200; Población: 100



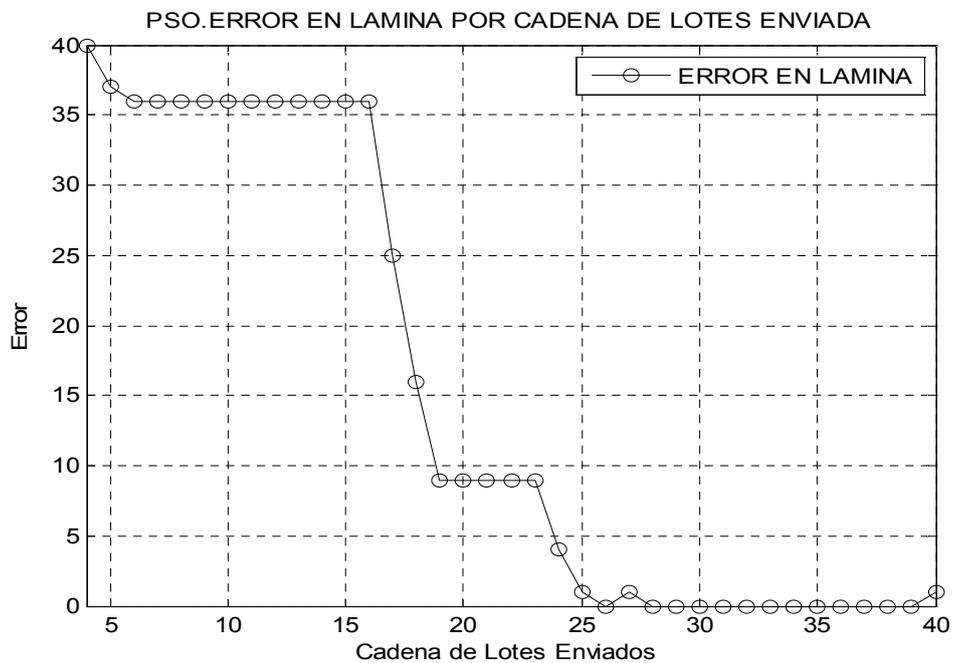
Los parámetros utilizados para simular las gráficas de la Figura B-43, Figura B-44 y la Figura B-45 son los siguientes:

- Épocas = 500.
- Población = 1000.

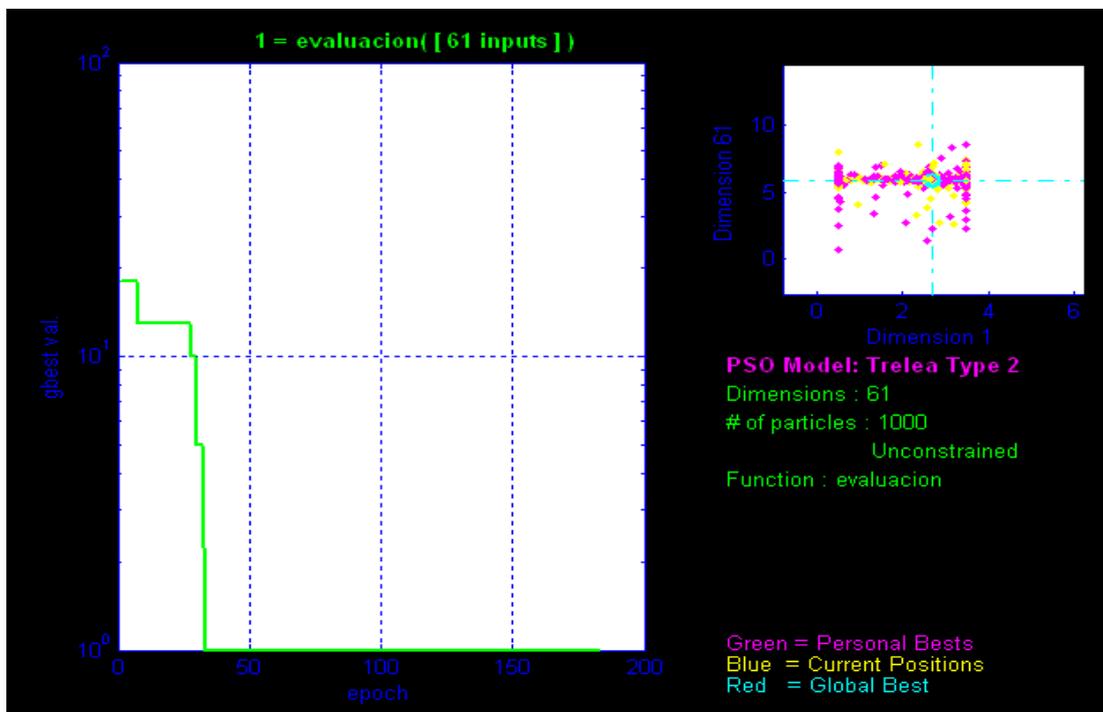
**Figura B-43:** Error y desperdicio. Épocas: 500; Población: 1000



**Figura B-44:** Error en riego y fertirriego. Épocas: 500 ; Población: 1000



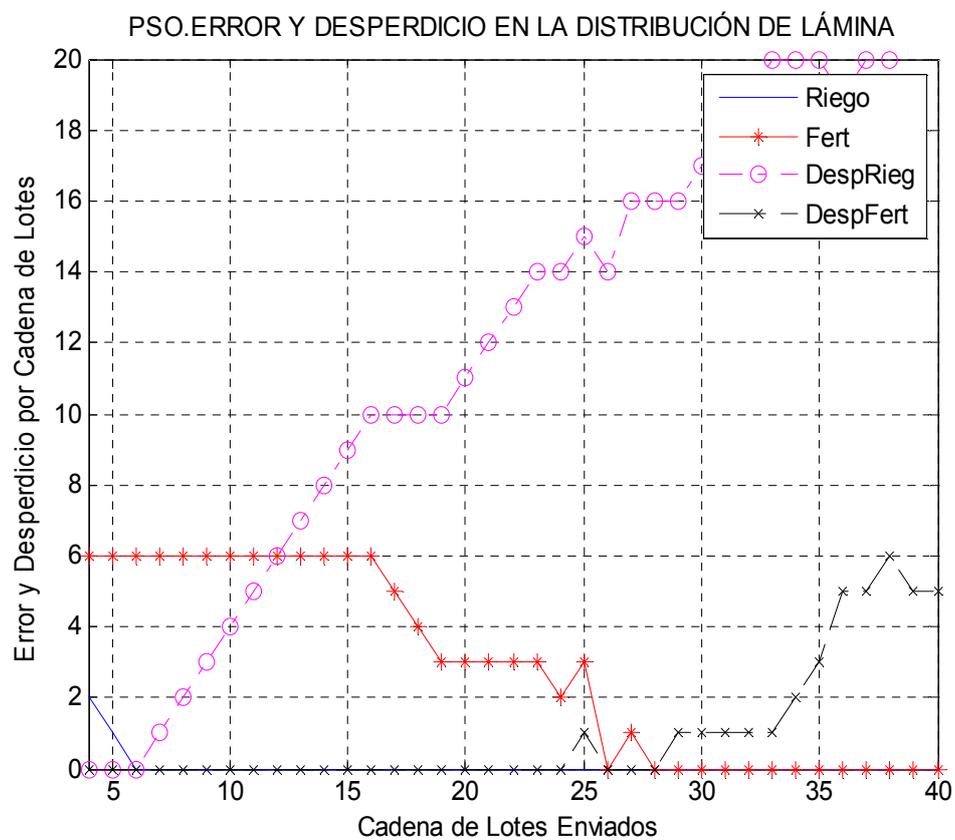
**Figura B-45:** Error de minimización. Épocas: 500; Población: 1000



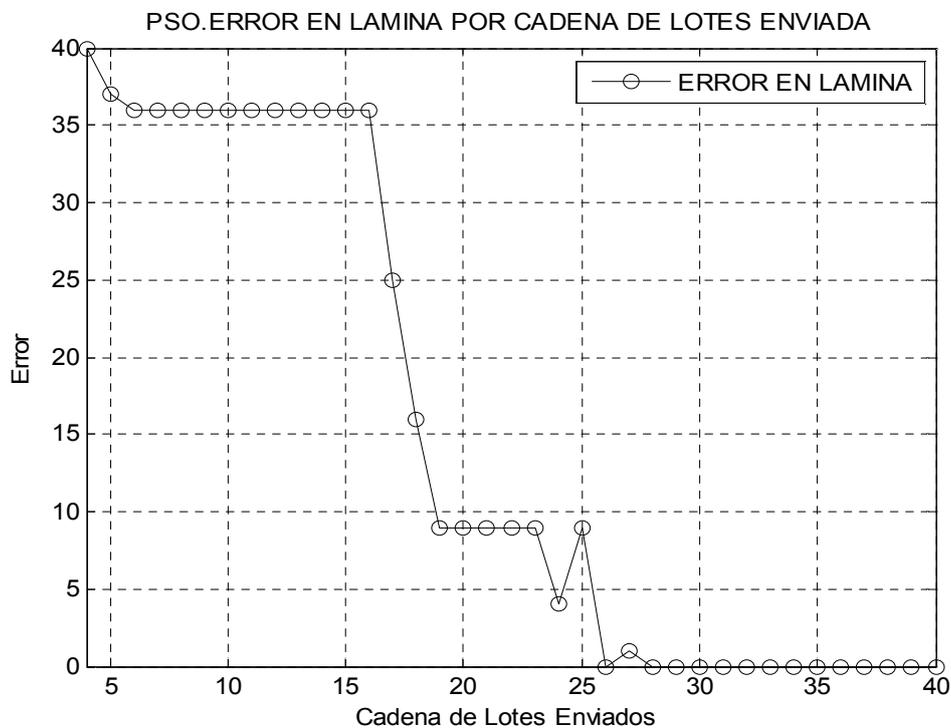
Los parámetros utilizados para simular las gráficas de la Figura B-46, Figura B-47 y la Figura B-48 son los siguientes:

- Número de épocas 200.
- Población 2000.

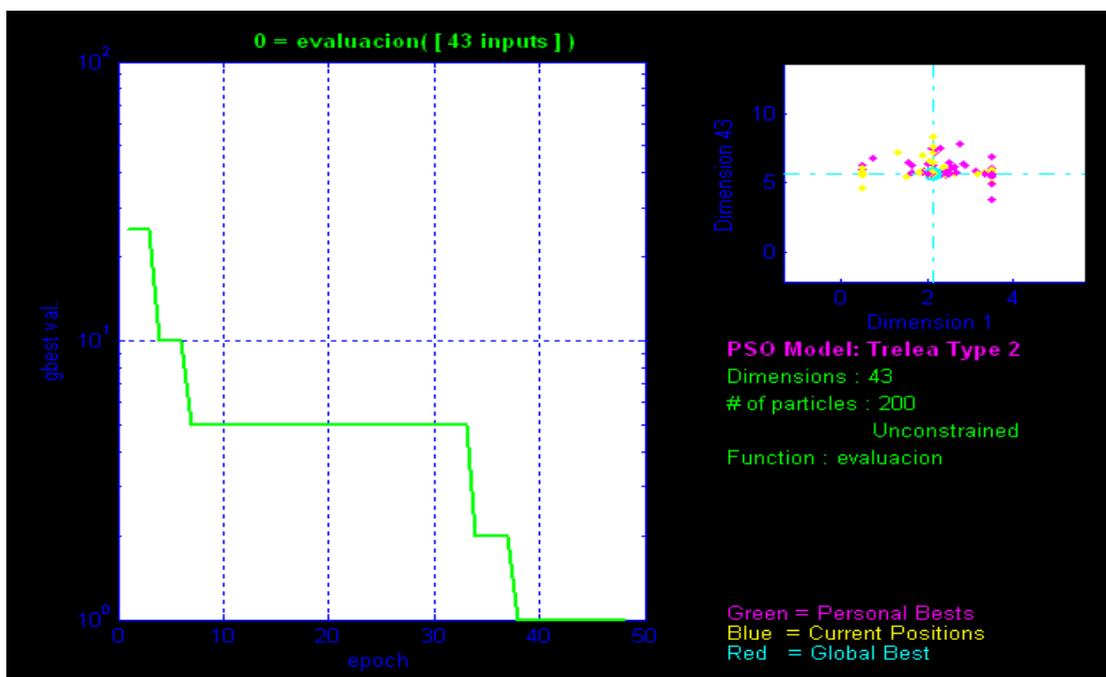
**Figura B-46:** Error y desperdicio. Épocas 200. Población 2000



**Figura B-47:** Error en riego y fertirriego. Épocas: 200 ; Población: 2000



**Figura B-48:** Error de minimización. Épocas 200. Población 2000



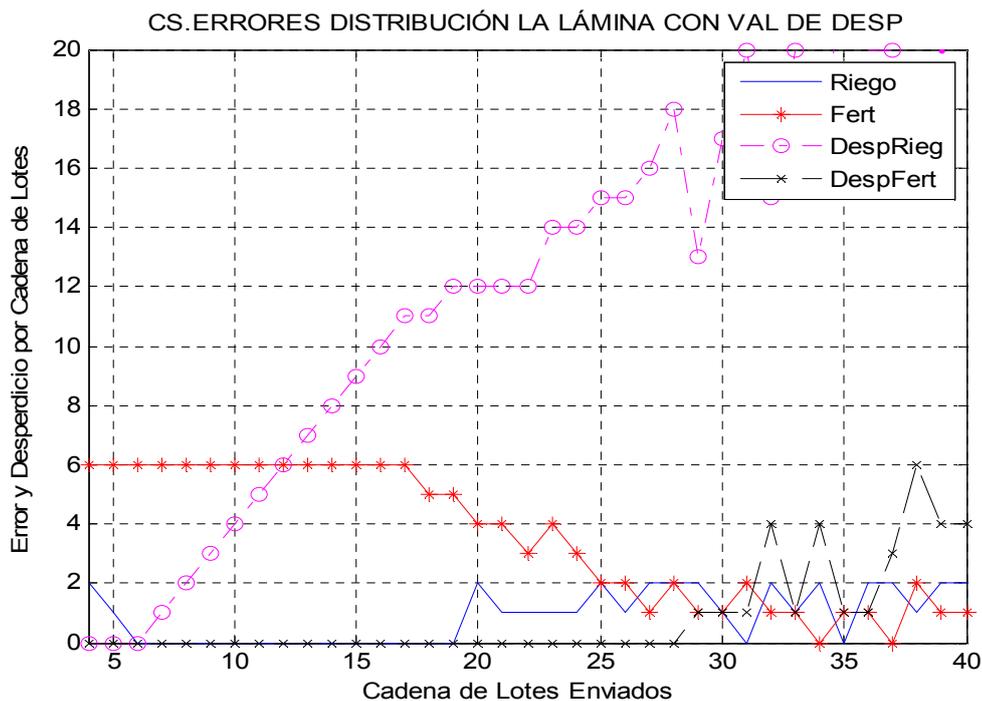
## C. Anexo: Simulaciones con el algoritmo Cuckoo Search

Las siguientes gráficas corresponden a las cinco simulaciones seleccionadas del conjunto de 25 realizadas con el algoritmo *ruta* (Algoritmo 2-1), el algoritmo *eval* (Algoritmo 2-2) y el algoritmo **Cuckoo Search**, para diferentes valores de épocas y población

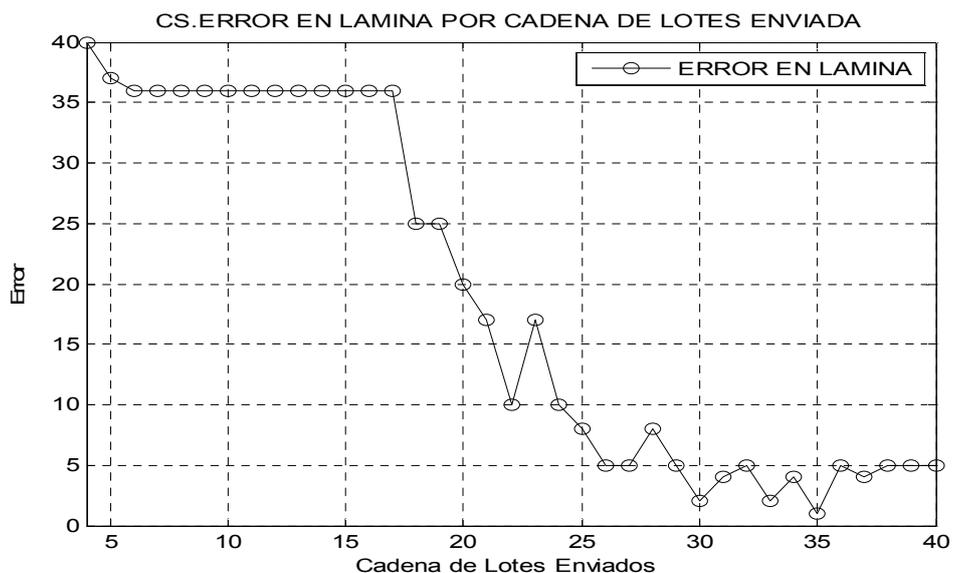
Los parámetros utilizados son los siguientes para la Figura C-49, Figura C-50

- Número de nidos (  $n$  ) = 25
- Probabilidad de descubrimiento ( $p_a$ ) = 0.25
- Número de iteraciones (N) = 100

**Figura C-49:** Error y desperdicio. N = 100



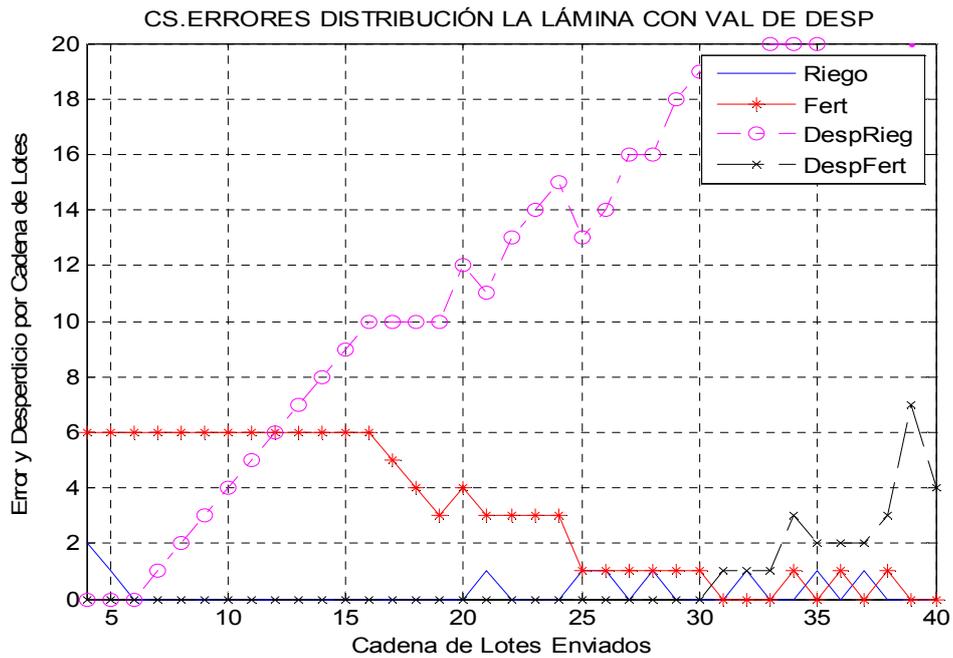
**Figura C-50:** Error en riego y fertiriego N = 100



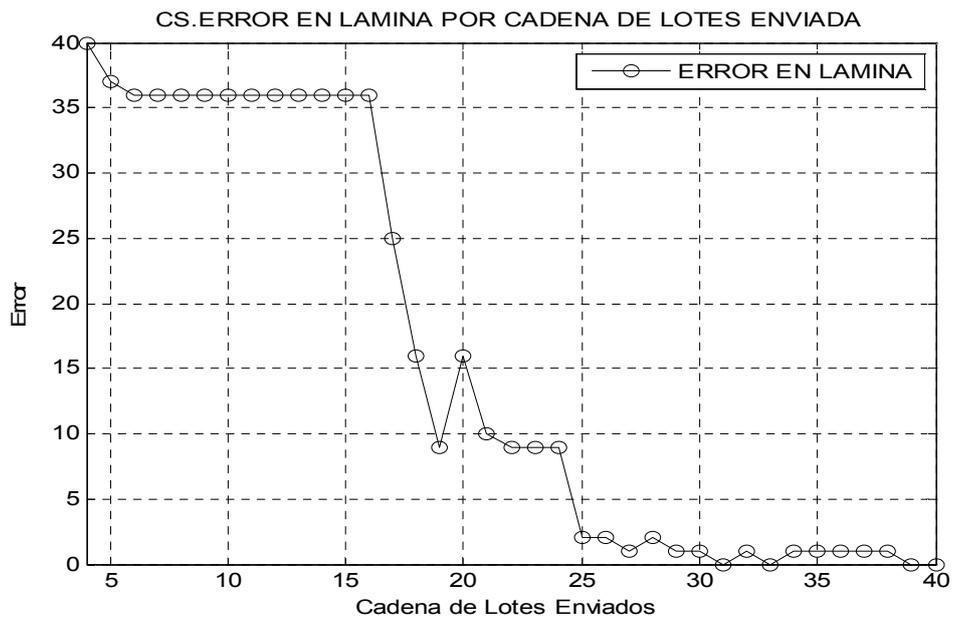
Los parámetros utilizados son los siguientes para la Figura C-51, Figura C-52

- Número de nidos (  $n$  ) = 25
- Probabilidad de descubrimiento (  $p_a$  ) = 0.25
- Número de iteraciones (  $N$  ) = 300

**Figura C-51:** Error y desperdicio. N= 300



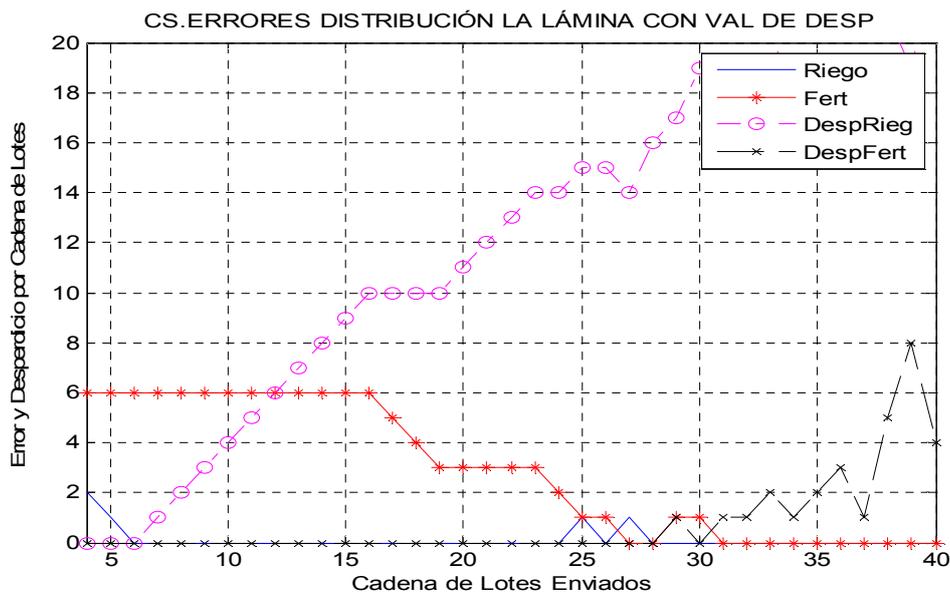
**Figura C-52:** Error en riego y Fertiriego. N= 300



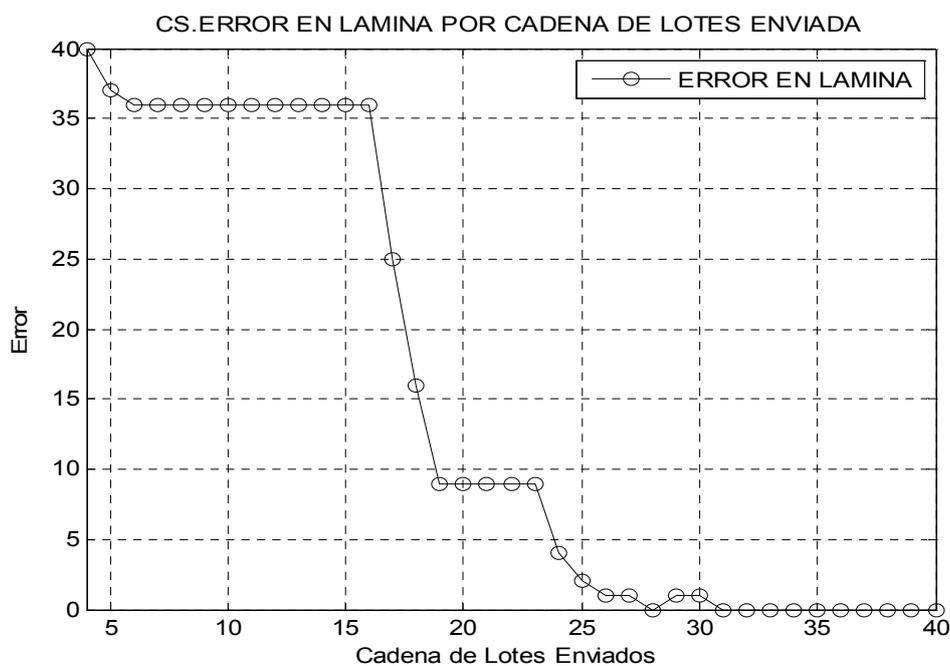
Los parámetros utilizados son los siguientes para la Figura C-53, Figura C-54

- Número de nidos (  $n$  ) = 25
- Probabilidad de descubrimiento ( $p_a$ ) = 0.25
- Número de iteraciones (N) = 500

**Figura C-53:** Error y desperdicio. N = 500



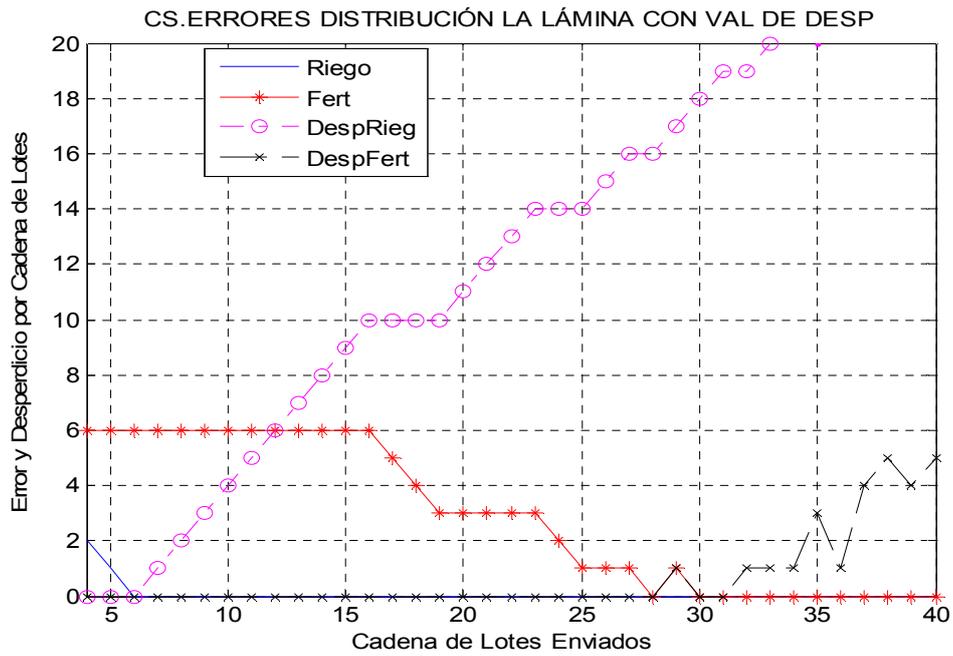
**Figura C-54:** Error en riego y fertirriego. N = 500



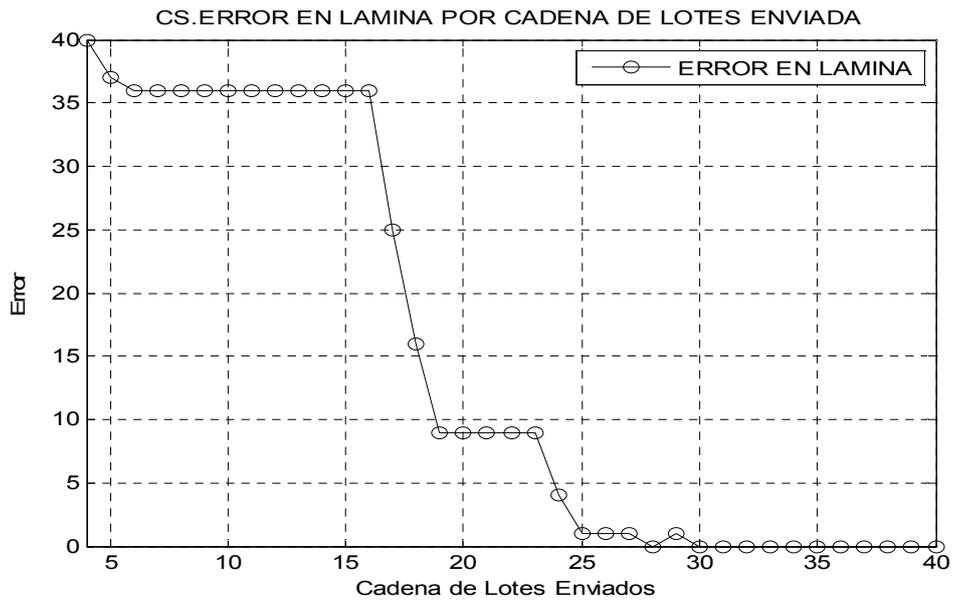
Los parámetros utilizados son los siguientes para la Figura C-55, Figura C-56

- Número de nidos (  $n$  ) = 25
- Probabilidad de descubrimiento ( $p_a$ )= 0.25
- Número de iteraciones (N) = 1000

**Figura C-55:** Error y desperdicio. N = 1000



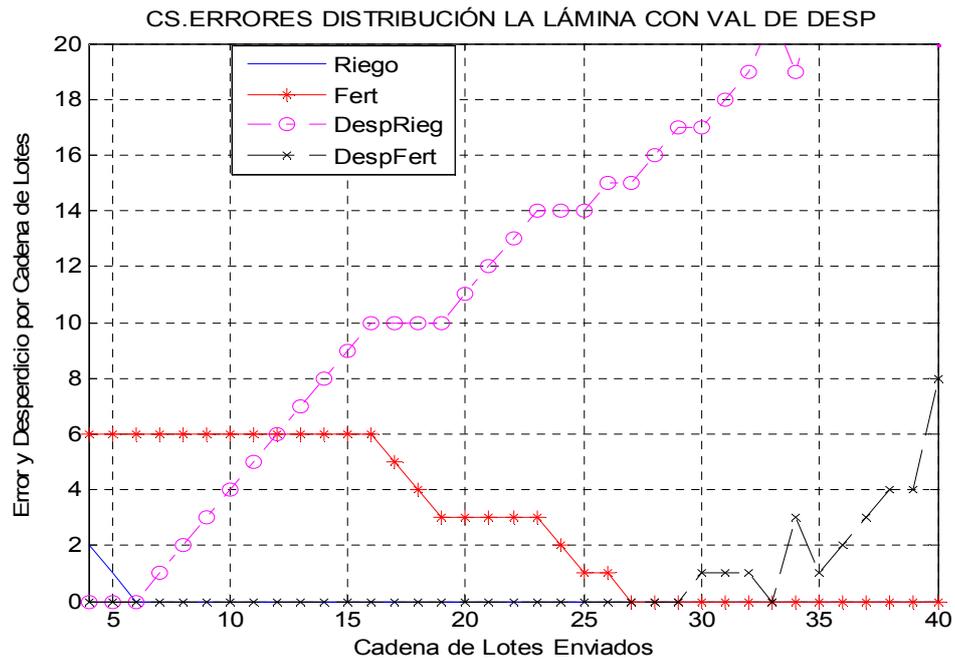
**Figura C-56:** Error en riego y fertiriego. N = 1000



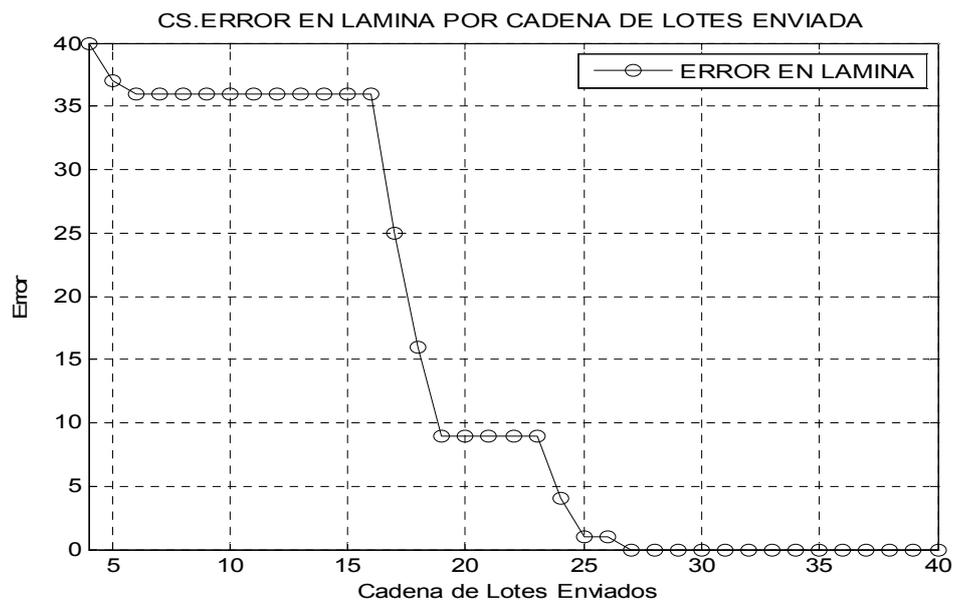
Los parámetros utilizados son los siguientes para la Figura C-57, Figura C-58

- Número de nidos (  $n$  ) = 25
- Probabilidad de descubrimiento (  $p_a$  ) = 0.25
- Número de iteraciones (  $N$  ) = 1500

**Figura C-57:** Error y desperdicio.  $N = 1500$



**Figura C-58:** Error en riego y fertiriego.  $N = 1500$



## D. Anexo: Especificaciones de los componentes de la red hidráulica

A continuación se detallan las especificaciones de los componentes de la red hidráulica.

- El diámetro de la tubería 4 pulgadas.
- El espesor de la tubería 4.96 mm.
- El diámetro externo 114 mm. L
- Razón diámetro / espesor: 21.
- Presión de operación de la red: de 4.5 bar a 5 bar.
- Caudal de la bomba principal; 18 m<sup>3</sup>/h.
- Diámetro y Caudal de las válvulas hidráulicas de riego: 2 in
- Caudal de las válvulas hidráulicas de riego: 40 m<sup>3</sup>/h.
- Hidrómetro: con indicador de flujo, registro de volumen en m<sup>3</sup>, precisión +/- 0.1l, rango del dial de 0 m<sup>3</sup> a 4 m<sup>3</sup>.
- Capacidad del inyector venturi: 200 l/h a 220 l/h con presión diferencial de 1.7 bares a 2.0 bares y presión +/- 0.01 l.
- electrobomba (*booster*) de 1.4 kW.
- Medidor de volumen de fertilizante: 1 Litro por pulso, precisión de +/-0.01 l
- Longitud de las cintas de goteros 30 m.
- Espaciamiento entre goteros en la cinta 30 cm.
- Caudal por gotero 2.15 l/h a 1bar.
- Cintas de goteros por cama 4.

## E. Anexo: Aproximación al cálculo de una lámina de riego

El cálculo de una lámina de riego o fertirriego por plántula, dependen de la metodología y técnicas usadas por la ingeniería agronómica [29,2], y del conocimiento y experiencia del agrónomo. Para este trabajo se asignó un valor de 0.3 litros \* día por plántula para riego y lo mismo para fertirriego [28, 30]. Se aclara que el volumen de fertirriego expresado aquí, está en la proporción agua–fertilizante necesaria para la planta. Esta proporción la define el agrónomo, se programa en el controlador de riego, y se mezclan en el momento de la inyección de fertilizante en la tubería.

La lámina por planta se debe entregar bajo la siguiente condición. En un ciclo podrá depositarse de 0.075 litros a 0.15 litros. Es decir, mínimo 0.075 litros, máximo 0.15 litros. El ciclo se debe repetir en un intervalo de tiempo de 0.5 a 1 hora, es decir mínimo cada 0.5 hora, máximo cada 1 hora. Hasta entregar la totalidad de la lámina, 0.3 litros. Con estos datos y los dados en el Anexo D, se procede a programar la lámina para el cultivo que se ha seleccionado como modelo. En la tabla E-10 se puede observar los volúmenes programados para distribuir en el cultivo.

**Tabla E-10:** Programa de riego fertirriego (Lámina deseada) a distribuir en el cultivo.

Válvula	V11	V12	V21	V22
Volumen (m <sup>3</sup> ) Riego	7.68	0	0	7.68
Volumen (m <sup>3</sup> ) Fertirriego	0	7.68	7.68	0

A continuación se calcula la equivalencia de un lote de riego o fertirriego con la siguiente ecuación:

$$L_{r-fr} = D_{VV} * D_{VV} \quad (E.1)$$

donde :

$L_{r-fr}$  = Lote de riego o fertirriego

$D_{VV}$  = Distancia entre dos válvulas consecutivas

$D_{VV}$  = Volumen por unidad de longitud.

$L_{r-fr}$  = 255 litros

A continuación se obtiene la lámina programada especificada en cantidad de lotes.

Como hay una cantidad programada para cada válvula la expresión es:

$$L_{LP} = \frac{V_{PV}}{L_{r-fr}} \quad (E.2)$$

Donde:

$L_{LP}$  = Lotes por lámina programada

$V_{PV}$  = Volumen programado por válvula

$L_{LP}$  = 30 lotes

Esta nueva representación la lámina programada queda como se muestra en la Tabla E-11

**Tabla E-11:** Lámina deseada expresada en unidades de lotes

Válvula	V11	V12	V21	V22
Lotes de Riego	30.1	0	0	30.1
Lotes de Fertirriego	0	30.1	30.1	0

## F. Anexo: Funciones y rutinas de llamado a los algoritmos evolutivos

Las siguientes figuras describen los pseudocódigos de los programas principales para llamar a los algoritmos evolutivos y los algoritmos *eval* y *ruta* y poder visualizar las respuestas.

**Figura F-59:** Pseudocódigo del programa principal para llamar al algoritmo *Cuckoo Search*.

```

Función principal_ALGORITMO CS
Definir el número de lotes mínimo: lotmin
Definir el número de lotes maximo: lotmax
Definir longitud de la palabra binaria vexp
para lotes ← lotmin hasta lotmax
Llamar la función que corre el CS x=cuckoo_search_new (lotes,vexp);
Definir Matriz que representa el modelo de la red hidráulica
Llamar el vector solución x , generado por el CS
Definir vector de lámina de riego deseado aguad
Definir vector de lámina de fertirriego deseado fertd
Definir Vector de las válvulas de riego y desperdicio : numval
Definir cantidad de lotes de riego y fertirriego para realizar la simulación
Definir válvulas desperdicio agua :valea
Definir válvulas de desperdicio válvulas fertirriego: valef.

procedimiento principal.
Elegir del vector x los n primer elementos para conformar el vector riego-fertirriego, VRF.
Convertir VRF a binario.
Conformar el vector de Válvulas VAL con los x-n.
Crear matriz DIST con los vectores VRF y VAL.
Inicializar lista de vectores en ceros.
Inicializar vectores agua y fert.
para i←1 hasta número de válvulas
ruta (MATRIZ, valvula).
Actualizar agua y fert con número de lotes de riego o fertirriego.

```

```

Actualizar lista con agua y fert.
fin para.
errora ← sum (abs (aguad - agua) ).
errorf ← sum (abs (ferd - fert) ) .
error ← errora2+ errorf2.
despa ← sum(lotes agua, valea)
despf ← sum(lotes fert, valef)
fin procedimiento
fprintf(fid,'Error en agua = %i\n',errora);
fprintf(fid,'Error en fert:%i\n',errorf);
fprintf(fid,'Despilfarro en agua:%i\n',despa);
fprintf(fid,'Despilfarro en fert:%i\n',despf);
fprintf(fid,'distribucion lámina riego.%i\n',DIST);
fprintf(fid,'error de minimización por lote.%i\n',fval);
fprintf(fid,'\n');
fin para

```

**Figura F-60:** Pseudocódigo del programa principal para llamar al algoritmo Genético.

```

Función principal_ALGORITMO GENÉTICO

Definir el número de lotes mínimo: lotmin
Definir el número de lotes maximo: lotmax

para lotes ← lotmin hasta lotmax

MATRIZ: Matriz de lotes. Cada  $m_{ij}$  representa el número de lotes de del nodo i al nodo j.
x: vector de programación riego, fertirriego y válvulas de riego, generado por el algoritmo
optimización
aguad: lámina de riego deseado.
fertd: lámina de fertirriego deseado.
numval: Vector que define las válvulas de riego
lotes: cantidad de lotes de riego y fertirriego para un recorrido completo en la red.

```

```

vexp = longitud binaria
valea: válvulas desperdicio agua
valef: válvulas desperdicio fert

f=@(x)eval(x,MATRIZ,aguad,ferd,numval,valea,valef,lotes,vexp,);% formato para llamar el AG
la
función objetivo
nd = lotes + lotes/2 +2; % dimensión en x del problema
lb=zeros(1,nd);% Límite inferior en x
ub=ones(1,nd)*length(numval));% Límite superior en x

options=gaoptimset(options,'PlotFcns',{@gaplotbestf}, 'Display','iter');
options = gaoptimset(options,'InitialPopulation');
[x,fval]=ga(f,nd,[],[],[],[],Lb,Ub,[],options);

procedimiento principal.
Elegir del vector x los n primer elementos para conformar el vector riego-fertiriego, VRF.
Convertir VRF a binario.
Conformar el vector de Válvulas VAL con los x-n.
Crear matriz DIST con los vectores VRF y VAL.
Inicializar lista de vectores en ceros.
Inicializar vectores agua y fert.
para i←1 hasta número de válvulas
ruta (MATRIZ, valvula).
Actualizar agua y fert con número de lotes de riego o fertiriego.
Actualizar lista con agua y fert.
fin para.
errora ← sum (abs (aguad - agua) ).
errorf ← sum (abs (ferd - fert) ) .
error← errora2+ errorf2.
despa←sum(lotes agua, valea)
despf←sum(lotes fert, valef)

fin procedimiento
fprintf(fid,'Error en agua = %i\n',erra);

```

```

fprintf(fid,'Error en fert:%i\n',errf);
fprintf(fid,'Despilfarro en agua:%i\n',despa);
fprintf(fid,'Despilfarro en fert:%i\n',despf);
fprintf(fid,'distribucion lámina riego:%i\n',DIST);
fprintf(fid,'error de minimización por lote:%i\n',fval);
fprintf(fid,'\n');

fin para

```

**Figura F-61:** Pseudocódigo del programa principal para llamar al algoritmo *PSO*.

```

Función principal_ALGORITMO PSO

Definir el número de lotes mínimo: lotmin
Definir el número de lotes maximo: lotmax

para lotes ← lotmin hasta lotmax

MATRIZ: Matriz de lotes. Cada  $m_{ij}$  representa el número de lotes de del nodo  $i$  al nodo  $j$ .
x: vector de programación riego, fertirriego y válvulas de riego, generado por el algoritmo
optimización
aguad: lámina de riego deseado.
fertd: lámina de fertirriego deseado.
numval: Vector que define las válvulas de riego
lotes: cantidad de lotes de riego y fertirriego para un recorrido completo en la red.
vexp = longitud binaria
valea: válvulas desperdicio agua
valef: válvulas desperdicio fert

funcname=@(x)eval(x,MATRIZ,aguad,fertd,numval,valea,valef,lotes,vexp,);% formato para
llamar el PSO la función objetivo
D = lotes + lotes/2 +2; % dimensión en x del problema
lb=zeros(1,nd);% Límite inferior en x

```

```

ub=ones(1,nd)*length(numval));% Límite superior en x
limt=[lb ub]; Matriz de limites
[x,fval]=pso_trelea_vectorized(funcname,D,NaN,limt,0,[200,2000,300,2,2,0.9,0.4,1500,1e+25,..
150,NaN,2,0]);

procedimiento principal.
Elegir del vector x los n primer elementos para conformar el vector riego-fertiriego, VRF.
Convertir VRF a binario.
Conformar el vector de Válvulas VAL con los x-n.
Crear matriz DIST con los vectores VRF y VAL.
Inicializar lista de vectores en ceros.
Inicializar vectores agua y fert.
para i← 1 hasta número de válvulas
ruta (MATRIZ, valvula).
Actualizar agua y fert con número de lotes de riego o fertiriego.
Actualizar lista con agua y fert.
fin para.
errora ← sum (abs (aguad - agua) ).
errorf ← sum (abs (ferd - fert) ) .
error← errora2+ errorf2.
despa←sum(lotes agua, valea)
despf←sum(lotes fert, valef)

fin procedimiento
fprintf(fid,'Error en agua = %i\n',erra);
fprintf(fid,'Error en fert:%i\n',errf);
fprintf(fid,'Despilfarro en agua:%i\n',despa);
fprintf(fid,'Despilfarro en fert.%i\n',despf);
fprintf(fid,'distribucion lámina riego.%i\n',DIST);
fprintf(fid,'error de minimización por lote.%i\n',fval);
fprintf(fid,'\n');
fin para

```

## Bibliografía

- [1] Amezcuita, C. E. (1994). Fertirrigación: algunos comentarios sobre su utilización. En: F. Silva Mojica (Ed.). *Fertilidad de los suelos: diagnóstico y control* (pp 283–292). Bogotá: Sociedad Colombiana de la Ciencia del Suelo.
- [2] Santos, F. L. (1996). Evaluation and adoption of irrigation technologies. I. Management-design Curves for Furrow and Level Basin Systems. *Agricultural Systems*, 52 (2/3), 317- 329.
- [3] Miranda, D. (2004). Componentes de los sistemas productivos colombianos. En M. Guzmán Palomino & J. López Galvéz (Eds). *Ferti-Riego: tecnologías y programación en agroplasticultura*. (pp 115-140) Madrid, España: Programa CYTED
- [4] Salas Sanjuán, M. del C., Urrestarazu Gavilán, M. (2001). Objetivos, controles y métodos de fertirrigación en cultivos sin suelo. *Revista Horticultura*, 157, 38-48.
- [5] Coello, C. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems. Springer Science & Business Media.
- [6] Bernon, J. L., Boudousq, V., Rohmer, J. F., Fourcade, M., Zanca, M., Rossi, M. & Mariano-Goulart, D. (2001). A comparative study of Powell's and Downhill Simplex
- [7] Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*, 39(4), 315-346.
- [8] Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*, 39(4), 315-346.
- [9] Duarte, C. & Quiroga J. (2010). System identification of a DC motor using PSO algorithm. *Revista Facultad de Ingeniería Universidad de Antioquia*, 55, 116-124.

- 
- [10] Bäck, T., Fogel D. & Michalewicz Z. (1997). Handbook of evolutionary computation. IOP New York: Oxford University Press.
- [11] Holland, J. (1975) Adaptation in natural and artificial systems. Cambridge, Massachusetts, MIT Press.
- [12] Kennedy J. & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks IV*. pp 1942-1956.
- [13] X. S. Yang and S. Deb, "Cuckoo Search via Levy Flights", Proceedings of World Congress on Nature & Biologically Inspired Computing, (2009), pp. 210-225].
- [14] Deuerlein, J. W. (2008). Decomposition model of a general water supply network graph. *Journal of Hydraulic Engineering*, 134(6), 822-832.
- [15] Abdel-Baset, M., Selim, I. M., & Hezam, I. M. (2015). Cuckoo search algorithm for stellar population analysis of galaxies. Doi: 10.5815.
- [16] Preparata, F.P. y Shamos, M.( 1985). *Computational geometry: an introduction*. New York: Springer-Verlag.
- [17] Duran-Ros M., Puig-Bargués, J., Arbat G., Barragán, J. & Ramírez de Cartagena, F. (2008). Definition of a SCADA system for a microirrigation network with effluents. *Computers and Electronics in Agriculture*, 64, 338–342.
- [18] Selman, B., & Gomes, C. P. (2006). Hill-climbing Search. Encyclopedia of Cognitive Science.
- [19] Bernon, J. L., Boudousq, V., Rohmer, J. F., Fourcade, M., Zanca, M., Rossi, M. & Mariano-Goulart, D. (2001). A comparative study of Powell's and Downhill Simplex

---

algorithms for a fast multimodal surface matching in brain imaging. *Computerized Medical Imaging and Graphics*, 25(4), 287-297.

[20] Gómez, E. (1984). Algunos aspectos en el diseño hidráulico y económico de redes en sistemas de riego por goteo. *Revista Ingeniería e Investigación*, 8 (39).

[21] Ed-dahhak, A., Guerbaoui, M., ElAfou, Y., Outanoute, M., Lachhab, A., Belkoura, L., & Bouchikhi, B. (2013). Implementation of fuzzy controller to reduce water irrigation in Greenhouse using Labview. *International Journal of Engineering and Advanced Technology Studies*, 1(2), 12-22.

[22] Jimenez-Bello, M. A., Martínez, F., Bou, V., & Bartolín, H. (2011). Analysis, assessment, and improvement of fertilizer distribution in pressure irrigation systems. *Irrigation science*, 29(1), 45-53.

[23] Morari, F. & Giardini, L. (June 2002). Irrigation automation with heterogeneous vegetation: the case of the Padova botanical garden. *Agricultural Water Management*, 55 (3), 183–201.

[24] Zhao Yandong & Guan Jinfeng. (2009). *Study on precision water-saving irrigation automatic control system by plant physiology*, IEEE.

[25] Castro Popoca, M., Águila Marín, F. M., Quevedo Nolasco, A., Kleisinger, S., Tijerina Chávez, L., & Mejía Sáenz, E. (2008). Sistema de riego automatizado en tiempo real con balance hídrico, medición de humedad del suelo y lisímetro. *Agricultura técnica en México*, 34(4), 459-470.

[26] Pereira, L. S., Calejo, M.J. & Lamaddalena, N. (November 2003). Design and performance analysis of low pressure irrigation distribution systems. *Irrigation and Drainage Systems*, 17(4), 305-324.

- 
- [27] Alperovits, E., & Shamir, U. (1977). Design of optimal water distribution systems. *Water resources research*, 13(6), 885-900.
- [28] Renault, D. & Godaliyadda, G. G. A. (1999). *Generic typology for irrigation systems operation*. Colombo, Sri Lanka: International Water Management Institute.
- [29] Shock, C. C. (2013). *Drip irrigation: an introduction*. Corvallis, Oregon: Extension Service, Oregon State University.
- [30] Payne, R. B., Sorenson, M. D., & Klitz, K. (2005). *The cuckoos*, New York: Oxford University Press.
- [31] Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E*, 49(5), 4677
- [32] Ashby, W. R. (1962) Principles of the self-organizing system. En: H. von Foerster & G. W. Zopf Jr. (Eds), *Principles of self organization: transactions of the University of Illinois Symposium* (pp 255–278). London, UK, Pergamon Press.
- [33] Keller, E F (2009) Organisms, machines, and thunderstorms: a history of self-organization, part two. Complexity, emergence, and stable attractors. *Historical Studies in the Natural Science*, 39(1), 1–31.
- [34] Wolpert, D.H. & Macready WG (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82.
- [35] Bernon, J. L., Boudousq, V., Rohmer, J. F., Fourcade, M., Zanca, M., Rossi, M. & Mariano-Goulart, D. (2001). A comparative study of Powell's and Downhill Simplex algorithms for a fast multimodal surface matching in brain imaging. *Computerized Medical Imaging and Graphics*, 25(4), 287-297.

- 
- [36] Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J Bio-inspir Comput* 2(2):78–84],[ Yang XS, Deb S (2010) Engineering optimization by cuckoo search. *Int J Math Modell Num Opt* 1(4):330–343.
- [37] Yang, X. S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver Press.
- [38] Kirkpatrick S., Gelatt C. & Vecchi M. (1983). Optimization by simulated annealing. *Science*, 220(4598).
- [39] Glover F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13, 533-549.
- [40] Ramírez, D. A. (2010). *Detección de daños en estructuras utilizando la técnica de optimización PSO*. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingeniería FísicoMecánica. Escuela de Ingeniería Mecánica. Trabajo de grado Ingeniero Mecánico
- [41] Duarte, C. & Quiroga J. (2010). System identification of a DC motor using PSO algorithm. *Revista Facultad de Ingeniería Universidad de Antioquia*, 55, 116-124.
- [42] Payne, R. B., Sorenson, M. D., & Klitz, K. (2005). *The cuckoos*, New York: Oxford University Press.
- [43] Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E*, 49(5), 4677.
- [44] Euler, L. *Solutio Problematis and Geometriam Situs Pertinentis*. *Comment. Acad. Sci. I. Petrop* 8, 128-40, 1741 (Reimpreso en *Opera Omnia Series Prima*, Vol. 7. pp. 1-10, 1766).

- 
- [45] Johnsonbaugh, R. (2005). Matemáticas discretas. México: Pearson Educación.
- [46] Optimization toolbox for use with MATLAB, User Guide, The MathWorks Inc.2006
- [47] Birge, B. (2006). *Particle swarm optimization toolbox. Matlab: www.mathworks.com/matlabcentral,*
- [48] Martínez E. Rene. (2004). Evaluación de un sistema de riego localizado (microirrigación). En M. Guzmán Palomino & J. López Galvéz (Eds). *Ferti-Riego: tecnologías y programación en agroplasticultura*. (pp 153-160) Madrid, España: Programa CYTED.
- [49] Peña A., Fernández V. (2004). Automatización de sistemas de fertirriego: una tecnología que requiere adaptación del campo mexicano. En M. Guzmán Palomino & J. López Galvéz (Eds). *Ferti-Riego: tecnologías y programación en agroplasticultura*. (pp 153-160) Madrid, España: Programa CYTED.
- [50] Peña A., Fernández V. (2004). Automatización de sistemas de fertirriego: una tecnología que requiere adaptación del campo mexicano. En M. Guzmán Palomino & J. López Galvéz (Eds). *Ferti-Riego: tecnologías y programación en agroplasticultura*. (pp 153-160) Madrid, España: Programa CYTED.
- [51] Merriam, J. L., Burt, C. M., Clemmens, A. J., Solomon, K. H., Howell, T. A., & Strelkoff, T. S. (1999). Irrigation performance measures: efficiency and uniformity. *Journal of Irrigation and Drainage Engineering*, 125 (2), 97-99.
- [52] Rodríguez D. Eduardo. (2004). Componentes de los sistemas de ferti-riego. En M. Guzmán Palomino & J. López Galvéz (Eds). *Ferti-Riego: tecnologías y programación en agroplasticultura*. (pp 153-160) Madrid, España: Programa CYTED.