



UNIVERSIDAD AUTÓNOMA CHAPINGO

POSGRADO EN INGENIERÍA AGRÍCOLA Y USO INTEGRAL DEL AGUA

**OPTIMIZACIÓN DEL DISEÑO DE REDES DE RIEGO CON ALGORITMOS
EVOLUTIVOS Y BIO-INSPIRADOS**

TESIS

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

PRESENTA:

EDUARDO JIMÉNEZ HERNÁNDEZ

BAJO LA SUPERVISIÓN DE: DR. IRINEO L. LÓPEZ CRUZ

CHAPINGO, ESTADO DE MÉXICO, DICIEMBRE DE 2017



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

OPTIMIZACIÓN DEL DISEÑO DE REDES DE RIEGO CON ALGORITMOS
EVOLUTIVOS Y BIO-INSPIRADOS

I

Tesis realizada por Eduardo Jiménez Hernández, bajo la supervisión del Comité Asesor indicado, aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN INGENIERÍA AGRÍCOLA Y USO INTEGRAL DEL AGUA

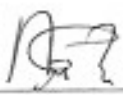
DIRECTOR:


DR. IRINEO L. LÓPEZ CRUZ

ASESOR:


DRA. RAQUEL SALAZAR MORENO

ASESOR:


DR. AGUSTÍN RUÍZ GARCÍA

Chapingo, Estado de México, diciembre de 2017

Agradecimientos

En primer lugar, agradezco mucho al CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA (CONACyT) por el apoyo otorgado para la realización de este trabajo de investigación.

A la UNIVERSIDAD AUTÓNOMA CHAPINGO, particularmente al Posgrado en Ingeniería Agrícola y Uso Integral del Agua, por proporcionarme la oportunidad de realizar mis estudios de posgrado.

Al DR. IRINEO L. LÓPEZ CRUZ, su ejemplo y enseñanzas acerca de la investigación científica han tenido gran impacto en mi formación académica.

A la DRA. RAQUEL SALAZAR MORENO, por su valiosa ayuda proporcionada para afrontar el reto de trabajar temas nuevos y complejos.

Al DR. AGUSTÍN RUÍZ GARCÍA, cuyo consejos, recomendaciones y orientación fueron fundamentales en el desarrollo del trabajo.

Al DR. MAURICIO CARRILLO GARCÍA quién me ha proporcionado valiosos consejos para la vida, particularmente en la decisión de hacer un posgrado en esta prestigiosa institución.

DATOS BIOGRÁFICOS

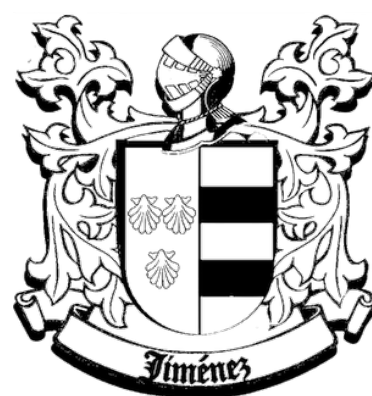
Eduardo Jiménez Hernández (20 de septiembre de 1988) es originario de Candalaria, Campeche.

Su educación básica la recibió en su lugar de origen, su educación media superior la recibe del año 2003 al 2006 en la Preparatoria Agrícola y posteriormente su licenciatura de 2006 a 2010 en el Departamento de Irrigación, ambos en la Universidad Autónoma Chapingo.

Durante su desempeño profesional trabajó como consultor para oficinas de proyectos de la Universidad Autónoma Chapingo, Instituto Interamericano de Cooperación para la Agricultura (IICA) y el Fondo de Fomento y Desarrollo de la Investigación Científica y Tecnológica de la Universidad Autónoma del Estado de México (FONDICT-UAEM).

Realizó sus estudios de maestría en ingeniería durante el periodo 2015 – 2017 en el Posgrado de Ingeniería Agrícola y Uso Integral del Agua (IAUIA) de la Universidad Autónoma Chapingo.

A mi familia.



Índice general

Agradecimientos	II
Resumen	XII
Abstract	XIII
1. Introducción	1
1.1. Breve descripción del estado actual del uso del agua en México	1
1.2. Motivación	4
2. Revisión de Literatura	7
2.1. Optimización	7
2.1.1. Optimización continua	8
2.1.2. Optimización combinatoria	9
2.1.3. Optimización multiobjetivo	10
2.2. Métodos de optimización	11
2.2.1. Complejidad computacional	13
2.2.2. Heurísticas, metaheurísticas e hyper-heurísticas	15
2.2.3. Computación evolutiva	16
Algoritmos genéticos	19
2.2.4. Algoritmos bio-inspirados	21
Optimización con colonia de hormigas	22
2.3. Trazo y dimensionamiento de redes de distribución de agua	25
2.4. Optimización de sistemas de distribución de agua	32
2.5. Optimización multiobjetivo de sistemas de distribución de agua	34

3. Materiales y métodos	36
3.1. Problema de trazo	38
3.1.1. Variables de decisión	38
3.1.2. Función objetivo	38
3.1.3. Restricciones	39
3.2. Problema de dimensionamiento	40
3.2.1. Variables de decisión	40
3.2.2. Función objetivo	41
3.2.3. Restricciones	41
3.3. Problema multiobjetivo	43
3.4. Procedimiento de solución	44
3.4.1. Solución con algoritmos genéticos	45
3.4.2. Solución con optimización con colonia de hormigas	47
3.4.3. Desarrollo de software para optimización de redes de riego	51
3.5. Ejemplos de aplicación	53
3.5.1. Red de dos circuitos	53
3.5.2. Red de 3×3	54
3.5.3. Red de 8×8	55
3.5.4. Red de riego de 12 nodos	55
3.6. Escenarios de prueba	56
4. Resultados y discusión	61
4.1. Problema de trazo	61
4.1.1. Red de 3×3	61
4.1.2. Red de 8×8	65
4.1.3. Red de riego de 12 nodos	69
4.2. Problema de dimensionamiento	73
4.2.1. Red de dos circuitos	73
4.2.2. Red de 3×3	79
5. Conclusiones	85
A. Código fuente	86

Bibliografía

Índice de figuras

1.1. Distribución del uso del agua concesionada en México en el año 2015 (CONAGUA, 2016)	2
2.1. Diferentes enfoques para optimización global (Modificado de Coello Coello et al. (2007))	12
2.2. La relación entre las clases de problemas P, NP, NP-Difícil y NP-Completo	15
2.3. Principales métodos de la computación evolutiva (Brabazon et al., 2015)	17
2.4. Esquema general de un algoritmo evolutivo (Eiben y Smith, 2015)	18
2.5. Esquema general de una red abierta de tuberías para riego	28
2.6. La red de dos circuitos, una red de prueba usada frecuentemente en la literatura.	33
2.7. Configuración de la red de Hanoi, una red de prueba usada frecuentemente en la literatura.	35
3.1. Procedimiento de trabajo	37
3.2. Diagramas de flujo de GA (izq.) y ACO (der.) para el problema de dimensionamiento de redes.	48
3.3. Diagrama de clases del código realizado para ACO	49
3.4. Construcción de una solución para el dimensionamiento de una red ACO	50
3.5. Esquema del <i>framework</i> desarrollado en Python para la optimización del diseño de redes de distribución de agua usando GA y ACO	52
3.6. Red de 3×3 nodos, usada frecuentemente en la literatura para probar el desempeño de metaheurísticas en el diseño de redes.	55
3.7. Red de 8×8 nodos, usada frecuentemente en la literatura para probar el desempeño de metaheurísticas en el diseño de redes	57
3.8. Distribución de los hidrantes de la red de riego de 12 nodos.	59

4.1. Ejemplos de las mejores soluciones para el trazo de la red de 3×3 encontradas con Algoritmos Genéticos y Optimización con Colonia de Hormigas. Todas las soluciones óptimas para este problema tienen un patrón similar.	62
4.2. Convergencia de las soluciones para el trazo de la red de 3×3 obtenidas con Algoritmos Genéticos (<i>izq.</i>) y Optimización con Colonia de Hormigas (<i>der.</i>).	63
4.3. Solución óptima para el trazo de la red de 8×8 encontrada por la metaheurística Optimización con Colonia de Hormigas.	67
4.4. Convergencia de las soluciones obtenidas con Optimización con Colonia de Hormigas para el trazo de la red de 8×8	68
4.5. Mejores soluciones para el trazo de la red de riego de 12 nodos, a la izquierda se presenta la obtenida con Algoritmos Genéticos que tiene una longitud mínima de 1,987.72 unidades. Mientras que a la derecha la solución obtenida con Optimización con Colonia de Hormigas y tiene una longitud de 1,946.04 unidades.	71
4.6. Convergencia de las soluciones para el trazo de la red de riego de 12 nodos obtenido con Algoritmos Genéticos (<i>izq.</i>) y Optimización con Colonia de Hormigas (<i>der.</i>).	72
4.7. Convergencia de las soluciones obtenidas para el dimensionamiento de la red de dos circuitos con Algoritmos Genéticos.	76
4.8. Análisis hidráulico de la solución obtenida por Zhou et al. (2015) para la red de dos circuitos, se muestran los diámetros en mm y las presiones en m (<i>izq.</i>), así como las velocidades de flujo en $m s^{-1}$ (<i>der.</i>).	77
4.9. Análisis hidráulico de la mejor solución para la red de dos circuitos obtenida con Algoritmos Genéticos, se muestran los diámetros en mm y las presiones en m (<i>izq.</i>), así como las velocidades de flujo en $m s^{-1}$ (<i>der.</i>). . .	78
4.10. Convergencia de las soluciones obtenidas con Algoritmos Genéticos para el dimensionamiento de la red de 3×3	82
4.11. Análisis hidráulico de la solución obtenida por Saleh y Tanyimboh (2013) para la red de 3×3 , se muestran los diámetros en mm y las presiones en m (<i>izq.</i>), así como las velocidades de flujo en m/s (<i>der.</i>).	83

- 4.12. Análisis hidráulico de la mejor solución para la red de 3×3 obtenida con Algoritmos Genéticos, se muestran los diámetros en mm y las presiones en m (*izq.*), así como las velocidades de flujo en $m\ s^{-1}$ (*der.*). 84

Índice de cuadros

2.1. Algoritmos de ACO de acuerdo a su aparición cronológica	27
2.2. Velocidades máximas permisibles recomendadas para el diseño hidráulico.	31
3.1. Valores de los parámetros usados en el algoritmo genético.	47
3.2. Valores de los parámetros utilizados en la ejecución del algoritmo <i>Ant System</i> de ACO.	51
3.3. Datos de los nodos de la red de dos circuitos	53
3.4. Precios unitarios por diámetro de tubo para la red de dos circuitos	54
3.5. Demandas de agua en cada nodo de la red de 3×3	56
3.6. Datos de los nodos (hidrantes) de la red de riego de 12 nodos	58
3.7. Precios unitarios por diámetro de tubo	58
3.8. Escenarios de prueba de las metaheurísticas.	60
4.1. Soluciones para el trazo de la red 3×3 obtenidas con las metaheurísticas Algoritmos Genéticos (GA) y Optimización con Colonia de Hormigas (ACO).	64
4.2. Soluciones para el trazo de la red de 8×8 obtenidas con Optimización con Colonia de Hormigas	66
4.3. Soluciones para el trazo de la red de 12 nodos obtenidas con Algoritmos Genéticos (GA) y Optimización con Colonia de Hormigas (ACO).	70
4.4. Soluciones reportadas para el trazo de la red de riego de 12 nodos usando diferentes metaheurísticas: Programación Lineal (LP), Evolución Diferencial (DE), Colonia Artificial de Abejas (ABC), Algoritmos Genéticos (GA) y Optimización con Colonia de Hormigas (ACO).	72
4.5. Diámetros de tuberías de la solución obtenida con Algoritmos Genéticos (GA) para la red de dos circuitos, comparados con los reportados en la literatura obtenidos con Algoritmo de Transición de Estado (STA).	74

4.6. Soluciones para el dimensionamiento de la red de dos circuitos obtenidas con Algoritmos Genéticos	75
4.7. Comparación de la solución obtenida con Algoritmos Genéticos (GA) para la red de dos circuitos, con las reportadas usando Gradiente de Programación Lineal (LPG) y Algoritmo de Transición de Estado (STA). . .	77
4.8. Diámetros de tuberías en <i>mm</i> de la solución obtenida con Algoritmos Genéticos para la red de 3×3 , comparados con los reportados en la literatura.	80
4.9. Soluciones para el dimensionamiento de la red de 3×3 obtenidas con Algoritmos Genéticos.	81
4.10. Comparación del costo de la solución obtenida con las reportadas en la literatura para el dimensionamiento de la red de 3×3 . Las soluciones se obtuvieron con Algoritmos Genéticos (GA) y con la técnica <i>Non-dominated Sorting Genetic Algorithm</i> (NSGA-II).	82

RESUMEN

Optimización del diseño de redes de riego con algoritmos evolutivos y bio-inspirados ¹Jiménez-Hernández, Eduardo²; López-Cruz, Irineo L.³

La planeación general de una red de distribución de agua consiste de tres fases: *a)* trazo, *b)* dimensionamiento, y *c)* operación; y pesar de que cada fase es dependiente de las otras, éstas se pueden formular y resolver como problemas independientes. El trazo consiste en determinar la topología de la red, mientras que el dimensionamiento implica encontrar los diámetros de las tuberías que generen la red de menor costo. En la presente investigación, el trazo y dimensionamiento de redes de tuberías para riego se plantearon como problemas de optimización combinatoria de variables discretas con restricciones. Para resolver estos problemas se usaron las metaheurísticas: Algoritmos Genéticos con codificación entera (GA por sus siglas en inglés de Genetic Algorithms) y Optimización con Colonia de Hormigas (ACO por sus siglas en inglés de Ant Colony Optimization). Además, se utilizó el programa computacional EPANET 2 para verificar el funcionamiento hidráulico de las soluciones generadas. Ambos algoritmos evolutivos y bio-inspirados se utilizaron para optimizar las fases del diseño de redes de riego usando un enfoque secuencial. El desempeño de estos algoritmos evaluó con la solución de redes de referencia usadas en la literatura. Se obtuvieron soluciones aceptables de manera eficiente con ambos algoritmos. La mayoría de las soluciones fueron iguales o muy cercanas a las mejores reportadas hasta el momento. El GA resultó ser más eficiente para obtener soluciones para el problema de dimensionamiento, mientras que ACO fue más eficiente para encontrar soluciones al trazo. Los resultados sugieren que es necesario solucionar los problemas de trazo y dimensionamiento de manera simultánea para obtener redes hidráulicamente aceptables y de costo mínimo.

Palabras clave: sistema de distribución de agua, red de riego, algoritmos genéticos, optimización con colonia de hormigas, EPANET 2

¹Tesis de maestría en Ingeniería. Posgrado en Ingeniería Agrícola y Uso Integral del Agua, Universidad Autónoma Chapingo.

²Autor

³Director de tesis

ABSTRACT

Optimal design of irrigation networks with evolutionary and bio-inspired algorithms¹Jiménez-Hernández, Eduardo²; López-Cruz, Irineo L.³

The overall planning process of water distribution systems consists of three phases: *a)* layout, *b)* sizing, and *c)* operation; although they are dependent on each other, they can be formulated as independent problems. The layout work involves determining the network topology, whereas sizing consists in finding the pipe diameters that give the minimum cost network. In the current research the layout and pipe sizing of irrigation networks were formulated as combinatorial optimization problems of discrete variables with constraints. In order to solve these problems, Genetic Algorithms (GA) and Ant Colony Optimization (ACO) metaheuristics were used. EPANET 2 software was used to verify the hydraulic operation of the generated solutions. Both evolutionary and bio-inspired algorithms were used to optimize the design phases of irrigation networks following a sequential approach. Their performance was tested by solving benchmark networks used commonly in the literature. Acceptable solutions were found efficiently with both algorithms. Most of these solutions were equal to or close enough to the best reported so far in the literature. The GA was more efficient in finding solutions for the sizing problem, whereas the ACO algorithm was more efficient in finding solutions for the layout problem. Results suggest that it is necessary to solve both the layout and sizing problems simultaneously in order to obtain hydraulically acceptable and minimal cost networks.

Keywords: water distribution system, irrigation network, genetic algorithms, ant colony optimization, EPANET 2

¹Master thesis. Posgrado en Ingeniería Agrícola y Uso Integral del Agua, Universidad Autónoma Chapingo.

²Author

³Advisor

Lista abreviaturas y acrónimos

ABC	Colonia Artificial de Abejas (del inglés Artificial Bee Colony)
ACO	Optimización con Colonia de Hormigas (del inglés Ant Colony Optimization)
AS	Ant System
DE	Evolución Diferencial (del inglés Differential Evolution)
DR	Distrito de Riego
EA	Algoritmo Evolutivo (del inglés Evolutionary Algorithm)
GA	Algoritmos Genéticos (del inglés Genetic Algorithms)
LP	Programación Lineal (del inglés Linear Programming)
LPG	Gradiente de Programación Lineal (del inglés Linear Programming Gradient)
MMAS	Max-Min Ant System
NSGA-II	(Fast Elitist) Nondominated Sorting Genetic Algorithm II
PSO	Optimización con Enjambre de Partículas (del inglés Particle Swarm Optimization)
STA	algoritmo de Transición de Estado (del inglés State Transition Algorithm)
UR	Unidad de Riego

Capítulo 1

Introducción

1.1. Breve descripción del estado actual del uso del agua en México

En la actualidad existe una preocupación generalizada por el suministro de agua, ya que la visión que se tiene a futuro de este recurso es de escasez. Además el crecimiento demográfico acelerado demanda incrementar la productividad agrícola, lo que, entre otras cosas, implica hacer un uso eficiente del agua que se destina a este sector. En México, y a nivel mundial, las actividades agrícolas consumen alrededor del 75 % del agua disponible, lo que refleja la gran magnitud de la situación a la que se debe dar atención.

Anualmente México recibe aproximadamente 1,449,471 millones de metros cúbicos de agua en forma de precipitación. De esta agua, se estima que el 72.5 % se evapotranspira y regresa a la atmósfera, el 21.2 % escurre por los ríos o arroyos, y el 6.3 % restante se infiltra al subsuelo de forma natural y recarga los acuíferos. Tomando en cuenta los flujos de salida (exportaciones) y de entrada (importaciones) de agua con los países vecinos, el país cuenta anualmente con 446,777 millones de metros cúbicos de agua dulce renovable (CONAGUA, 2016).

A grandes rasgos se puede decir que tres cuartas partes del agua disponible que se tiene en el país se destina para actividades agrícolas. En la figura (1.1), se observa que el mayor volumen concesionado para usos consuntivos por la Comisión Nacional del Agua, se destina para el uso agrupado agrícola, que es principalmente para riego (CONAGUA, 2016).

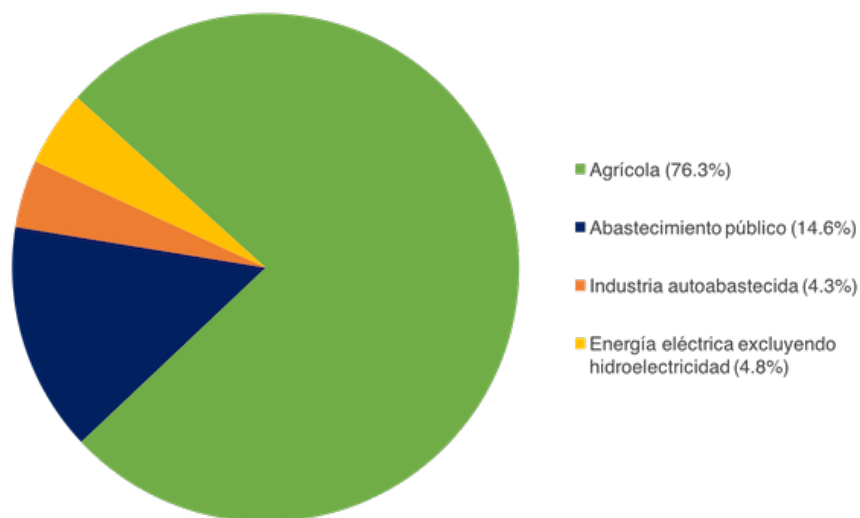


FIGURA 1.1: Distribución del uso del agua concesionada en México en el año 2015 (CONAGUA, 2016)

México ocupa el séptimo lugar mundial en términos de superficie con infraestructura de riego con 6.5 millones de hectáreas, de las cuales un poco más de la mitad corresponde a 86 distritos de riego, y el restante a más de 40 mil unidades de riego. Para 2015, la superficie cuantificada en unidades agrícolas de producción fue de 30.2 millones de hectáreas, de las cuales 18 % eran de riego y el resto tenían régimen de temporal. La agricultura de riego en el país tiene un rendimiento en toneladas por hectárea de 2.2 a 3.3 veces mayor que la superficie en régimen de temporal (CONAGUA, 2016), lo que refleja la importancia de mejorar los sistemas de producción ya que se estima que el crecimiento de la población demandará una mayor cantidad de alimento en un futuro cercano.

En el entorno actual es previsible la disminución de la disponibilidad del agua debido al cambio climático, por ello es imperativo elevar la eficiencia con la que se usa dicho recurso. La productividad del agua en los Distritos de Riego (DR) es un indicador clave para evaluar la eficiencia con la que se utiliza el agua para la producción de alimentos, que depende de la conducción desde la fuente de abastecimiento hasta las parcelas y su utilización. Pedroza González (2014) menciona que en México, el único indicador de desempeño en la distribución del agua en los DR es la *eficiencia*; aunque también se

calculan otros indicadores como la eficiencia global del distrito y la eficiencia de aplicación, pero éstas últimas no involucran directamente el proceso de distribución. En lo que respecta a valores de eficiencia, el autor hace una recopilación de eficiencias de conducción y distribución del agua de 19 DR del país, obteniendo un promedio de 58.49 %. Cabe señalar que dicha información se generó hace varias décadas por la antigua Secretaría de Agricultura y Recursos Hidráulicos, sin embargo, la razón por la que aún es frecuente utilizar estos valores como referencia es la carencia de información más reciente.

Según Román López et al. (2013), el riego por gravedad se utiliza en más del 90 % de la superficie con agricultura de riego, y el 10 % de la superficie utiliza sistemas de riego presurizado. Esto tiene un origen histórico, ya que en la época que se diseñaron y crearon los DR y UR del país, la tecnología prevaleciente para la aplicación del agua en las parcelas era por gravedad. Estos métodos de riego tienen una eficiencia muy baja y en consecuencia se tienen grandes pérdidas de agua. Además, en muchos casos sólo se construyeron las redes de canales y drenes principales, quedando las obras parcelarias a cargo de los usuarios. Lo anterior, sumado al deterioro de la infraestructura, acumulado en varias décadas por la insuficiencia de recursos económicos para su conservación y mejoramiento, propiciaron una baja en la eficiencia global del manejo del agua.

Palacios Vélez (2004) estima que en promedio en los DR del país se pierde un 40 % del agua en la conducción; es decir la eficiencia media de conducción es del orden del 60 %. Según Peña Peña (2007) la eficiencia de conducción en los DR del país era de 64.7 % para el año 2000. Mientras que Román López et al. (2013) señala que para el método de riego por gravedad se estima una eficiencia global de riego de 47 %, lo cual significa que se pierde el 53 % durante su conducción, distribución y aplicación. Estudios de caso particulares sugieren diferentes valores de eficiencias de uso del agua de riego, como es el caso de García Domínguez et al. (2010) que para el Módulo de riego IV del DR 017 Comarca Lagunera, sugieren un valor máximo del 82.18 % para la eficiencia de conducción en las laterales y sublaterales, y una eficiencia de aplicación a nivel parcelario del 61.37 %; mientras que Ortega-Gaucin et al. (2009) sugieren un valor de 60.1 % para la eficiencia de conducción del DR 005 Delicias, Chihuahua.

Se puede observar que la información existente es obsoleta e incompleta, además

únicamente se tienen datos para algunos DR y no son suficientes para ser representativos de la superficie de riego del país que está conformada por distritos y unidades de riego. Estos valores, sugieren que aún queda mucho trabajo por hacer lo que respecta al incremento de la eficiencia y la productividad del agua en el sector agrícola.

Al respecto, Peña Peña (2007) menciona que revistiendo y entubando canales, y tecnificando el riego a nivel parcelario, se puede mejorar la eficiencia de conducción, y tecnificando el riego a nivel parcelario se mejora la eficiencia parcelaria. Para el efecto, además de las inversiones necesarias en infraestructura, se requiere mejorar las capacidades y las habilidades de los operadores de las obras y de los sistemas correspondientes. En el mismo sentido Román López et al. (2013) afirma que para que los sistemas de riego presurizado (aspersión o goteo) puedan lograr su eficiencia global esperada deben ser bien diseñados y posteriormente operados adecuadamente. Aunque con el tiempo ya se han estado implementado algunas de estas acciones para solucionar el problema (principalmente por instituciones gubernamentales), sigue siendo necesario intensificar y mejorar la manera en que se realizan.

1.2. Motivación

En este contexto, la optimización del diseño de las redes de riego es fundamental. En primer lugar la tecnificación de sistemas involucra tanto obras de gran tamaño, como redes de conducción y distribución, así como redes parcelarias más pequeñas. Esto generalmente requiere de una inversión de capital muy grande, por lo que si se considera desde un principio instalar una red con un diseño óptimo, se asegura que los recursos se están aprovechando de manera eficiente. Por otro lado, la optimización del diseño de una red de riego tiene la finalidad de minimizar su costo de construcción, sin embargo, también se abaratan otros costos de manera indirecta, entre ellos están los costos de uso de energía eléctrica (generados por bombeo), costos de operación, mantenimiento, futuras reparaciones, etc., cuyo estudio está fuera del alcance de este trabajo.

Eiger et al. (1994) mencionan que el proceso de planeación de una red de distribución de agua consiste de tres fases: *a)* trazo, *b)* dimensionamiento, y *c)* operación; y a pesar de que cada fase es dependiente de las otras, éstas se pueden formular y resolver

como problemas independientes. Entonces el procedimiento de diseño de redes se lleva a cabo mediante iteraciones de estas tres fases. El diseño de redes de tuberías en general es un problema sobre el que se ha realizado mucha investigación con el objetivo de minimizar los costos de su construcción, instalación, rehabilitación, etc.; sobre todo por su importancia en la distribución y suministro de agua potable en centros urbanos. Sin embargo, aunque el ahorro debido a la optimización del trazo de las redes es generalmente mayor al que se obtiene por el dimensionamiento de sus componentes, la mayor parte de la investigación se ha enfocado en éste último, que es el problema más fácil. Una de las razones por las que el dimensionamiento de redes es el problema sobre el que existe más investigación, es debido a que generalmente se estudian redes de uso urbano en las que su trazo es conocido previamente, sin embargo en las redes de riego es común que se tengan que resolver ambos problemas.

Generalmente el diseño de sistemas de distribución de agua se hace planteando un problema para cada una de sus fases y se soluciona uno a la vez de manera secuencial. Para una red de tuberías primero se plantea y soluciona un problema para su trazo y posteriormente se hace la determinación de los diámetros de sus tuberías.

Dado que cada una de las fases se pueden plantear como problemas de optimización en los que se obtienen soluciones a prueba y error, es común que se utilicen métodos heurísticos para resolverlos. Uno de los métodos más utilizados extensivamente, sobre todo para el dimensionamiento de redes, es Algoritmos Genéticos cuya naturaleza permite obtener la mejor red dentro de un conjunto de propuestas generadas al azar. Una tendencia reciente consiste en el diseño de redes de distribución de agua como un problema multiobjetivo donde se resuelven las fases del diseño de manera simultánea, lo que también es posible solucionar usando las metaheurísticas apropiadas.

Este trabajo se centró en el planteamiento y solución de ciertos problemas de diseño de redes de distribución de agua como problemas matemáticos de optimización. Para ello se utilizaron algunas técnicas de optimización global, conocidas en investigación de operaciones como metaheurísticas. No se utilizaron técnicas basadas en gradiente puesto que se sabe que convergen a mínimos locales, en cambio se le dio preferencia a algoritmos de computación evolutiva y bio-inspirados.

El objetivo de la investigación fue plantear el problema de optimización del diseño de redes de riego y obtener soluciones usando metaheurísticas de optimización. Para

ello se requirió de los siguientes objetivos particulares:

- Plantear las fases del diseño de redes de riego (el trazo y dimensionamiento), como problemas de optimización combinatoria de variables discretas con restricciones;
- Solucionar los problemas planteados mediante: *a)* Algoritmos Genéticos y *b)* Optimización con Colonia de Hormigas; y
- Generar una herramienta de cómputo para apoyar en el diseño óptimo de redes de riego.

En el capítulo 2 se presenta una revisión de los trabajos realizados en el diseño de redes para riego y algunos puntos que se retoman del problema de diseño de redes de agua potable puesto que, aunque son problemas diferentes, tienen algunos aspectos en común. En el capítulo 3 Materiales y métodos, se menciona a detalle la manera en la que se plantearon los problemas de optimización del diseño de redes de riego en los dos enfoques de trabajo y las técnicas heurísticas que se utilizaron para resolver estos problemas. En el capítulo 4 Resultados y discusión, se hace una comparación y análisis de los resultados obtenidos para los problemas planteados. En el capítulo 5 Conclusiones, se mencionan las ideas con las que se concluye el trabajo, sus principales hallazgos e ideas para continuar avanzando este trabajo. Finalmente se Anexa el artículo científico realizado para la investigación, donde se plantea el problema, se describe la metodología y se presentan los resultados y discusión.

Capítulo 2

Revisión de Literatura

2.1. Optimización

Matemáticamente, es posible escribir la mayor parte de los problemas de optimización en la forma genérica:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f_i(\mathbf{x}), \quad (i = 1, 2, \dots, M), \quad (2.1)$$

sujeto a

$$h_j(\mathbf{x}) = 0, \quad (j = 1, 2, \dots, J), \quad (2.2)$$

$$g_k(\mathbf{x}) \leq 0, \quad (k = 1, 2, \dots, K), \quad (2.3)$$

donde $f_i(\mathbf{x})$, $h_j(\mathbf{x})$ y $g_k(\mathbf{x})$ son funciones del vector de diseño

$$\mathbf{x} = (x_1, x_2, \dots, x_d)^T \quad (2.4)$$

Los componentes x_i de $\mathbf{x}$ se llaman *variables de decisión* o de *diseño*, y éstas pueden ser continuas, discretas o una combinación de ambas (Yang, 2014).

Las funciones $f_i(\mathbf{x})$ donde $i = 1, 2, \dots, M$ se llaman *funciones objetivo* o simplemente *funciones de costo* y en el caso de $M = 1$ hay solo un objetivo. El espacio abarcado por las variables de decisión se llama *espacio de diseño* o *espacio de búsqueda* $\mathbb{R}^d$, mientras que el espacio formado por los valores de la función objetivo se llama *espacio de solución* o

espacio de respuesta. Las igualdades h_j y las desigualdades g_k se llaman restricciones. Cabe mencionar que las desigualdades también se pueden escribir en el sentido contrario, ≥ 0 , y que los objetivos también se pueden formular como un problema de maximización. De acuerdo al número de objetivos los problemas de optimización se pueden clasificar en dos categorías: de un objetivo $M = 1$ y multiobjetivo $M > 1$. En la literatura a la optimización multiobjetivo también se le llama optimización multicriterio o incluso multiatributo. En los problemas del mundo real la mayor parte de las tareas de optimización son multiobjetivo (Yang, 2014).

Los problemas de optimización se dividen de manera natural en dos categorías: problemas de optimización con variables continuas y problemas de optimización con variables discretas. A éstos últimos se les llama problemas de optimización combinatoria. En problemas de tipo continuo, se busca la solución sobre un conjunto de números reales con ciertas propiedades de continuidad y generalmente de convexidad; en los problemas de optimización combinatoria, se busca la solución sobre un conjunto finito o infinito numerable de objetos para los que se pueda definir una función que evalúe la “calidad” de cada objeto. Estas dos clases de problemas tienen diferentes retos y los métodos para resolverlos son distintos (De los Cobos Silva et al., 2010).

2.1.1. Optimización continua

Para el caso de minimización en optimización continua, el problema es encontrar $x_{opt} \in S$ que satisfaga:

$$f(x_{opt}) \leq f(x) \quad \forall x \in S \quad (2.5)$$

donde S es el conjunto de todas las soluciones posibles; y en el caso de la maximización, consiste en encontrar $x_{opt} \in S$ que satisfaga:

$$f(x_{opt}) \geq f(x) \quad \forall x \in S \quad (2.6)$$

A la solución x_{opt} se le llama una solución globalmente óptima y $f_{opt} = f(x_{opt})$ denota el costo óptimo, mientras que S_{opt} denota el conjunto de soluciones óptimas.

Dado un problema de optimización continua se define una vecindad para $x \in S$ con radio $r \in \mathbb{R}^+$ como:

$$Br(x) = \{y \in S \mid \|x - y\| < r\} \quad (2.7)$$

donde $\|\cdot\|$ es alguna norma definida en S .

Sea un problema de optimización continua y $S_{\bar{x}}$ una vecindad de $\bar{x}$, entonces $\bar{x}$ se llama una solución óptima local o simplemente un óptimo local con respecto a $S_{\bar{x}}$ si $\bar{x}$ es mejor que o igual a, todas sus soluciones vecinas con respecto a la función costo. Específicamente, en el caso de minimización $\bar{x}$ se llama solución mínima local o simplemente un mínimo local si

$$f(\bar{x}) \leq f(y), \quad y \in S_{\bar{x}} \quad (2.8)$$

y en el caso de maximización, $\bar{x}$ se llama una solución máxima local o simplemente un máximo local si

$$f(\bar{x}) \geq f(y), \quad y \in S_{\bar{x}} \quad (2.9)$$

2.1.2. Optimización combinatoria

Una instancia de un problema de optimización combinatoria puede formalizarse como una pareja (S, f) , donde S denota el conjunto finito de todas las soluciones posibles y f es una función real, la función de costo o función objetivo, mapeo definido por

$$f : S \rightarrow \mathbb{R} \quad (2.10)$$

Para el caso de minimización, el problema es encontrar $i_{opt} \in S$ que satisfaga:

$$f(i_{opt}) \leq f(i) \quad \forall i \in S \quad (2.11)$$

en el caso de la maximización, la $i_{opt} \in S$ que satisfaga

$$f(i_{opt}) \geq f(i) \quad \forall i \in S \quad (2.12)$$

A la solución i_{opt} se le llama una solución globalmente óptima y $f_{opt} = f(i_{opt})$ denota el costo óptimo, mientras que S_{opt} denota el conjunto de soluciones óptimas. Un problema de optimización combinatoria es un conjunto de I instancias.

2.1.3. Optimización multiobjetivo

El problema de optimización multiobjetivo (también llamado problema de optimización multicriterio o vectorial) se puede definir como el problema de encontrar (Osy-czka, 1985):

“un vector de variables de decisión que satisfaga un cierto conjunto de restricciones y optimice un vector de funciones cuyos elementos representan las funciones objetivo. Estas funciones forman una descripción matemática de los criterios de desempeño que suelen estar en conflicto unos con otros. En este sentido, el término “optimizar” significa encontrar una solución tal que al tomador de decisiones se le proporcione un conjunto de valores aceptables para todos los objetivos”

Formalmente un problema de optimización multiobjetivo se puede definir como (Coello Coello (2001); Coello Coello et al. (2007)) encontrar el vector $\mathbf{x}^* = [x_1^*, x_2^*, \dots, x_n^*]$ que satisfaga las m restricciones de desigualdad:

$$g_i(\mathbf{x}) \leq 0, \quad i = \{1, \dots, m\} \quad (2.13)$$

las p restricciones de igualdad

$$h_j(\mathbf{x}) = 0, \quad j = \{1, \dots, p\} \quad (2.14)$$

y optimice el vector de funciones:

$$F(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_k(\mathbf{x})]^T \quad (2.15)$$

donde $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ es el vector de variables de decisión.

En otras palabras, se desea determinar a partir del conjunto $\mathcal{F}$ de todos los números que satisfacen a las restricciones (2.13) y (2.14), el conjunto particular $x_1^*, x_2^*, \dots, x_n^*$ que genera valores óptimos en todas las funciones objetivo.

Al tener más de una función objetivo, la noción de “óptimo” cambia, porque en problemas multiobjetivo la finalidad es encontrar buenos compromisos (en inglés “trade-offs”) más que una solución única como en optimización global. La noción de “óptimo” más adoptada es la originalmente por Francis Ysidro Edgeworth y posteriormente generalizada por Vilfredo Pareto. Aunque algunos autores llaman a esta noción *óptimo de Edgeworth-Pareto*, el término más comúnmente aceptado es *óptimo de Pareto*. La definición formal se presenta a continuación (Coello Coello et al., 2007):

Se dice que una solución $\mathbf{x} \in \Omega$ es un óptimo de Pareto con respecto a Ω si y solo si no existe ninguna solución $\mathbf{x}' \in \Omega$ para la cual $\mathbf{v} = F(\mathbf{x}') = (f_1(\mathbf{x}'), \dots, f_k(\mathbf{x}'))$, que domine a $\mathbf{u} = F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$. La frase óptimo de Pareto se toma con respecto al espacio completo de las variables de decisión a menos que se especifique otra cosa.

Se dice que un vector $\mathbf{u} = (u_1, \dots, u_k)$ domina a otro vector $\mathbf{v} = (v_1, \dots, v_k)$ (denotado por $\mathbf{u} \preceq \mathbf{v}$) si y solo si $\mathbf{u}$ es parcialmente menor que $\mathbf{v}$, por ejemplo, $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$.

2.2. Métodos de optimización

Coello Coello et al. (2007), clasifican de manera muy general las diferentes técnicas de búsqueda y optimización en tres categorías: enumerativas, determinísticas y estocásticas (aleatorias). Aunque una búsqueda enumerativa es determinística se hace una distinción ya que ésta no emplea heurísticas. Algunos ejemplos comunes se muestran en la figura (2.1)

Como ya se mencionó la optimización se puede clasificar como optimización discreta o continua, dependiendo del tipo de variables que se utilicen. Estas variables pueden tener límites o restricciones. Además la optimización puede ser un problema estático o dinámico dependiendo si las salidas están en función del tiempo. Tradicionalmente la optimización se resuelve con métodos basados en cálculo o métodos de búsqueda (al azar o enumerativa) (Du y Swamy, 2016).

Las técnicas de optimización generalmente se dividen en métodos derivativos y métodos no derivativos, dependiendo si se requieren o no las derivadas de la función objetivo para el cálculo del óptimo. Los métodos derivativos son basados en cálculo, los cuales pueden ser métodos de búsqueda con gradiente o métodos de segundo orden.

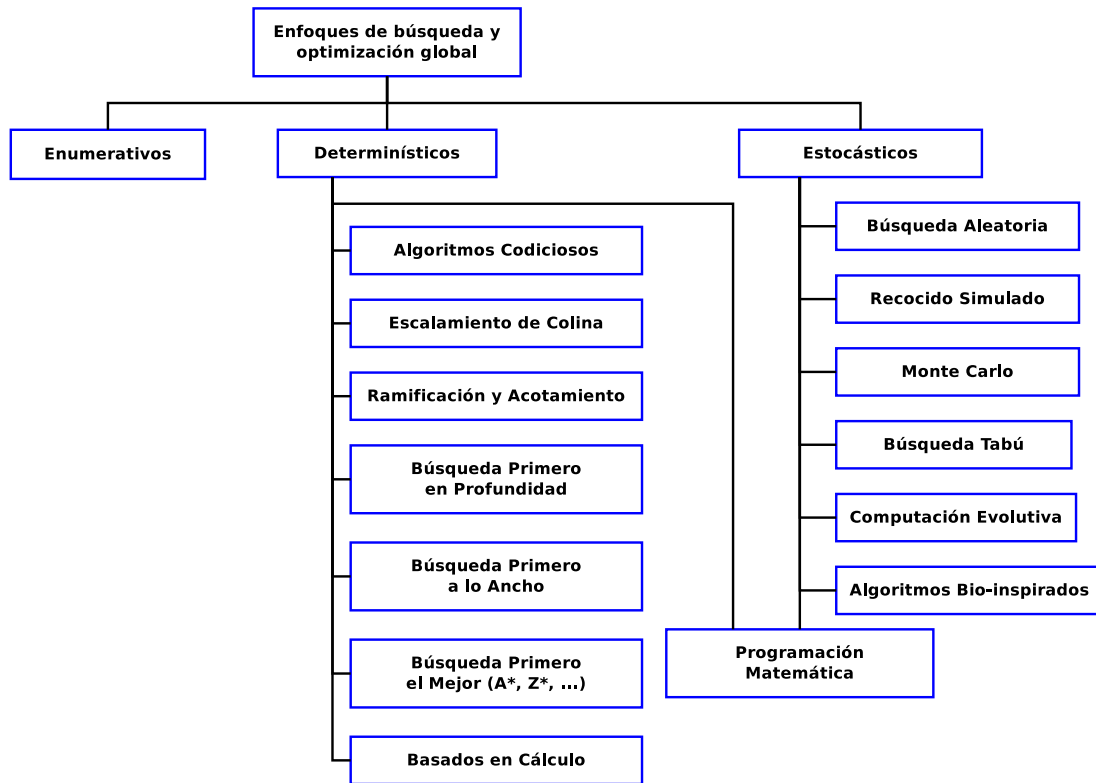


FIGURA 2.1: Diferentes enfoques para optimización global (Modificado de Coello Coello et al. (2007))

A éstos se les conoce como métodos de optimización local. El método del gradiente, también conocido como descenso más pronunciado, busca un mínimo local tomando pasos sobre la dirección negativa del gradiente de la función. Algunos ejemplos de métodos de segundo orden son el método de Newton, el método de Gauss-Newton, métodos quasi-Newton, el método de la región de confianza y el método de Levenberg-Marquardt. Los métodos de gradiente conjugado y métodos de gradiente natural también pueden ser clasificados como formas reducidas del método quasi-Newton (Du y Swamy, 2016).

Los métodos que no requieren información del gradiente para realizar una búsqueda y explorar el espacio de solución de manera consecutiva se llaman *métodos de búsqueda directa*. Éstos mantienen un grupo de soluciones (puntos) y utilizan algún tipo de exploración determinística para buscar dentro del espacio y casi siempre utilizan un método codicioso para actualizar el grupo de soluciones. La búsqueda simplex y

búsqueda de patrones son dos ejemplos efectivos de métodos de búsqueda directa (Du y Swamy, 2016).

Algunos métodos típicos para funciones multivariadas, que no usan la derivada de la función objetivo, son escalamiento de colina con reinicio al azar, búsqueda al azar, así como una gran cantidad de métodos heurísticos y metaheurísticas, y sus híbridos. El escalamiento de colina es un intento por optimizar el óptimo local de una función discreta o continua, cuando opera sobre el espacio continuo se llama gradiente ascendente. Otros métodos de búsqueda no derivativos incluyen la búsqueda univariada paralela a un eje (p.e. método de búsqueda coordinado), método simplex secuencial y métodos de aceleración en búsqueda directa tales como el método de Hooke-Jeeves, el método de Powell y el método de Rosenbrock. Los métodos de punto interior son técnicas que representan el estado del arte para solucionar problemas de optimización lineales, cuadráticos y no lineales (Du y Swamy, 2016).

2.2.1. Complejidad computacional

Un problema relacionado con la eficiencia y la eficacia de un algoritmo es la dificultad del problema que se pretende resolver. Para estudiar la dificultad de un problema de optimización, primero se debe transformar en un problema de decisión (problemas cuya respuesta puede ser *si* o *no*). Los problemas que se pueden resolver usando un algoritmo de tiempo polinomial son *tratables*. Un algoritmo de tiempo polinomial tiene un límite superior $O(n^k)$ en su tiempo de ejecución, donde k es una constante y n es el tamaño del problema (tamaño de entrada). Los problemas tratables generalmente son fáciles de resolver porque el tiempo de ejecución se incrementa poco a poco conforme aumenta el valor de n . En contraste, se dice que los problemas son intratables si no se pueden resolver con un algoritmo de tiempo polinomial y si existe un límite inferior en el tiempo de ejecución $\Omega(k^n)$, donde $k > 1$ es una constante y n es el tamaño de entrada (Du y Swamy, 2016).

La clase de complejidad P (que significa complejidad de tiempo polinomial) se define como el conjunto de problemas de decisión que se pueden resolver con una máquina de Turing determinista usando un algoritmo de complejidad que, en el peor de los casos, es de tiempo polinomial. Usualmente, los problemas de la clase P son fáciles ya que existen algoritmos que los resuelven en tiempo polinomial.

La clase NP (que significa complejidad de tiempo polinomial no determinista) es el conjunto de todos los problemas de decisión que pueden verificarse, en el peor de los casos, por una máquina de Turing no determinista usando un algoritmo no determinista de tiempo polinomial. Aunque los algoritmos no deterministas no pueden ejecutarse directamente en computadoras convencionales, este concepto es importante y útil para el análisis de complejidad computacional. Todos los problemas en P también pertenecen a la clase NP, p.e. $P \subseteq NP$. También existen problemas donde las soluciones correctas no pueden verificarse en tiempo polinomial (Du y Swamy, 2016).

Todos los problemas de decisión en P son tratables. Aquellos problemas que están en NP, pero no en P, son difíciles ya que no se conocen algoritmos de tiempo polinomial para ellos y generalmente se cree que no existen, aunque la pregunta ¿ $P=NP$?, es decir, si las dos clases de problemas coinciden, es considerada una de las más importantes en la teoría de ciencias computación (De los Cobos Silva et al., 2010). Existen problemas en NP donde no hay algoritmos polinomiales disponibles, pero que con esfuerzo polinomial pueden transformarse en otro. Se dice que un problema es NP-Difícil, si un algoritmo para resolver este problema es reducible en tiempo polinomial a un algoritmo que es capaz de resolver cualquier problema en NP. Por eso, los problemas NP-Difíciles son al menos tan difíciles como cualquier otro problema en NP, pero no necesariamente son NP (Du y Swamy, 2016).

El conjunto de problemas NP-Completo es un subconjunto de NP. Se dice que un problema de decisión A es NP-Completo, si A es NP y NP-Difícil. Los problemas NP-Completo son los más difíciles en NP. Son difíciles ya que no se conocen algoritmos de tiempo polinomial para ellos. Los problemas de decisión que no están en NP son incluso más difíciles. La relación entre todas estas clases se presenta en la figura 2.2.

Finalmente, se puede decir que los problemas de optimización difíciles son problemas que no se pueden resolver de manera óptima, por ningún método exacto (determinístico) dentro de un límite de tiempo “razonable”. Para encontrar soluciones satisfactorias a estos problemas se pueden usar metaheurísticas. Una metaheurística es un algoritmo diseñado para resolver de manera aproximada un amplio rango de problemas de optimización difíciles, sin la necesidad de adaptarse específicamente para cada problema. Las metaheurísticas se aplican generalmente a problemas para los que

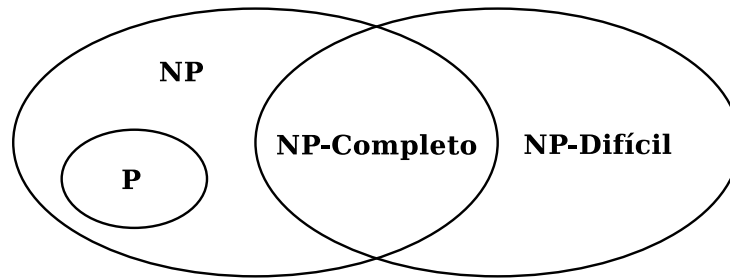


FIGURA 2.2: La relación entre las clases de problemas P, NP, NP-Difícil y NP-Completo

no existen algoritmos específicos para resolverlos de manera satisfactoria, son ampliamente usadas para resolver problemas complejos en la industria y servicios, en áreas que van desde las finanzas hasta ingeniería (Boussaïd et al., 2013).

2.2.2. Heurísticas, metaheurísticas e hyper-heurísticas

De los Cobos Silva et al. (2010) mencionan que la palabra heurística proviene del griego *euriskein*, al igual la famosa palabra *eureka* y quiere decir encontrar. Además de brindar el significado etimológico señala que la palabra heurística se utiliza para caracterizar a las técnicas por las cuales se encuentra o se mejora la solución de un problema intratable. Se utilizan algoritmos heurísticos para obtener “buenas soluciones” (soluciones subóptimas o cercanas a la óptima global) de problemas cuyos algoritmos exactos de solución no son factibles en tiempos polinomiales. La palabra metaheurística, combina el prefijo meta del griego (que significa “más allá”, “a un nivel más alto”) con la palabra heurística, en otras palabras, se puede decir que metaheurística se refiere a una estrategia de nivel superior que guía a otras heurísticas para buscar soluciones factibles para problemas complejos.

Las metaheurísticas pueden ser: basadas en una solución o basadas en una población de soluciones. Las primeras se basan en tener una solución única en cualquier tiempo y comprenden metaheurísticas de búsqueda local tales como recocido simulado, búsqueda tabú, búsqueda local iterada, búsqueda local guiada, búsqueda de patrones o búsqueda aleatoria, algoritmo de Solis-Wets y búsqueda de vecindario variable. En las metaheurísticas basadas en poblaciones, un número de soluciones se actualiza de manera iterativa hasta que se satisface un criterio de término, generalmente se clasifican

en Algoritmos Evolutivos y algoritmos de inteligencia de enjambres. Las metaheurísticas basadas en una solución están más orientadas a la explotación de una solución única, mientras que las basadas en poblaciones están más orientadas a la exploración del espacio de búsqueda (Boussaïd et al. (2013); Du y Swamy (2016)).

La idea de una hyper-heurística se puede rastrear hasta la década de 1960. Las hyper-heurísticas se pueden concebir como heurísticas para elegir otras heurísticas o como algoritmos de búsqueda que exploran el espacio de métodos disponibles para solucionar problemas. Una hyper-heurística es un método heurístico que busca automatizar el proceso de seleccionar, combinar, generar o adaptar varias heurísticas más simples para resolver eficientemente problemas de búsqueda difíciles. Las heurísticas de bajo nivel son operadores simples de búsqueda local o heurísticas dependientes del dominio que operan directamente en el espacio de solución para una instancia de un problema dada (Du y Swamy, 2016).

Actualmente la selección y generación de heurísticas son los principales enfoques de operación de las hyper-heurísticas. En el primer método, la hyper-heurística selecciona de un conjunto de heurísticas conocidas de bajo nivel y dependientes del dominio. En el segundo método la hyper-heurística genera nuevas heurísticas de bajo nivel evolucionando los componentes de heurísticas existentes. Las hyper-heurísticas pueden basarse en programación genética o en evolución gramatical, que son excelentes candidatos para generación de heurísticas (Du y Swamy, 2016).

2.2.3. Computación evolutiva

Brabazon et al. (2015) mencionan que uno de los primeros trabajos publicados en la implementación de un algoritmo de tipo evolutivo fue el realizado por Friedberg en 1958, estos estudios fueron compilados en una serie de artículos que posteriormente dieron origen a la computación evolutiva. El trabajo de Friedberg representa a lo que hoy se conoce como Programación Genética, y que posteriormente se popularizó con el trabajo de Cramer, Dickmans y Koza en la década de 1980. Durante las décadas de 1960 y 1970 dos líneas de investigación significantes e independientes se desarrollaron en Europa y los EE.UU. Los europeos desarrollaron Estrategias Evolutivas (Rechenberg (1965); Rechenberg (1973)) y los norteamericanos desarrollaron Programación Evolutiva (Fogel et al., 1966) y Algoritmos Genéticos (Holland, 1975). Recientemente Storn

y Price (1997) agregaron Evolución Diferencial a la familia de computación evolutiva. El esquema completo de los métodos que integran la computación evolutiva se presenta en la figura 2.3.

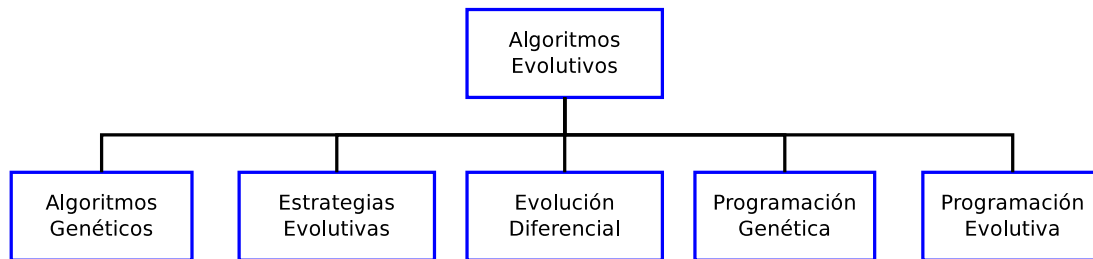


FIGURA 2.3: Principales métodos de la computación evolutiva (Brabazon et al., 2015)

La idea principal detrás de un algoritmo evolutivo es la siguiente: dada una población de individuos dentro de un ambiente que tiene recursos limitados, la competencia por dichos recursos causará selección natural (supervivencia del más apto). Lo que en general incrementa de la adaptación de la población. Dada una función de calidad a ser maximizada se puede crear un conjunto al azar de de soluciones candidatas, por ejemplo, elementos dentro del dominio de la función, y entonces se puede aplicar la función de calidad como una medida abstracta de la aptitud de las soluciones, entre más alta mejor. Tomando en cuenta los valores de aptitud se seleccionan algunos de los mejores candidatos para procrear la siguiente generación. Esto se hace mediante la aplicación de recombinación y mutación. La recombinación es un operador que se aplica a dos o más candidatos previamente seleccionados (llamados padres), que produce uno o más nuevos candidatos (llamados hijos). La mutación se aplica a un candidato y resulta un nuevo candidato. Por consiguiente ejecutar las operaciones de recombinación y mutación sobre los padres lleva a la creación de un conjunto nuevo de candidatos (los descendientes). Éstos últimos son evaluados con respecto a su aptitud y entonces compiten con los viejos por un lugar en la siguiente generación. Este proceso se puede iterar hasta que se encuentre un candidato con la calidad suficiente (una solución) o hasta que se alcance un límite computacional previamente establecido (Eiben y Smith, 2015). En la figura 2.4 se muestra el esquema general de un algoritmo evolutivo.

Hay dos fuerzas principales que forman la base de un sistema evolutivo: *a)* los operadores de variación (recombinación y mutación) crean la diversidad necesaria dentro

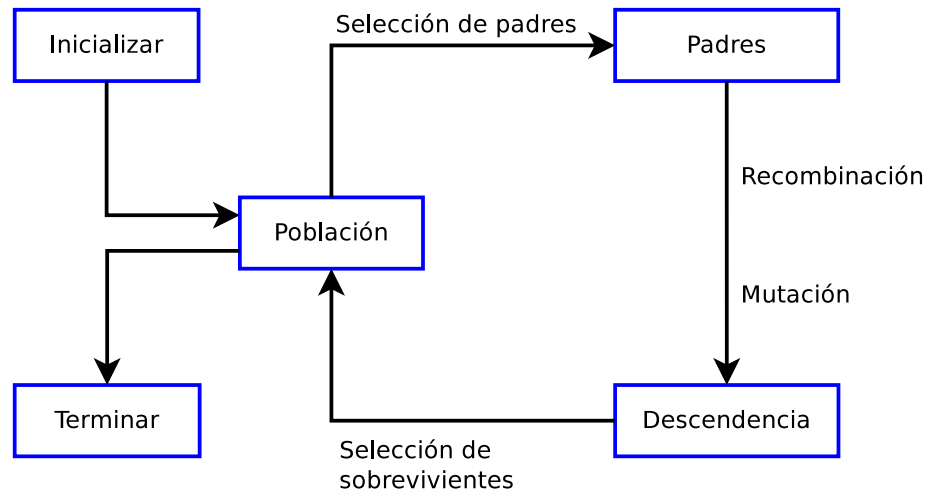


FIGURA 2.4: Esquema general de un algoritmo evolutivo (Eiben y Smith, 2015)

de la población y de este modo facilitan la existencia de innovación; y *b*) la selección actúa como una fuerza que incrementa la calidad promedio de las soluciones en la población (Eiben y Smith, 2015).

Los algoritmos evolutivos tienen características que ayudan a clasificarlos como métodos de prueba y error: *a*) son basados en población, procesan una colección entera de soluciones candidatas simultáneamente; *b*) la mayoría de ellos usa recombinación, mezclando información de dos o más candidatos para crear uno nuevo; y *c*) son estocásticos (Eiben y Smith, 2015).

Hay un número de componentes, procedimientos u operadores que se deben especificar para definir un algoritmo evolutivo (EA por sus siglas en inglés de *Evolutionary Algorithm*). Los más importantes son: *a*) representación (definición de individuos); *b*) función de evaluación (o función de aptitud); *c*) población; *d*) mecanismo de selección de padres; *e*) operadores de variación, recombinación y mutación; y *f*) mecanismo de selección de sobrevivientes (reemplazo).

Para crear un algoritmo capaz de ejecutarse exitosamente, es necesario especificar cada componente y definir el procedimiento de inicialización. Si se desea que el algoritmo se detenga en cierta fase, se debe proveer de un criterio de término.

Se dice que los objetos que forman posibles soluciones en el contexto del problema original se llaman fenotipos, mientras que su codificación como un individuo de un

algoritmo evolutivo se llama genotipo. Este primer paso de diseño se llama representación, que consiste en hacer un mapeo de los fenotipos en un conjunto de genotipos que los representen. La literatura de computación evolutiva maneja diferentes sinónimos: a) en el lado del problema original los términos solución candidata, fenotipo e individuo se usan para denotar a las posibles soluciones, el espacio de todas las soluciones candidatas posibles comúnmente se llama espacio de fenotipo; b) en el lado del EA los términos genotipo, cromosoma e individuo denotan los puntos en el espacio donde la búsqueda del algoritmo toma lugar, al que comúnmente se le llama espacio de genotipo; c) a los espacios o elementos de un individuo se les llama: variable, posición o gen. Un objeto que ocupa ese lugar se puede llamar valor o alelo (Eiben y Smith, 2015).

Algoritmos genéticos

Los algoritmos genéticos (GA por sus siglas en inglés de *Genetic Algorithm*) son el tipo de algoritmo evolutivo más extensamente conocido. Inicialmente fue concebido por Holland (1975), sin embargo son ampliamente considerados como métodos de optimización de funciones, en parte esto se debe al trabajo de Goldberg (1989) y al éxito de su aplicación en problemas de optimización.

Yang (2014) menciona que la esencia de un GA involucra la codificación de las soluciones de una función de optimización como arreglos de bits o cadenas de caracteres para representar cromosomas, la manipulación éstas cadenas de caracteres con operadores genéticos y su selección de acuerdo a su aptitud, con la finalidad de encontrar buenas soluciones (o incluso óptimas) al problema de interés.

Frecuentemente esto se hace mediante el siguiente procedimiento: 1) codificación de las funciones objetivo o de costo; 2) definición de una función de aptitud o un criterio de selección; 3) creación de una población de individuos; 4) llevar a cabo un ciclo de evolución o iteraciones que involucra la evaluación de la aptitud de todos los individuos en la población, la creación una nueva población realizando cruzamiento y mutación, y el reemplazo de la población vieja con la nueva; 5) decodificación de los resultados para obtener la solución al problema. Estos pasos se pueden representar como pseudocódigo de la manera mostrada en el algoritmo 1.

En GA una iteración para la creación de una nueva población se llama una *generación*; además la codificación de la función objetivo frecuentemente se hace en forma de

Algoritmo 1: Algoritmo Genético

```

Definir la función objetivo  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ 
Codificar las soluciones en cromosomas
Definir la función de aptitud  $F$ 
Generar población inicial
Inicializar probabilidad de cruzamiento ( $p_c$ ) y mutación ( $p_m$ )
Evaluar las soluciones y asignar aptitud a la población
mientras  $t < \text{Numero máximo de generaciones}$  hacer
    Generar nueva solución con cruzamiento y mutación
    Cruzar con la probabilidad de cruzamiento ( $p_c$ )
    Mutar con la probabilidad de mutación ( $p_m$ )
    Reemplazar la población existente con la nueva
    Actualizar  $t \leftarrow t + 1$ 
fin
Decodificar las soluciones

```

arreglos de bits (0, 1) o arreglos de números reales. Los operadores genéticos incluyen cruzamiento, mutación y selección de la población.

El cruce de dos padres es el operador principal y tiene una probabilidad alta p_c de aplicarse en cada generación. Se lleva a cabo mediante el intercambio de un segmento, escogido a una posición aleatoria, de un cromosoma con el segmento correspondiente de otro cromosoma. El cruzamiento que se lleva a cabo de esta manera se llama cruzamiento a un punto, sin embargo también se puede hacer cruzamiento en varios puntos, en los que se intercambian múltiples segmentos con los correspondientes de otro cromosoma. El cruzamiento a múltiples puntos se usa frecuentemente en algoritmos genéticos para incrementar la eficiencia evolutiva de los algoritmos.

La operación de mutación se logra cambiando genes seleccionados de manera aleatoria, la probabilidad de mutación p_m generalmente es pequeña. Además, la mutación puede ocurrir en múltiples sitios simultáneamente, lo cual puede tener ventajas en la práctica.

La elección de un individuo entre la población se lleva a cabo mediante la evaluación de su aptitud y puede permanecer en la nueva generación si alcanza un cierto umbral de aptitud. La selección también puede basarse en la aptitud de manera que la reproducción de una población sea proporcional a su aptitud, de esta manera los

individuos con mayor aptitud tienen mayor probabilidad de reproducirse.

Como se mencionó anteriormente, los algoritmos genéticos tienen tres operadores principales: cruzamiento, mutación y selección; y sus roles son muy diferentes.

- **Cruzamiento:** consiste en cambiar partes de una solución con otra y su principal rol es proveer de un mezclado de las soluciones y convergencia en un sub-espacio.
- **Mutación:** es el cambio de partes de una solución de manera aleatoria incrementa la diversidad de la población y proporciona un mecanismo para escapar de óptimos locales; y
- **Selección del más apto o elitismo:** se seleccionan las soluciones con aptitud más alta y se pasan a las siguientes generaciones, lo cual frecuentemente se lleva a cabo mediante alguna forma de selección de las mejores soluciones.

Los algoritmos genéticos han sido aplicados exitosamente a muchos problemas de optimización combinatoria, sin embargo éste éxito frecuentemente se ha alcanzado saliendo del esquema clásico descrito anteriormente. En particular, la codificación de las soluciones en cromosomas puede omitirse por completo (trabajando directamente en las soluciones) o diseñarse de manera específica para usar operadores de cruzamiento y mutación especializados (Gendreau y Potvin, 2005).

2.2.4. Algoritmos bio-inspirados

La naturaleza ha inspirado a muchos investigadores de diversas maneras, entre las que se encuentra el desarrollo de algoritmos. En la actualidad, la mayoría de los algoritmos nuevos son inspirados en la naturaleza, debido a que se han desarrollados de alguna manera usando inspiración de la naturaleza y por consiguiente, en el término más genérico, se puede hacer referencia a casi todos ellos como *algoritmos inspirados en la naturaleza*. Sin embargo, se pueden tener diferentes niveles de clasificación, dependiendo del detalle y las sub-fuentes de inspiración se deseen usar. Por simplicidad, se considera que en el nivel más alto de fuentes de inspiración están la biología, la física y la química. La mayor parte de estos algoritmos están basados en alguna característica de sistemas biológicos, por ello, la mayor parte de los algoritmos inspirados en

la naturaleza son algoritmos inspirados en biología, o abreviando: bio-inspirados. Entre los algoritmos bio-inspirados, una clase especial de algoritmos se ha desarrollado a partir de la inteligencia de enjambres. De hecho, los algoritmos basados en inteligencia de enjambres están entre los más populares, algunos ejemplos son la optimización con colonia de hormigas, optimización con enjambre de partículas, búsqueda cuckoo, algoritmo de murciélagos y algoritmo de luciérnagas (Fister et al., 2013).

Optimización con colonia de hormigas

Optimización con colonia de hormigas (ACO por sus siglas en inglés de *Ant Colony Optimization*) es una técnica discreta de optimización combinatoria desarrollada Marco Dorigo desde principios de la década de los 1990. Uno de los primeros algoritmos de esta familia es *Ant System* que se introdujo en el trabajo de Dorigo et al. (1996), una descripción completa de ACO se puede encontrar en Dorigo y Stützle (2004) o en el sitio web del autor (www.aco-metaheuristic.org) donde también se puede encontrar código computacional. La familia de algoritmos de ACO se inspiran en el comportamiento colectivo de una colonia de hormigas en búsqueda de alimento, que al paso del tiempo es capaz de encontrar la ruta más corta entre su nido y la fuente de alimento.

Uno de los problemas estudiados por los etólogos fue el entender, cómo ciertos animales casi ciegos pueden establecer la ruta más corta desde su colonia hasta las fuentes de comida e ir de regreso. Se encontró que el medio usado entre los individuos para comunicar información relacionada con estas rutas y usarla para decidir hacia dónde ir, consiste en rastros de feromona. Una hormiga en movimiento deja cierta feromona (en cantidades variables) en el terreno haciendo una ruta con el rastro de esta sustancia. Mientras que una hormiga sola se mueve prácticamente al azar, una hormiga que encuentra un rastro previo puede detectarlo y decidir seguirlo con una probabilidad alta, reforzando el rastro con su propia feromona. El comportamiento colectivo que emerge es una forma de comportamiento autocatalítico en el que entre más hormigas estén siguiendo un rastro, éste se vuelve más atractivo para que otras hormigas lo sigan. Entonces, este proceso se caracteriza por ser un ciclo de refuerzo positivo, donde la probabilidad con la cual una hormiga elige una ruta incrementa con el número de hormigas que escogieron la misma ruta (Dorigo et al., 1996).

ACO es una metaheurística en la que una colonia de hormigas artificiales es capaz de encontrar buenas soluciones a problemas difíciles de optimización discreta de manera cooperativa. La cooperación es el componente clave en el diseño de los algoritmos de ACO, esto se hace destinando recursos computacionales a un conjunto de agentes relativamente simples (hormigas artificiales) que se comunican de manera indirecta por medio del ambiente (en inglés a este principio se le conoce como *stigmergy*). La obtención de buenas soluciones es una propiedad emergente de la interacción cooperativa de dichos agentes (Dorigo y Stützle, 2004).

ACO ha sido utilizado exitosamente en la solución de problemas de optimización combinatoria NP duros como el problema del agente viajero o el problema de asignación cuadrática, comúnmente ACO se aplica a problemas de combinatoria de tipo: enrutamiento, asignación, programación de actividades y problemas de subconjuntos (Dorigo y Stützle, 2004).

Para aplicar ACO a un problema de optimización combinatoria se requiere representarlo como un grafo. Considérese el problema de minimización $(\mathcal{S}, f, \Omega)$, donde $\mathcal{S}$ es el *conjunto de soluciones candidatas*, f es la *función objetivo* que asigna un valor (costo) $f(s, t)$ a cada solución candidata $s \in \mathcal{S}$, y $\Omega(t)$ es un *conjunto de restricciones*. El parámetro t indica que la función objetivo y las restricciones pueden ser dependientes del tiempo, como en el caso de problemas dinámicos. El objetivo es encontrar una solución factible *óptima global* s^* , que es una solución factible de costo mínimo. Dada esta formulación, las hormigas artificiales construyen soluciones al realizar recorridos aleatorios en la grafo completo $G_C(C, L)$ cuyos nodos son los componentes C y el conjunto L conecta completamente los C componentes. El grafo G_C se llama *gráfica de construcción* y los elementos de L son *conexiones* (aristas) (Dorigo y Stützle, 2004).

El algoritmo de ACO opera mediante la generación iterativa de una población de soluciones donde cada solución es una representación de la ruta que cada hormiga ha seguido. Las soluciones se construyen de manera creciente conforme las hormigas se mueven de un punto de decisión a otro hasta que todos han sido cubiertos. Después de que la hormiga k ha creado una solución se calcula su costo usando la función objetivo. Cuando se han completado estos dos pasos se dice que se ha completado un ciclo k .

En cada punto de decisión (que son los nodos o vértices del grafo) una hormiga

debe seleccionar una conexión (un componente) basado en una regla de decisión probabilística, llamada *regla proporcional aleatoria*. La probabilidad p de que una hormiga k situada en el nodo i elija moverse hacia el nodo j es:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{si } j \in \mathcal{N}_i^k, \quad (2.16)$$

donde τ_{ij} es la concentración de feromona asociado al arista (i, j) , $\eta_{ij} = 1/d_{ij}$ es un valor heurístico disponible a priori, d_{ij} es la distancia que existe entre i y j , α y β son dos parámetros que determinan la influencia relativa del rastro de feromona y la información heurística, y $\mathcal{N}_i^k$ es el vecindario factible para la hormiga k cuando se encuentra en i , esto es el conjunto de nodos no visitados por k (la probabilidad de escoger un nodo que se encuentra fuera de $\mathcal{N}_i^k$ es 0). El valor heurístico η_{ij} es análogo a proveer a las hormigas con visión por lo que algunas veces es llamado “visibilidad”.

Una iteración consiste de m ciclos, donde m corresponde al número de hormigas artificiales usadas. Al final de cada iteración cuando todas las hormigas han generado una solución, se actualiza la concentración de feromona en cada arista. Para hacer esto primero se disminuye el valor de la feromona en todas las aristas por un factor constante, y después se agrega feromona a los aristas que las hormigas usaron al crear las soluciones. La evaporación de feromona se hace mediante la ecuación:

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \quad \forall (i, j) \in L, \quad (2.17)$$

donde $0 < \rho \leq 1$ es la tasa de evaporación de feromona. El parámetro ρ se usa para evitar la acumulación ilimitada de feromona en las aristas y permite al algoritmo “olvidar” las malas decisiones tomadas previamente. De hecho, cuando una arista no es seleccionada por las hormigas, el valor de su feromona asociada decrece de manera exponencial con el número de iteraciones. Después de la evaporación, todas las hormigas depositan feromona en las aristas que seleccionaron al crear su solución:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad \forall (i, j) \in L, \quad (2.18)$$

donde m es el número de hormigas de la colonia, $\Delta\tau_{ij}^k$ es la cantidad de feromona que una hormiga k deposita en los aristas que ha seleccionado. Se define como:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{f(S^k)} & \text{si el arista } (i, j) \text{ pertenece a la solución de la hormiga } k. \\ 0 & \text{de otra manera.} \end{cases} \quad (2.19)$$

donde $f(S^k)$ es el costo producido por la solución de la hormiga k que es la suma de los costos en cada una de las aristas en la solución. En la ecuación 2.19 entre mejor sea la solución generada por la hormiga, más feromona recibirán las aristas de la solución. En general las aristas que son usadas por muchas hormigas y que son parte de soluciones menos costosas, reciben más feromona y es más probable que sean elegidas por las hormigas en futuras iteraciones del algoritmo.

Los pasos de la metodología descrita hasta ahora son para el algoritmo de ACO llamado *Ant System*. Un inconveniente de este algoritmo es que su rendimiento disminuye drásticamente conforme se incrementa el tamaño de la instancia a la que se aplica. Por eso mucha de la investigación desarrollada sobre ACO se enfoca en mejorar el algoritmo AS. Una compilación de los algoritmos de ACO que se han desarrollado se presenta en el cuadro 2.1. El procedimiento se muestra en forma de pseudocódigo en el algoritmo 2.

2.3. Trazo y dimensionamiento de redes de distribución de agua

Los sistemas de distribución de agua generalmente incluyen a las redes que tienen la función de conducir el agua desde una fuente de abastecimiento hasta los consumidores. Estas redes pueden ser de canales o tuberías a presión. Las redes de tuberías se pueden clasificar en redes abiertas (también conocidas como ramales o redes con topología de árbol) y redes cerradas, de acuerdo a su topología. Las redes abiertas se caracterizan por no tener ningún “circuito cerrado” en el sistema. Ejemplos de este tipo de redes son: sistemas de tuberías que unen una batería de pozos de agua con un tanque de abastecimiento o una planta de tratamiento, la red contra incendios en el interior de una edificación o las tuberías de un sistema de riego localizado. La característica primordial de las redes cerradas es tener algún tipo de “circuito cerrado” (*loop*

Algoritmo 2: Optimización con Colonia de Hormigas

```

Leer instancia del problema
Calcular matriz de distancias
Calcular lista de vecinos más cercanos
Calcular matriz con información para selección
Inicializar las  $m$  hormigas
Inicializar parámetros (rastro de feromona  $\tau$ )
Inicializar estadísticas
mientras no se cumpla la condición de término hacer
    Construir soluciones
    Búsqueda local (opcional)
    Actualizar estadísticas
    Evaporar feromona ( $\rho$ )
    para  $k \leftarrow 0$  a  $k \leq m$  hacer
        Depositar feromona de hormiga  $k$ 
    fin
    Recalcular matriz con información para selección
    Actualizar iteración  $i \leftarrow i + 1$ 
fin
Obtener la solución de la mejor hormiga  $k^*$ 

```

en inglés) en el sistema. El objetivo de esto es tener un sistema redundante de tuberías donde cualquier zona dentro del área cubierta por el sistema pueda ser alcanzada simultáneamente por más de una tubería, aumentando así la confiabilidad del abastecimiento. Este tipo de red se usa con frecuencia en sistemas de distribución de agua potable en una ciudad (Saldarriaga, 2007). En la distribución de agua con finalidades agrícolas se usan solo redes abiertas, ya que la implementación de ciclos dentro de la red, resulta costosa e innecesaria, pues el abastecimiento no es tan demandante y riguroso en el caso de uso urbano. En la figura 2.5 se muestra un esquema de una red de riego con las características descritas.

Ángeles Montiel (2002) menciona que los elementos fundamentales que componen una red de distribución de agua para riego son los siguientes:

1. Una obra de captación de aguas superficiales o subterráneas, que puede ser un río, embalse, canal, manantial, pozo a cielo abierto o profundo, con una instalación para el filtrado del agua en caso de ser necesario;

CUADRO 2.1: Algoritmos de ACO de acuerdo a su aparición cronológica

Algoritmo de ACO	Principales referencias
Ant System (AS)	Dorigo (1992); Dorigo et al. (1996)
Elitist AS	Dorigo (1992); Dorigo et al. (1996)
Ant-Q	Gambardella & Dorigo (1995); Dorigo & Gambardella (1996)
Ant Colony System	Dorigo & Gambardella (1997a,b)
$MAX - MIN$ AS	Stützle y Hoos (2000); Stützle (1999)
Ranked-based AS	Bullnheimer, Hartl & Strauss (1997, 1999c)
ANTS	Maniezzo (1999)
Hyper-cube AS	Blum, Roli & Dorigo (2001); Blum & Dorigo (2004)

2. Un dispositivo para impulsar el agua a presión que normalmente es un equipo de bombeo, un tanque elevado, etc;
3. Un dispositivo de regulación que puede ser un depósito, un embalse o una torre de carga, colocado entre la red que consume un caudal variable y las bombas que proporcionan un caudal constante;
4. Una red de tuberías, que transporta el agua hasta los hidrantes y que generalmente va enterrada. Los tipos de tubería utilizados, en general, son: el fibrocemento, el acero, el hormigón (armado o pretensado), plásticos, etc;
5. Tomas de riego o hidrantes, situados a la entrada de las parcelas dentro de una estructura de protección y operación denominada registro. Que pueden ser tomas individuales (una por parcela) o colectivas (una para varias parcelas). Para un control adecuado del agua de riego, los hidrantes pueden equiparse con reguladores de gasto o de presión, limitadores de gasto y medidores del agua; e
6. Instalaciones específicas correspondientes al método de riego a emplear, por ejemplo en el caso de riego a presión serían:
 - a) Líneas laterales: que pueden ser sistemas fijos o móviles y estos últimos pueden ser a la vez manuales o mecanizados.
 - b) Emisores: Pueden ser tuberías perforadas, tuberías con compuertas, aspersores, goteros, microaspersores o borboteadores.

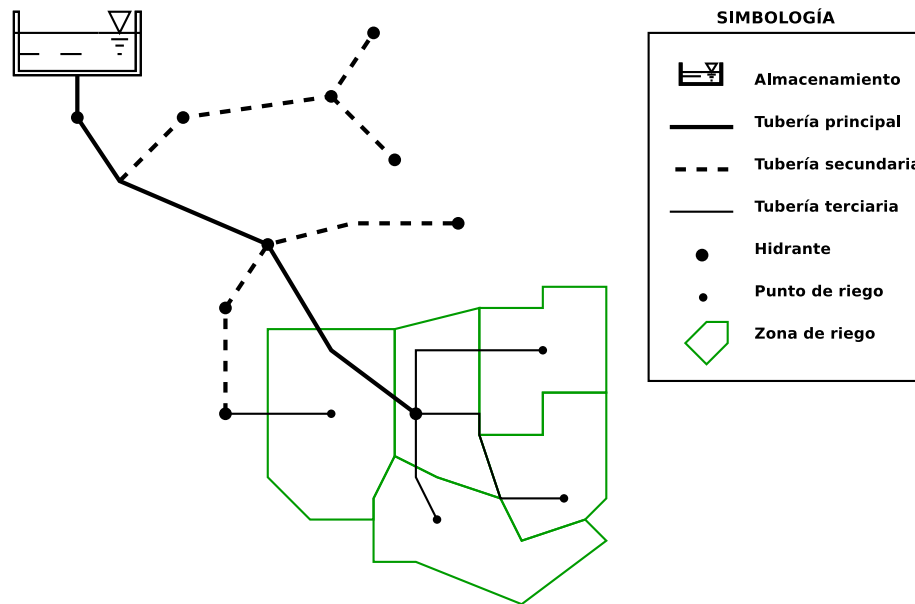


FIGURA 2.5: Esquema general de una red abierta de tuberías para riego

Eiger et al. (1994) mencionan que el proceso de planeación de una red de distribución de agua consiste de tres fases: *a)* trazo, *b)* dimensionamiento, y *c)* operación; y a pesar de que cada fase es dependiente de las otras, éstas se pueden formular y resolver como problemas independientes. Entonces el procedimiento de diseño de redes se lleva a cabo mediante iteraciones de estas tres fases.

El diseño de una red consiste en hacer una propuesta de su trazo y dimensiones (diámetros), y posteriormente hacer un análisis para determinar si la propuesta es aceptable. Una red aceptable es aquella que cumple las tres leyes básicas de mecánica de fluidos: conservación de masa, primera ley de la termodinámica y la segunda ley de Newton (conservación de *momentum*).

El trazo de una red de riego se define como sigue: dado un conjunto de puntos de consumo, un conjunto de puntos de abastecimiento y el conjunto de todas rutas potenciales de tuberías entre ellos, el problema de trazo de una red de distribución de agua consiste en la selección de las rutas que conecten cada punto de consumo con uno de abastecimiento. El objetivo de la optimización es encontrar la red más económica que cumpla con lo anterior, además de un conjunto de restricciones hidráulicas.

Gonçalves et al. (2014) mencionan que este problema es una instancia del problema de *Árbol de Steiner*, por lo tanto se puede decir que el trazo de una red de distribución de agua es un problema NP-hard dado que es una generalización de un problema NP-hard bien conocido.

Es común hacer el trazo por los linderos o usando el método de Girette, mientras que para el dimensionamiento se puede emplear alguna técnica de optimización como multiplicadores de Lagrange, programación dinámica, programación lineal o el método discontinuo de Laybe et al. (1988); para minimizar el costo total de la red, sujeto a un conjunto de restricciones hidráulicas (Ángeles Montiel, 2002).

El problema de dimensionamiento consiste en la selección óptima de diámetros de tuberías para constituir una red de distribución de agua con respecto a ciertos requerimientos de presión.

Yates et al. (1984) demostraron que el dimensionamiento de redes es un problema NP-hard, principalmente por dos razones: la existencia de ecuaciones no lineales y el uso de un conjunto de valores discretos para los diámetros de tubería (Zhou et al., 2015).

Una vez que se tiene realizado el trazo y se han seleccionado los diámetros de las tuberías de la red se debe realizar un análisis para comprobar que se cumplen las condiciones hidráulicas de operación.

El comportamiento de una red de distribución de agua está modelado matemáticamente por el siguiente sistema de ecuaciones algebraicas no lineales. La ecuación 2.20 es la ecuación de continuidad que se deriva de la ley de conservación de la masa y establece que en cada nodo de la red se debe cumplir la continuidad del flujo, por lo que se plantea una ecuación por cada nodo de la red. La ecuación 2.21 es una generalización de las ecuaciones utilizadas para calcular las pérdidas de carga en un tramo de tubería (Darcy-Weisbach o Hazen-Williams), por lo que se debe plantear una para cada tramo de tubería de la red. Las ecuaciones de conservación de energía son no lineales y se aplican en circuitos cerrados simples o entre dos nodos con cargas de presión fija conocidas en trayectorias independientes de una red. Sea cual sea el caso, se debe cumplir la condición que la diferencia entre el total de las pérdidas de carga y la línea de gradiente debe ser cero, lo que se expresa con la ecuación 2.22.

$$\sum Q_e - \sum Q_s - DM = 0 \quad (2.20)$$

$$H_i - H_k = r_j Q_j |Q_j|^{\alpha-1} = \omega \frac{L_j}{C^\alpha D_j^\beta} Q_j |Q_j|^{\alpha-1} \quad (2.21)$$

$$\sum_{j \in L_s} \omega \frac{L_j}{C^\alpha D_j^\beta} Q_j |Q_j|^{\alpha-1} - \sum_{j \in L_s} HGL_j = 0 \quad (2.22)$$

donde Q_e es el caudal o gasto que entra al nodo, Q_s es el caudal que sale del nodo y DM es la demanda en el nodo, H_i y H_k son las cargas de presión en los nodos i y k respectivamente, r_j es el factor de resistencia para la tubería j , Q_j es el flujo en la tubería j , ω es una constante de conversión numérica que depende de las unidades usadas, L_j es la longitud de la tubería j , C es el coeficiente de rugosidad del material de la tubería, α y β son coeficientes, L_s son los índices de las tuberías en un circuito cerrado simple o una trayectoria abierta, HGL_j es la línea piezométrica en la fuente de abastecimiento j , $H_{\min}$ es la carga de presión mínima requerida (que se supone es conocida), H_i es la carga de presión obtenida del análisis hidráulico y N es el número de nodos de la red. Los coeficientes ω , α y β toman valores de acuerdo a la ecuación de pérdidas de carga utilizada.

El análisis hidráulico de la red se puede realizar con hojas de cálculo y programas de cálculo como Mathcad o MATLAB®. Sin embargo, esto es útil únicamente en el caso de redes pequeñas, ya que para redes más grandes la tarea de programación se vuelve engorrosa, sobretodo porque cada sistema analizado requiere de crear una descripción única y un sistema de ecuaciones muy grande a ser resuelto (Potter et al., 2011).

Este sistema de ecuaciones comúnmente se soluciona utilizando métodos numéricos como el de Hardy Cross o el método de gradiente propuesto por Todini y Pilati (1987). Éste último método se encuentra codificado en el programa de cómputo EPANET 2, lo que facilita el trabajo de hacer el análisis de la red.

EPANET 2 es un programa desarrollado por la Agencia de Protección Ambiental de los EE.UU. (EPA, por sus siglas en inglés de *United States Environmental Protection Agency*). Es un programa muy completo que simula tanto flujo hidráulico como calidad de agua en redes de tuberías presurizadas. Para el análisis hidráulico de redes de tuberías, utiliza un algoritmo híbrido de circuito y nodo denominado *método de gradiente* para determinar las cargas piezométricas y caudales desconocidos. La red puede contener tuberías, uniones, bombas de velocidad fija o variable, varios tipos de válvulas,

depósitos y tanques de almacenamiento de agua con altura piezométrica constante o variable y sistemas de control y operación basados en tiempo, nivel o presión. La pérdida de carga por fricción se calcula con las fórmulas de Darcy-Weisbach, Hazen-Williams o Chezy-Manning. Además es capaz de aceptar datos de la bomba o de potencia útil (Potter et al. (2011); Saldarriaga (2007)).

En el diseño de redes se deben respetar los límites de velocidades permisibles, según Richter (1962) para redes de distribución para agua potable e industrial va desde una velocidad mínima de 0.5 m/s hasta una máxima de 3 m/s . En México, la norma NMX-O-177-SCFI-2011 (2011) contiene una tabla de velocidades recomendadas para el diseño hidráulico de que van desde una mínima de 1 m/s hasta una máxima de 2.5 m/s dependiendo del diámetro de la tubería, los valores específicos se muestran en el cuadro 2.2.

CUADRO 2.2: Velocidades máximas permisibles recomendadas para el diseño hidráulico.

Diámetro interno del tubo [m]	Velocidad máxima [m/s]
0.101	2.0
0.127	2.0
0.160	2.0
0.200	2.0
0.254	2.1
0.305	2.1
0.356	2.2
0.400	2.2
0.457	2.3
0.500	2.4
0.610	2.5
0.700	2.5

2.4. Optimización de sistemas de distribución de agua

El análisis de sistemas de recursos hídricos es la ciencia de desarrollar y aplicar metodologías matemáticas de investigación de operaciones a problemas de manejo de recursos hídricos compuestos de obras de almacenamiento, ríos, cuencas, agua subterránea, sistemas de distribución, entre otros, como sistemas independientes o integrados, para sistemas mono-objetivo o multiobjetivo, determinísticos o estocásticos (Ostfeld, 2011).

Según Ostfeld (2011) durante las pasadas cuatro décadas se han propuesto numerosos modelos para el análisis de sistemas de recursos hídricos. Una posible clasificación es:

1. Métodos basados en descomposición, en el cual se soluciona un problema de punto “interior” lineal o cuadrático para un conjunto de variables de decisión, mientras que ese conjunto se altera por un problema de punto “exterior” usando una técnica de gradiente o subgradiente;
2. Métodos que utilizan una formulación sencilla de programación no lineal;
3. Métodos que se basan en la interacción conjunta de un programa de simulación con un código de optimización general no lineal; y
4. Métodos que emplean técnicas evolutivas, como algoritmos genéticos.

Debido a su naturaleza, este tipo de problemas también se pueden resolver con metaheurísticas de optimización discreta. Zheng et al. (2013) hacen una recopilación de las principales técnicas que se han utilizado para resolver problemas de optimización de sistemas de distribución de agua. Entre éstos, Bi et al. (2015) menciona que los GAs destacan por ser un método de optimización evolutiva basado en el concepto de sobrevivencia del más fuerte que ha sido usado extensivamente para el diseño óptimo y la operación de sistemas de distribución de agua, sin embargo su aplicación ha sido principalmente en problemas de prueba relativamente sencillos. Maier et al. (2001) y Maier et al. (2003) señalan que los algoritmos de ACO han resultado ser una alternativa atractiva para el diseño óptimo de sistemas de distribución de agua ya que encuentra

buenas soluciones usando una menor cantidad de evaluaciones de la función objetivo que GA.

Aunque el ahorro debido a la optimización del trazo de las redes es generalmente mayor al que se obtiene por el dimensionamiento de sus componentes, la mayor parte de la investigación se ha enfocado en éste último, que es el problema más fácil (Afshar y Mariño, 2006). Algunas redes de prueba clásicas usadas frecuentemente en la literatura para optimización de los diámetros de tuberías, son: 1) la red de dos circuitos (Alperovits y Shamir, 1977); 2) el problema de 14 tuberías (Simpson et al., 1994); 3) el sistema de túneles de New York (Dandy et al., 1996); 4) la red Hanoi; y 5) la red Hanoi extendida (Bi et al., 2015). Como ejemplo se muestra la configuración la red de dos circuitos en la figura 2.6 y la red de Hanoi en la figura 2.7.

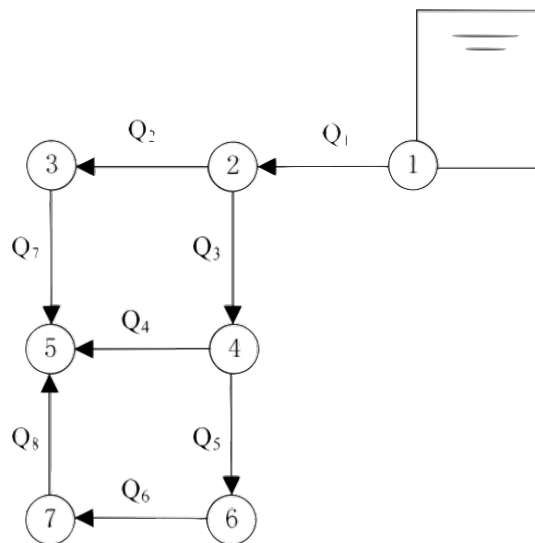


FIGURA 2.6: La red de dos circuitos, una red de prueba usada frecuentemente en la literatura.

ACO ha sido utilizado ampliamente en el manejo de recursos hídricos, una revisión muy completa de sus aplicaciones se encuentra en Afshar et al. (2015), donde se pueden encontrar áreas de aplicación como *a)* operación de presas y manejo de aguas superficiales, *b)* sistemas de distribución de agua, *c)* ingeniería de aguas residuales y drenaje, *d)* sistemas de agua subterránea incluyendo saneamiento, monitoreo y manejo, *e)* problemas de manejo de cuencas y ambientales, *f)* entre otras aplicaciones.

2.5. Optimización multiobjetivo de sistemas de distribución de agua

Durante las pasadas dos décadas una tendencia ha sido plantear y resolver el análisis de redes como un problema de optimización multiobjetivo. Formiga et al. (2003) realizaron el diseño óptimo de redes usando como objetivos el costo de la red, la entropía del sistema y la capacidad de suministro de la red. Por otro lado, Keedwell y Khu (2006) usaron un enfoque de autómatas celulares para optimización del costo y déficit de carga de redes del mundo real, obteniendo mejores resultados que NSGA-II. Farmani et al. (2005) realizaron un estudio comparativo de diferentes métodos de optimización evolutiva multi-objetivo (NSGA-II y SPEA2) que son comúnmente aplicados al diseño de sistemas de distribución de agua, siendo los objetivos el costo y el déficit de carga de la red.

El trazo y dimensionamiento simultáneo de redes ramales como las que se usan para riego, ha sido resuelto como un problema multiobjetivo por Afshar (2006) usando el algoritmo *Max-Min Ant System* de ACO. Siguiendo el mismo enfoque Afshar y Jabbari (2008) solucionaron problemas similares usando GA, mientras que Saleh y Tanyimboh (2013) utilizaron NSGA-II, por su parte Mtolera et al. (2014) utilizaron PSO.

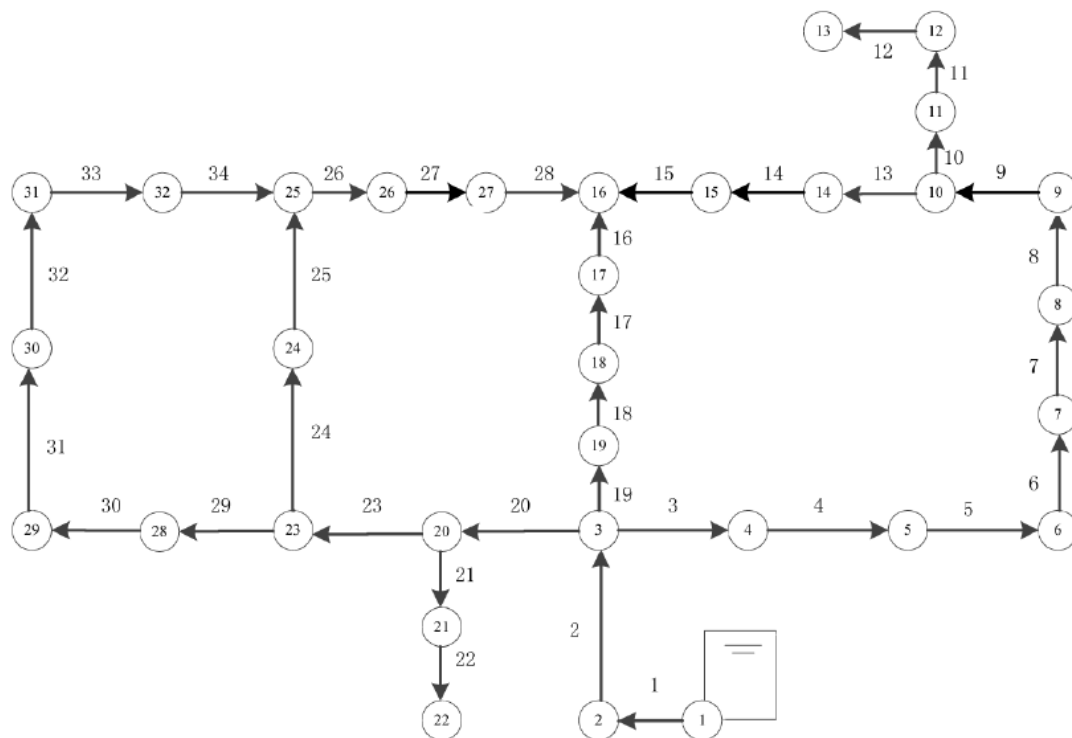


FIGURA 2.7: Configuración de la red de Hanoi, una red de prueba usada frecuentemente en la literatura.

Capítulo 3

Materiales y métodos

Para la realización de esta investigación se propuso el esquema de trabajo mostrado en la figura 3.1, donde se puede observar que se plantearon dos enfoques diferentes. En el primer enfoque se propuso plantear problemas de optimización de un objetivo para resolver por separado y de manera secuencial el trazo y el dimensionamiento de redes de riego. En el segundo enfoque se propuso plantear un problema de optimización multiobjetivo en el que tanto el trazo como el dimensionamiento de la red se resuelven de manera simultánea.

En las siguientes secciones se describe la forma en la que se plantearon los problemas para ambos enfoques así como las estrategias seleccionadas para resolverlos. Primeramente se presenta la formulación de los problemas de un objetivo: a) el trazo y b) el dimensionamiento de la red.

Para hacer la formulación del diseño de redes de riego como problemas de optimización matemática, se consideraron los siguientes supuestos (Saleh y Tanyimboh, 2013): *a)* la configuración de la red es conocida, incluyendo todas las conexiones factibles; *b)* las variables de decisión son las conexiones entre los nodos (tuberías) y los diámetros que éstas deben tener, el objetivo es encontrar y dimensionar el conjunto subóptimo de tuberías; *c)* las demandas de agua están localizadas en los nodos, aunque en la realidad las demandas pueden ocurrir a lo largo de las tuberías, sin embargo con fines de facilitar el análisis éstas se acumulan y se colocan en los nodos; *d)* las demandas en los nodos son conocidas con certidumbre, aunque en la práctica las demandas pueden ser inciertas, no es el objetivo estudiar éste y otros aspectos que causan incertidumbre en sistemas de distribución de agua; *e)* la carga de presión requerida en cada nodo con demanda es conocida; y finalmente, *f)* el diseño se optimizará considerando

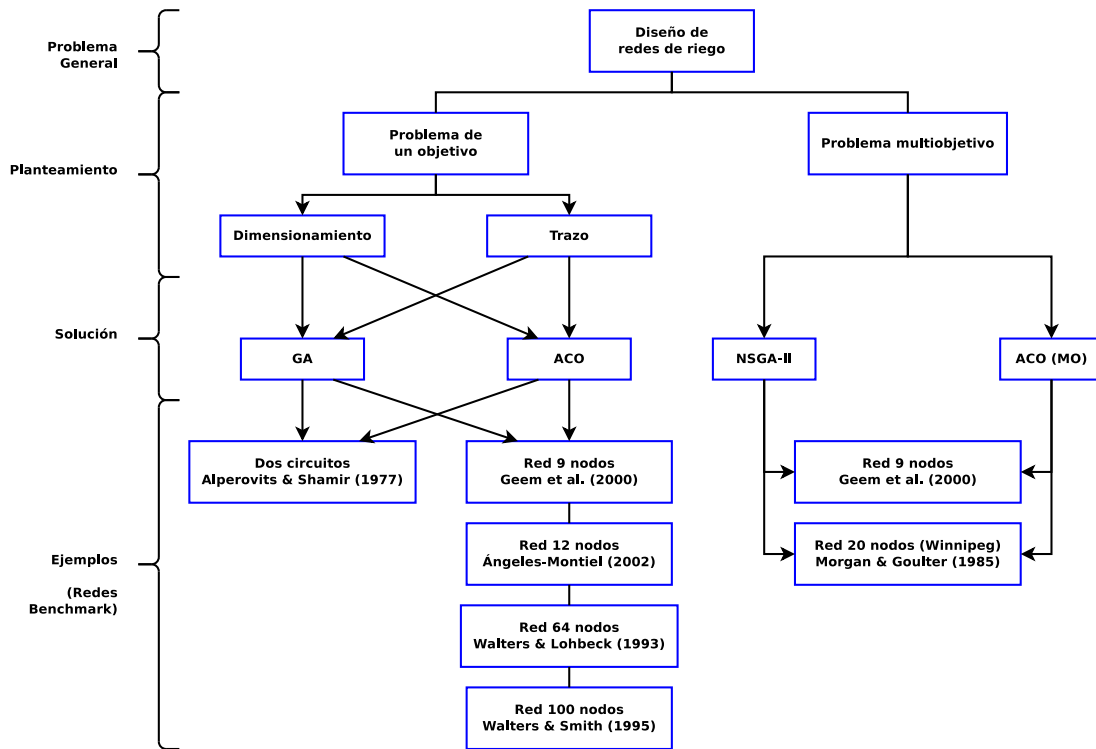


FIGURA 3.1: Procedimiento de trabajo

únicamente los costos de construcción iniciales.

Una red de tuberías se puede representar mediante un esquema gráfico con nodos y líneas. Usando esta representación el problema de diseño de redes se enuncia como sigue:

Dado un conjunto de nodos, el diseño consiste en encontrar la combinación de tuberías y sus diámetros, que debe tener una red para que su costo sea mínimo.

En este caso se entiende por una tubería como una línea que une dos nodos, entonces al unir todos los nodos se tiene una red con una topología particular. Además cada tramo de tubería debe tener un diámetro asignado con la finalidad de suministrar el agua requerida en las tomas (en este caso representadas por los nodos). Usando este enunciado se puede notar que las variables de decisión a considerar son: *a)* la topología de la red; y *b)* el diámetro de cada tramo de tubería. Matemáticamente este es un problema de optimización combinatoria de variables discretas con restricciones.

3.1. Problema de trazo

3.1.1. Variables de decisión

En el problema de trazo de redes abiertas de tuberías, se considera como principal variable de decisión a la topología de la red. Para ello se debe seleccionar la combinación de conexiones entre los nodos que formen la red de menor costo. La configuración de la red se puede representar con un vector de números enteros que indican la forma en la que un conjunto de nodos se conectan entre si, en teoría de grafos a esto se le conoce como una lista de adyacencia:

$$\mathbf{T} = [n_1, n_j, \dots, n_N] \quad \forall j \in \{1, \dots, N\} \quad (3.1)$$

donde $\mathbf{T}$ es una lista de adyacencia que representa la topología de la red, N es el número de nodos de la red. Donde el valor de n_j representa el nodo inicial de la tubería y la posición j representa el nodo final de la tubería. Esta representación solo es apropiada para redes abiertas.

3.1.2. Función objetivo

Dado que se está considerando que posteriormente se realizará el dimensionamiento de la red, la función objetivo para el trazo se puede plantear con la finalidad de minimizar la longitud asociada a la geometría de las conexiones entre los nodos, como en la ecuación 3.2:

$$\min f_1 = \sum_{i=1}^P L_i \cdot \left(\sum_{j=1}^N pN_j + \sum_{j=1}^N pS_j \right) \quad (3.2)$$

donde f_1 representa la longitud total de la red, L_i es la longitud de la tubería i , P es el número de tuberías de la red, pN_j es el valor de la penalización si el nodo j está desconectado, pS_j es la penalización por falta de suministro de agua en el nodo j , las demás variables se definieron anteriormente. En el caso de que se cumplan todas las restricciones el término entre paréntesis se elimina de la función objetivo, éste término únicamente aparece cuando hay incumplimiento de alguna restricción.

La forma de aplicar las penalizaciones al modelo se conoce como método de penalizaciones externas y tiene la ventaja de no requerir de soluciones factibles para operar, por lo que se realiza la búsqueda de una solución óptima en toda la extensión del espacio de búsqueda. Este mismo enfoque se utilizó para el planteamiento de la función objetivo del problema de dimensionamiento presentada más adelante.

Si se considera que el costo total de la red está asociado únicamente a la topología, este problema se puede plantear de diferente manera por lo que se tendría que usar una función objetivo distinta (Geem et al. (2000); Afshar y Mariño (2006)).

3.1.3. Restricciones

Debido a la naturaleza estocástica de los métodos heurísticos es necesario considerar restricciones que faciliten la tarea de descartar las soluciones no factibles y escoger únicamente las soluciones factibles.

Para este problema se consideran dos restricciones relacionadas con la topología de la red. La primera restricción de topología indica que no deben quedar nodos desconectados, esto sucede cuando un nodo se conecta consigo mismo, es decir cuando en la lista de adyacencia el valor de $n_j = j$. En el caso de que esto pase, se debe aplicar una penalización de acuerdo a la función:

$$pN_j = \begin{cases} \geq 1 & \text{si } n_j = j \quad \forall \quad j \in \{1, \dots, N\} \\ 0 & \text{de otra manera} \end{cases} \quad (3.3)$$

donde pN_j es el valor de penalización para el nodo j , n_j es el nodo inicial de la tubería, j es del nodo final de la tubería y N es el número de nodos de la red.

La segunda restricción indica que todos los nodos deben estar conectados dentro de una sola red con topología de árbol, por lo que se deben evitar redes que presenten circuitos. Para asegurar esto se creó un vector llamado “orden de suministro” el cual es un índice asignado a cada nodo de la red, que indica que tan lejos se encuentra de la fuente de abastecimiento. El orden 0 se asigna a las fuentes de abastecimiento, los nodos de orden 1 están conectados directamente a la fuente de abastecimiento, los nodos de orden 2 se conectan a nodos de orden 1, los nodos de orden 3 a su vez se conectan a nodos de orden 2, etc. De esta manera tan simple se asegura que los nodos se conecten

en una estructura de árbol. Se aplica penalización por suministro cuando se tienen más fuentes de abastecimiento de las deseadas, por ejemplo si se tiene solo una fuente de abastecimiento se penalizará cuando haya más de un cero en el vector.

$$pS_j = \begin{cases} \geq 1 & \text{si } s_j = 0 \quad \forall \quad j \in \{1, \dots, N\} \\ 0 & \text{de otra manera} \end{cases} \quad (3.4)$$

donde pS_j es la penalización por falta de suministro para el nodo j , s_j es el valor del orden de suministro para el nodo j y N es el número de nodos de la red.

La forma en la que se hizo el planteamiento asegura que las redes que tienen el número de fuentes de abastecimiento deseadas serán redes ramales. En el caso de que se tenga algún circuito en la red se incumplirá esta restricción, por lo que se descartarán en automático las soluciones candidatas que presenten circuitos.

3.2. Problema de dimensionamiento

3.2.1. Variables de decisión

El problema de dimensionamiento considera como variable de decisión el diámetro de cada tubería de la red, esto se representa como un vector de números enteros en el que existe un elemento por cada tramo de tubería y su valor representa el diámetro de dicha tubería. Los diámetros se elijen de un conjunto finito de diámetros comerciales previamente seleccionado:

$$\mathbf{D} = [\{d_1, d_i, \dots, d_P\} \quad \forall d_i \in \Omega] \quad (3.5)$$

donde el vector $\mathbf{D}$ representa el vector de diámetros de las tuberías de la red, P es el número de tuberías de la red, d_i es el diámetro seleccionado para la tubería i , que a su vez debe estar dentro del conjunto de diámetros comerciales Ω permitidos.

3.2.2. Función objetivo

El objetivo de este problema es encontrar la red de menor costo asociada a los diámetros de las tuberías. Para ello se proponen los diámetros de una red con trazo conocido previamente y se obtienen sus costos. La función objetivo para el problema de dimensionamiento se representa con la ecuación 3.6:

$$\min f_2 = \sum_{i=1}^P L_i \cdot c(d_i) \cdot \left(\sum_{j=1}^N pH_j + \sum_{i=1}^P pV_i \right); \quad \forall d_i \in \Omega \quad (3.6)$$

Sujeto a:

El sistema de ecuaciones formado por (2.20), (2.21) y (2.22).

donde f_2 es el costo total de la red, L_i es la longitud de la tubería i , d_i es el diámetro de la tubería i , pH_j es la penalización por déficit de carga de presión en el nodo j , pV_i es la penalización por incumplimiento de velocidad de flujo permisible en la tubería i , Ω es un conjunto de diámetros comerciales de tuberías, $c(d_i)$ es una función para obtener el costo por unidad de longitud para cada diámetro de tubo comercial, las demás variables se definieron previamente. En el caso de que se cumplan todas las restricciones el término entre paréntesis se elimina de la función objetivo, éste término únicamente aparece cuando hay incumplimiento de alguna restricción.

3.2.3. Restricciones

Las restricciones hidráulicas se manejan con el sistema de ecuaciones obtenido al aplicar las ecuaciones 2.20, 2.21 y 2.22 sobre la red que se esté trabajando.

Por otro lado, la carga de presión mínima expresada con la ecuación 3.7, asegura que existe suficiente presión en los nodos para cubrir las demandas de agua.

$$H_j \geq H_{j\min}, \quad \forall j \in \{1, \dots, N\} \quad (3.7)$$

donde H_j es la presión obtenida para el nodo j , $H_{j\min}$ es la presión mínima requerida en el nodo j . En este caso se penaliza cuando la presión en los nodos obtenida del análisis hidráulico es menor a la presión requerida en cada nodo:

$$pH_j = \begin{cases} \geq 1 & \text{si } H_j < H_{j\min} \quad \forall \quad j \in \{1, \dots, N\} \\ 0 & \text{de otra manera} \end{cases} \quad (3.8)$$

donde pH_j es la penalización por incumplimiento de presión en el nodo j , las demás variables se definieron previamente.

Además en cada tubería se verifica que la velocidad del flujo se encuentre dentro del rango de velocidades permisibles $[v_{\min}, v_{\max}]$ de acuerdo a la expresión 3.9:

$$v_{\min} < v_i < v_{\max} \quad \forall \quad i \in \{1, \dots, P\} \quad (3.9)$$

donde v_i es la velocidad del flujo obtenida del análisis hidráulico para la tubería i , P es el número de tuberías de la red y $v_{\min}$, $v_{\max}$ son las velocidades de flujo mínima y máxima permisibles respectivamente. Usando la ecuación 3.9 se plantea la función para penalización por incumplimiento de velocidad de flujo como se muestra en la ecuación 3.10:

$$pV_i = \begin{cases} \geq 1 & \text{si } v_i > v_{\max}; \text{ o bien si } v_{\min} < v_i \quad \forall \quad i \in \{1, \dots, P\} \\ 0 & \text{de otra manera} \end{cases} \quad (3.10)$$

donde pV es la penalización por incumplimiento de velocidad en la tubería i .

El problema de dimensionamiento requiere que la red propuesta cumpla con las condiciones del sistema de ecuaciones compuesto por las expresiones 2.20, 2.21 y 2.22, que modelan el comportamiento del caudal, la presión y velocidad del agua dentro de las tuberías de la red. En este trabajo, se considera que una red es hidráulicamente aceptable si al hacer un análisis, se obtiene que dicha red cumple estas condiciones.

El análisis hidráulico se realizó mediante el software de simulación EPANET 2 que cuenta con algoritmos robustos para analizar cualquier tipo de red, específicamente se utilizó la librería de funciones de *EPANET 2 Toolkit*, cuyo código es de dominio público y se encuentra en lenguaje ANSI C. Para usar esta librería en Python se utilizó el módulo *epanet-python* de Open Water Analytics.

3.3. Problema multiobjetivo

Para el enfoque multiobjetivo se planteó un problema de optimización combinatoria con dos objetivos llamados costo de la red e infactibilidad, los cuales se deben minimizar (Saleh y Tanyimboh, 2013). Si c_{ij} denota el costo por unidad de longitud de la tubería ij , la función objetivo para el costo de la red se puede escribir como:

$$f_1 = \sum_{ij} (c_{ij} \cdot L_{ij}) \quad (3.11)$$

Cualquier diseño propuesto debe satisfacer las condiciones hidráulicas y de topología. Para ello se debe asegurar cierta confiabilidad en el suministro de cada nodo con demanda de agua. Una forma simple de tener confiabilidad hacer que el agua llegue a un nodo por varias rutas de suministro. Además se debe asegurar que se tenga la presión suficiente para cubrir el caudal de demanda, sino se cumple la presión requerida se dice que existe déficit de carga. El déficit de carga total para los nodos que tienen una presión menor a la deseada se expresa como (Saleh y Tanyimboh, 2013):

$$DCR = \sum_{\forall j: H_j < H_j^{des}} (H_j^{des} - H_j) \quad (3.12)$$

donde DCR es el déficit de carga residual para la red completa, H_j es la carga de presión para el nodo j resultante del análisis hidráulico, H_j^{des} es la carga de presión mínima deseada en el nodo j . En caso de que $H_j > H_j^{des}$ el déficit de carga es cero. En otras palabras el déficit de carga residual DCR será de cero únicamente cuando el diseño sea factible (Saleh y Tanyimboh, 2013).

Para optimizar la topología y las dimensiones de tuberías simultáneamente, se introdujo un diámetro de cero para permitir que se generen topologías diferentes al eliminar ciertas conexiones de la red. Un diámetro de cero genera un costo de cero, lo que refleja la situación real de que dicha conexión no existe. Sin embargo, esto resulta en un valor indefinido en la ecuación de pérdidas de carga 2.21, para resolver este problema al momento de la simulación de la red, se consideró como una tubería cerrada en donde no existe una conexión. Dado que la eliminación de conexiones se hace de manera aleatoria, especialmente en las primeras fases de la optimización, algunos nodos o

incluso partes de la red podrían quedar desconectadas de los nodos de abastecimiento de agua. Para evitar esto, se modificó la ecuación 3.12 para incluir la infactibilidad de la red. Para que una red abierta sea factible se requiere que exista como mínimo una ruta de suministro para cada nodo con demanda de agua. Para redes cerradas se requieren al menos dos rutas de suministro para cada nodo. La infactibilidad de la red se cuantificó como el número faltante de rutas de suministro con respecto a las requeridas para cada nodo. La ecuación 3.12 aumentada se define como sigue (Saleh y Tanyimboh, 2013):

$$f_2 = \sum [(H_j^{des} - H_j) + (R - NP_j)] ; \quad \forall j : H_j < H_j^{des} \quad y \quad \forall j : NP_j < R \quad (3.13)$$

donde NP_j es el número de rutas de suministro independientes para el nodo j , mientras que R es el número mínimo de rutas independientes requeridas. Cuando en un nodo sucede que $NP_j > R$, entonces al segundo término de la ecuación 3.13 se le asigna un valor de cero. $R = 1$ para redes abiertas y $R = 2$ para redes cerradas. El valor total de la infactibilidad en la ecuación 3.13 será igual a cero si y solo si un diseño es topológica e hidráulicamente factible (Saleh y Tanyimboh, 2013). Cabe señalar que en ésta última ecuación es posible agregar las condiciones de velocidad máxima y mínima permitidas para el flujo en tuberías.

3.4. Procedimiento de solución

El problema de diseño de redes de riego se resolvió en dos partes, en la primer parte se obtuvo una solución para el problema de trazo de la red (ecuación 3.2) y a partir de ella, se resolvió el dimensionamiento de las tuberías (ecuación 3.6). No se consideró el problema de operación de la red, ya que estos problemas son generalmente estudios de caso particulares que requieren una gran cantidad de supuestos. Las soluciones se obtuvieron mediante el uso de métodos heurísticos que manejan un objetivo y que se usan frecuentemente para resolver problemas de optimización discreta. Una de las metaheurísticas utilizadas fue GA, por ser una de las técnicas de computación evolutiva más utilizadas y que mejores resultados ha dado en este tipo de problemas; por otra

parte se utilizó ACO por ser uno de los métodos inspirados en la naturaleza que mejor desempeño tienen para resolver problemas que se pueden representar mediante gráficos (como es el caso de las redes de tuberías).

Los componentes de ambos problemas de optimización (funciones objetivo, restricciones) y las metaheurísticas se codificaron en el lenguaje de programación Python® 2.7 para facilitar su manejo.

3.4.1. Solución con algoritmos genéticos

Los algoritmos genéticos, operan sobre una población de cromosomas (también llamados individuos) representados por vectores. Cada cromosoma o individuos está compuesto por un conjunto de genes, y a un conjunto de cromosomas se le conoce como una población. En el caso del problema de dimensionamiento de redes un gen es un número entero que representa el diámetro de una tubería, si se toma como ejemplo el dimensionamiento de una red de ocho tuberías, entonces un individuo es un vector de ocho elementos que representa la configuración de diámetros de una red particular donde cada elemento indica el diámetro de una tubería.

Sea Ω es el conjunto de los 14 diámetros de tubo comerciales (en milímetros) permitidos para este problema:

$$\Omega = [25, 51, 76, 102, 152, 203, 254, 305, 356, 406, 457, 508, 559, 610] \quad (3.14)$$

entonces, el individuo D_1 ,

$$D_1 = [11, 6, 9, 0, 8, 6, 6, 0] \quad (3.15)$$

representa una red en la que su primer tubería tiene un diámetro de 508 mm, debido a que el número 11 en D_1 equivale al valor del elemento en la posición 11 de Ω (en el lenguaje Python® los vectores empiezan en la posición 0); en la segunda tubería se tiene un diámetro de 254 mm, en la tercera tubería se tiene un diámetro de 406 mm, y de manera sucesiva se llega a que en la última tubería el diámetro es de 25 mm.

Una población se representada con una matriz que contiene individuos como los mostrados en la expresión 3.15 en cada uno de sus elementos. Al inicio del algoritmo, se crea una población con individuos generados de manera aleatoria, para ello cada

elemento del individuo es un número entero entre 0 y 13 (la longitud de Ω) generado al azar. Una vez que se ha generado la población se realiza la evaluación de la aptitud de los individuos calculando el costo total de la red, utilizando una función que asocia un precio unitario por unidad de longitud a cada uno de los diámetros de tubo. Entonces el costo total se obtiene al sumar los resultados de multiplicar cada precio unitario por su correspondiente longitud de tubería en la red. Posteriormente se asigna un valor de aptitud a todos los individuos de la población, a los individuos de menor costo se les asigna valor de aptitud mayor que a los que tienen un costo elevado. Dado que los individuos se generan de manera aleatoria, se debe verificar la factibilidad de las redes bajo las siguientes condiciones: *a)* se debe cumplir la carga de presión mínima en cada nodo, y *b)* se deben abastecer las demandas de agua. Esto requiere de analizar las redes de tuberías mediante un sistema de ecuaciones obtenido de las ecuaciones 2.20, 2.21 y 2.22. Para facilitar éste análisis se utilizó la librería de funciones de *EPANET 2 Toolkit*, en la que una vez que se le han proporcionado los datos de la red, se obtienen los resultados del análisis (caudal, presión, velocidad del agua, etc.) en forma vectorial y únicamente es necesario verificar que éstos cumplan con las condiciones requeridas. En el caso de que no se cumplan dichas condiciones se aplica una penalización al costo obtenido de la función objetivo. Esta penalización es un valor multiplicativo que se incrementa cada que existe una violación a las condiciones requeridas por la red, así que entre más penalizaciones existan el costo de la red también lo hace y la aptitud del individuo disminuye. Una vez que se ha realizado la evaluación de todos los individuos de la población se aplican los operadores de recombinación a dos individuos (padres) para obtener uno nuevo (hijo) y por último se aplica un operador de mutación con una probabilidad baja. El siguiente paso es realizar una selección de los individuos que tienen mayor aptitud para crear una nueva población y repetir el proceso durante un cierto número de generaciones. Finalmente se obtiene la mejor solución de la población.

Dado que ambos problemas son de optimización combinatoria en los que se puede trabajar directamente sobre las soluciones, se utilizó codificación entera en vez de codificación binaria. No se utilizó ningún criterio de término en especial, en cambio se dejó evolucionar el algoritmo hasta cumplir un cierto número de generaciones determinado previamente. Los valores de los parámetros usados en el algoritmo genético se presentan en el cuadro 3.1, mismos que se tomaron de lo encontrado en la literatura (Afshar

y Jabbari, 2008) y algunos se modificaron de acuerdo a lo observado en ejecuciones preliminares. Se utilizó el operador de mutación por intercambio (*swap* en inglés) debido a que las soluciones del problema son una lista de números enteros obtenidos a partir de un conjunto predeterminado. En lo que respecta a la recombinación entre individuos se utilizó un operador de cruzamiento en dos puntos y para la selección de padres se utilizó un operador conocido como rueda de ruleta. Sin embargo, éste último es conocido por su convergencia prematura en soluciones no óptimas, especialmente en poblaciones de pocos individuos. Para evitar la convergencia prematura se recomienda el uso de alguna versión modificada de la rueda de ruleta, en este caso se utilizó la modificación por escalamiento lineal.

CUADRO 3.1: Valores de los parámetros usados en el algoritmo genético.

Parámetro	Trazo	Dimensionamiento
Población	200	200
Generaciones	1,000	500
Operador de mutación	Intercambio (<i>swap</i>)	Intercambio (<i>swap</i>)
Operador de cruzamiento	En dos puntos	En dos puntos
Operador de selección	Ruleta con escala lineal	Ruleta con escala lineal
Probabilidad de mutación	0.1	0.05
Probabilidad de cruzamiento	0.85	0.85

Finalmente se reporta la mejor solución obtenida de 20 ejecuciones el algoritmo genético para cada uno de los diferentes problemas resueltos. Cabe mencionar que únicamente se codificaron las funciones objetivo de cada problema y se utilizó un código de GA de un módulo llamado *pyevolve* 0.6rc1. El procedimiento utilizado se puede seguir en el diagrama de flujo de la figura 3.2 (*izq.*).

3.4.2. Solución con optimización con colonia de hormigas

El código computacional de ACO se realizó con el paradigma de programación orientada a objetos, en la que únicamente se crearon tres objetos: 1) un objeto “*Ant*”, que es una entidad simple con capacidades limitadas que representa de cierta manera a una hormiga real; 2) un objeto llamado “*Problem*” que representa la instancia del problema a resolver (como un problema de agente viajero o un problema de optimización

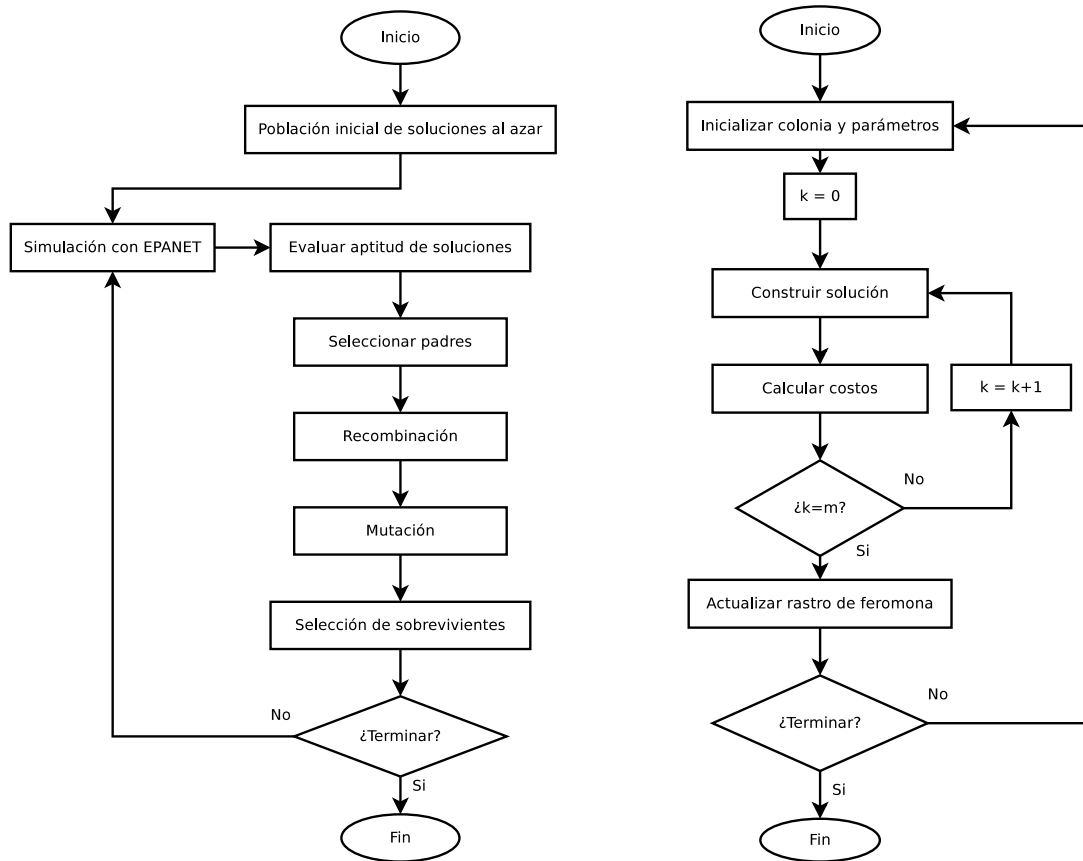


FIGURA 3.2: Diagramas de flujo de GA (izq.) y ACO (der.) para el problema de dimensionamiento de redes.

de redes), contiene los datos del problema como los nodos y las distancias entre ellos; 3) finalmente un objeto llamado “ACO” que utiliza objetos del tipo “Ant” para crear un vector (asemejando a una colonia de hormigas) y resolver la instancia representada en el objeto “Problem”. En la figura 3.3 se muestra un diagrama de clases para la implementación de ACO en el lenguaje Python®.

Usando este código se, obtuvieron soluciones al problema de trazo de redes abiertas siguiendo la metodología de ACO: el primer paso de ésta es crear una colonia de m hormigas, cada una de ellas construye una solución usando la información del problema y el rastro de feromonas. Las soluciones se obtienen de un grafo que representa los componentes de una red de riego. La representación apropiada de un problema de optimización para el dimensionamiento de redes de tuberías requiere de la definición del

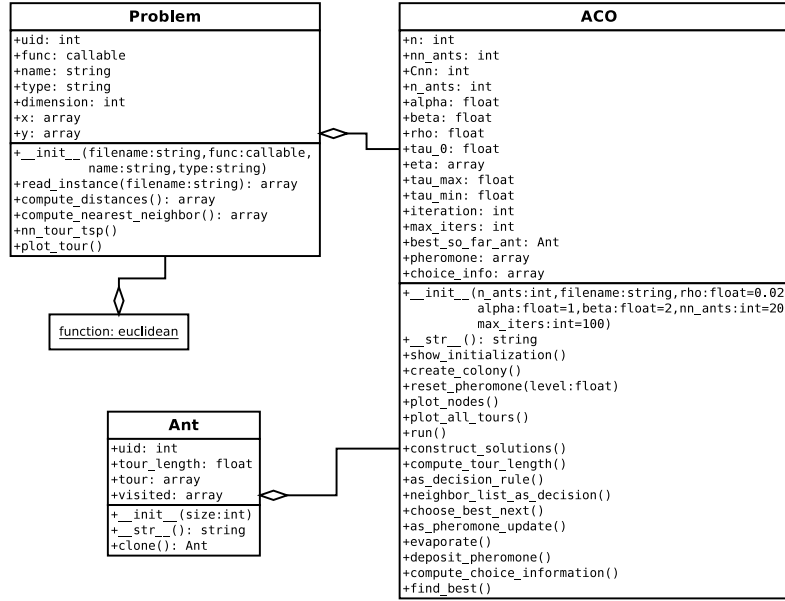


FIGURA 3.3: Diagrama de clases del código realizado para ACO

grafo $G = (D, L, C)$, donde los puntos de decisión D están asociados con cada una de las tuberías de la red. Para el ejemplo de la red de dos circuitos se tienen 8 tuberías, por lo que existen 8 puntos de decisión $(d_1, d_2, \dots, d_8)$. En cada punto de decisión hay un número de opciones L correspondiente a los diámetros de tubo disponibles ϕ_j , en este caso se tienen 14 diámetros disponibles $(\phi_1, \phi_2, \dots, \phi_{14})$ que corresponden a 14 opciones en cada punto de decisión $(l_{i(1)}, l_{i(2)}, \dots, l_{i(14)}; \quad i = 1, 2, \dots, 8)$. Los costos C correspondientes a cada una de estas opciones $(c_{i(1)}, c_{i(2)}, \dots, c_{i(14)}; \quad i = 1, 2, \dots, 8)$ se obtienen mediante el producto del costo unitario de los diámetros de tubo por la longitud de la tubería en consideración. La forma en la que una hormiga artificial crea una solución para el problema de dimensionamiento de redes se muestra en la figura 3.4.

En el caso del trazo el grafo consiste de n puntos de decisión $(d_i; \quad i = 1, 2, \dots, n)$ donde cada punto de decisión está conectado a su punto de decisión adyacente por medio de un conjunto de aristas; por ejemplo, $l_{ij}; \quad 1, 2, \dots, NA_i$ representa el conjunto de aristas disponibles en el punto de decisión i , donde NA_i es el número total de aristas disponibles en el punto de decisión (conexiones hacia los nodos que aún no son parte de la red ramal). Cada hormiga tiene la tarea de construir una solución factible, para ello la hormiga se posiciona en un nodo y se mueve hacia alguno de los que todavía no

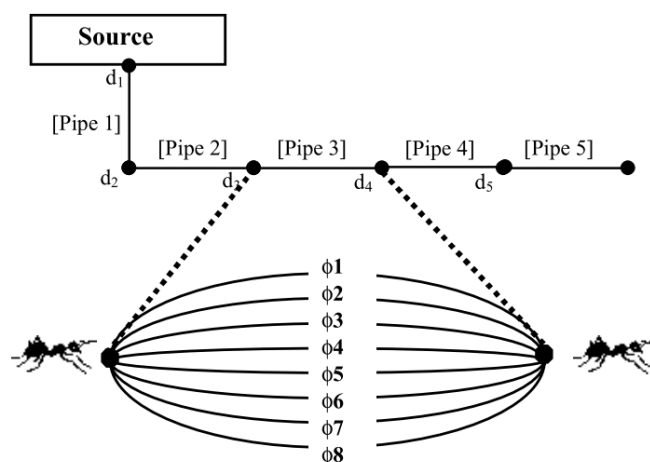


FIGURA 3.4: Construcción de una solución para el dimensionamiento de una red ACO

son parte de la red, la solución está completa una vez que se han unido todos los nodos en una red ramal. Para seleccionar el nodo siguiente, cada hormiga utiliza una *matriz con información para selección* que combina tanto la distancia como un rastro de feromonas, este rastro es una probabilidad que es mayor en los nodos que la misma colonia considera buenas elecciones y disminuye en las malas. Al paso del tiempo la mayor parte de las hormigas habrán seleccionado la red de menor costo como se presenta en la figura 3.2 (*der.*).

Para la solución de los problemas de trazo y dimensionamiento se realizó una implementación de código del algoritmo *Ant System*, puesto que es uno de los algoritmos más sencillos. Si se compara con otras técnicas de búsqueda bio-inspiradas o incluso con computación evolutiva, ACO requiere de muchos parámetros para su ejecución dependiendo del algoritmo utilizado. Los valores usados en este trabajo se muestran en el cuadro 3.2, éstos son valores propuestos en la literatura para solucionar problemas de optimización de redes de distribución de agua. El tamaño de la colonia m es el número de hormigas artificiales y generalmente es igual a la dimensión del problema a resolver n ; y C^{mn} es la longitud de una ruta (costo) generada con un algoritmo de vecinos más cercanos.

CUADRO 3.2: Valores de los parámetros utilizados en la ejecución del algoritmo *Ant System* de ACO.

Parámetro	Símbolo	Valor
Tamaño de colonia	m	n
Iteraciones		20
Tasa de evaporación	ρ	0.5
Importancia de info. heurística	α	1
Importancia de feromona	β	2
Feromona inicial	τ_0	m/C^{nn}

3.4.3. Desarrollo de software para optimización de redes de riego

Tanto los algoritmos de optimización como los algoritmos auxiliares necesarios para resolver estos problemas se programaron en el lenguaje de programación Python® y en conjunto todo el código computacional permitió generar un proyecto de código (términos de desarrollo de software se conoce como *framework*). Este tipo de proyectos se caracterizan por su flexibilidad, ya que en lugar de únicamente proporcionar conjunto de herramientas para resolver problemas, el usuario final tiene la posibilidad de ampliar y modificar las herramientas proporcionadas en un lenguaje de programación libre, fácil y potente como lo es Python®. La accesibilidad de este lenguaje de programación permite que el código desarrollado se pueda distribuir ampliamente, de manera que se tiene un alto potencial de ser utilizado por quienes tengan la necesidad de solucionar problemas similares. La organización del framework se desarrolló con el paradigma de programación orientada a objetos y su organización se presenta en la figura 3.5, cada elemento del esquema representa un módulo que encapsula clases y funciones que tienen una finalidad específica. Los módulos sombreados con color gris se tomaron de otros proyectos existentes a los que se dan los créditos correspondientes.

El módulo principal se llama *optimizer.py* el cual tiene funciones que permiten: *a)* definir el tipo de problema a resolver (trazo o dimensionamiento), *b)* indicar el archivo de texto en donde se encuentran los datos de la instancia del problema, *c)* seleccionar la heurística a utilizar para encontrar las soluciones al problema, y *d)* indicar el número de ejecuciones del algoritmo que se deben realizar. Con base en estos parámetros el módulo buscará las clases y funciones correspondientes para crear la instancia del problema y

resolverlo. Los módulos *GA* y *ACO* tienen el código correspondiente a las metaheurísticas, se tiene un módulo llamado *objectivefunctions.py* que contiene las funciones objetivo de los problemas, el módulo *hydraulic.py* realiza las llamadas a la librería de EPANET 2 para realizar la simulación hidráulica de la red en caso de que alguna función objetivo así lo requiera, y para ello el módulo *epanetfile.py* se encarga de generar los archivos de entrada requeridos para dicha simulación. El módulo *tree_growing.py* contiene el código del algoritmo de Crecimiento de Árbol requerido por algunas funciones objetivo y finalmente, los módulos *utilities.py* y *graphics.py* tienen código requerido para diversas tareas simples requeridas en la optimización y para hacer representaciones gráficas de los problemas y sus soluciones, respectivamente.

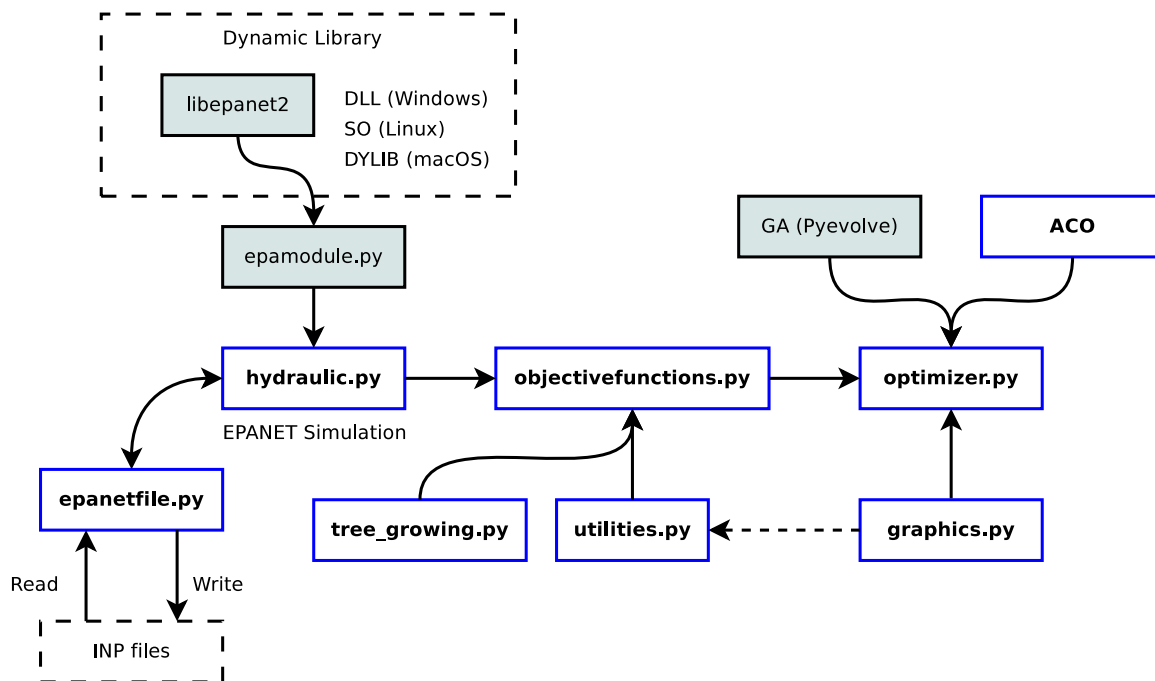


FIGURA 3.5: Esquema del *framework* desarrollado en Python para la optimización del diseño de redes de distribución de agua usando GA y ACO

El desarrollo, codificación y algunas pruebas se realizaron en una computadora con Darwin Kernel version 16.7.0 (macOS® Sierra 10.12), un procesador de 64 bits Intel® Core™i5-5250 @ 1.60 GHz y 8 GB de RAM. Algunas instancias, como el problema de la red de 8×8 , se ejecutaron en una computadora con Linux 4.4, un procesador de 64 bits Intel® Core™i7-4500 @ 1.80 GHz y 8 GB de RAM.

3.5. Ejemplos de aplicación

Las metaheurísticas se pusieron a prueba mediante la solución de redes de referencia usadas en la literatura. Se tienen que poner a prueba con problemas sencillos cuya solución óptima sea conocida debido a que no es posible asegurar que éstos métodos encuentren la solución óptima. Esto permite evaluar si las metaheurísticas son capaces de encontrar soluciones subóptimas aceptables, o en el mejor de los casos la solución óptima del problema. De esta manera se puede observar el comportamiento de la metaheurística al aplicarse en problemas similares de mayor tamaño cuya solución óptima es desconocida.

3.5.1. Red de dos circuitos

Para el problema de dimensionamiento se utilizó la red propuesta por Alperovits y Shamir (1977), que consiste de ocho tramos de tubería acomodados en dos circuitos (como se muestra en la figura 2.6) que es alimentada por una fuente de abastecimiento por gravedad con una carga fija de 210 m. La longitud de cada tramo es de 1000 m y las tuberías tienen un coeficiente de Hazen-Williams de $C = 130$, la demanda de agua en los nodos se supone constante y conocida, y cada uno requiere de una presión mínima de 30 m. La selección de los diámetros se limitó a un conjunto de 14 diámetros comerciales, de manera que el espacio de búsqueda resulta en $14^8 = 1,475,789,056$ redes posibles. Los datos de los nodos para esta red se muestran en el cuadro 3.3.

CUADRO 3.3: Datos de los nodos de la red de dos circuitos

Nodo	Elevación [m]	Presión mínima [m]	Demanda [m^3h^{-1}]
1	210.0	0	-1120.0
2	150.0	30.0	100.0
3	160.0	30.0	100.0
4	155.0	30.0	120.0
5	150.0	30.0	270.0
6	165.0	30.0	330.0
7	160.0	30.0	200.0

Para asegurarse de que el costo de las soluciones obtenidas en el presente trabajo pueda compararse con el costo de las obtenidas en la literatura, se utilizó el conjunto de diámetros comúnmente usado y sus respectivos precios unitarios. En el cuadro 3.4 se presentan los propuestos por Alperovits y Shamir (1977), aunque cabe señalar que estos precios son obsoletos y se utilizan únicamente con fines comparativos.

CUADRO 3.4: Precios unitarios por diámetro de tubo para la red de dos circuitos

Diámetro [<i>in</i>]	Precio unitario
1	2.0
2	5.0
3	8.0
4	11.0
6	16.0
8	24.0
10	32.0
12	50.0
14	60.0
16	90.0
18	130.0
20	170.0

3.5.2. Red de 3×3

La red de 3×3 originalmente propuesta por Geem et al. (2000) es una cuadrícula de 9 nodos, 12 conexiones posibles y una fuente de abastecimiento en el nodo 9 que tiene una elevación de 50 m. Mientras que los demás son nodos de demanda y tienen una elevación de cero. La carga de presión mínima requerida en los nodos es de 30 m, las tuberías tienen una longitud de 100 m y un coeficiente de Hazen-Williams de 130. La red base, que muestra todas las conexiones posibles entre los nodos, se muestra en la figura 3.6. Las demandas de los nodos se muestran en el cuadro 3.5.

El espacio de búsqueda se compone del número de combinaciones posibles entre sus conexiones que es de $2^{12} = 4096$, sin embargo, un gran número de estas soluciones

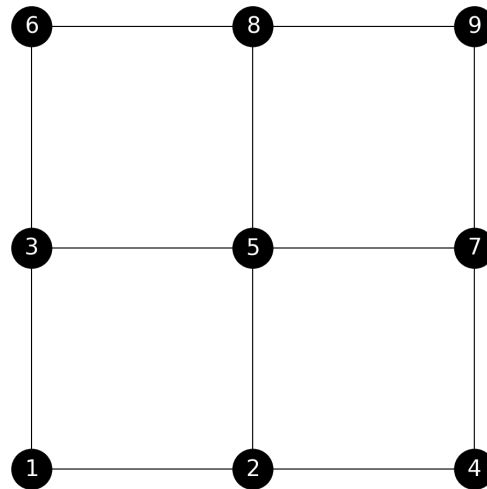


FIGURA 3.6: Red de 3×3 nodos, usada frecuentemente en la literatura para probar el desempeño de metaheurísticas en el diseño de redes.

no son factibles. El espacio de búsqueda se reduce drásticamente al considerar únicamente las soluciones factibles, en este caso el número de árboles que se pueden formar con la red base es de 192. Para este número de combinaciones encontrar las soluciones óptimas con una búsqueda por enumeración resulta fácil.

3.5.3. Red de 8×8

La red de 8×8 , o red de 64 nodos, fue propuesta y resuelta originalmente por Walters y Smith (1995) con base en problemas similares previamente propuestos por Walters y Lohbeck (1993). La gráfica base de esta red es una cuadrícula de 8×8 nodos que tiene una fuente de abastecimiento en el nodo número 64 y se muestra en la figura 3.7.

3.5.4. Red de riego de 12 nodos

La red de riego de propuesta por Ángeles Montiel (2002) es de un proyecto que se localiza en el predio “La Noria” del Ejido Roque 1, en el municipio de Celaya, en el DR 085 La Begoña, Guanajuato. Este predio tiene un área de 80.3 ha y se divide en 19 parcelas con usuarios independientes, el terreno es plano y tiene una pendiente de 0.2 %. La fuente de abastecimiento es una toma del canal lateral 11+141 que conduce agua

CUADRO 3.5: Demandas de agua en cada nodo de la red de 3×3

Nodo	Demanda [$l s^{-1}$]
1	10
2	20
3	10
4	20
5	10
6	20
7	10
8	20
9	-120

desde la presa derivadora La Soria, este punto se representa con el nodo 1 y tiene una carga piezométrica de 84.63 m. El material de las tuberías será PVC con un coeficiente de fricción de Hazen-Williams de 140.

Considerando la longitud de cada parcela, las curvas de nivel y los sentidos de riego se ubicaron las tomas parcelarias o hidrantes como se muestra en la figura 3.8. La carga requerida en cada toma, se determinó considerando las pérdidas de carga que se genera en la tubería lisa que abastece el agua desde el hidrante, las pérdidas de carga generadas por la tubería con compuertas usada para el riego (tubería con salidas múltiples), el desnivel topográfico y la carga media de operación. Las coordenadas de los hidrantes de la red (representadas por nodos), así como los requerimientos de carga de presión y caudal se muestran en el cuadro 3.6.

Otros autores (Afshar y Jabbari (2008); Saleh y Tanyimboh (2013)) utilizan un conjunto de diámetros diferente al del cuadro 3.4. El costo de las soluciones que obtienen se calcula utilizando los precios unitarios del cuadro 3.7.

3.6. Escenarios de prueba

De acuerdo con el esquema de trabajo presentado en la figura 3.1 se preparó un arreglo experimental, haciendo una combinación de los planteamientos de problemas de trazo y dimensionamiento de las redes de prueba descritas en las secciones anteriores y las metaheurísticas descritas.

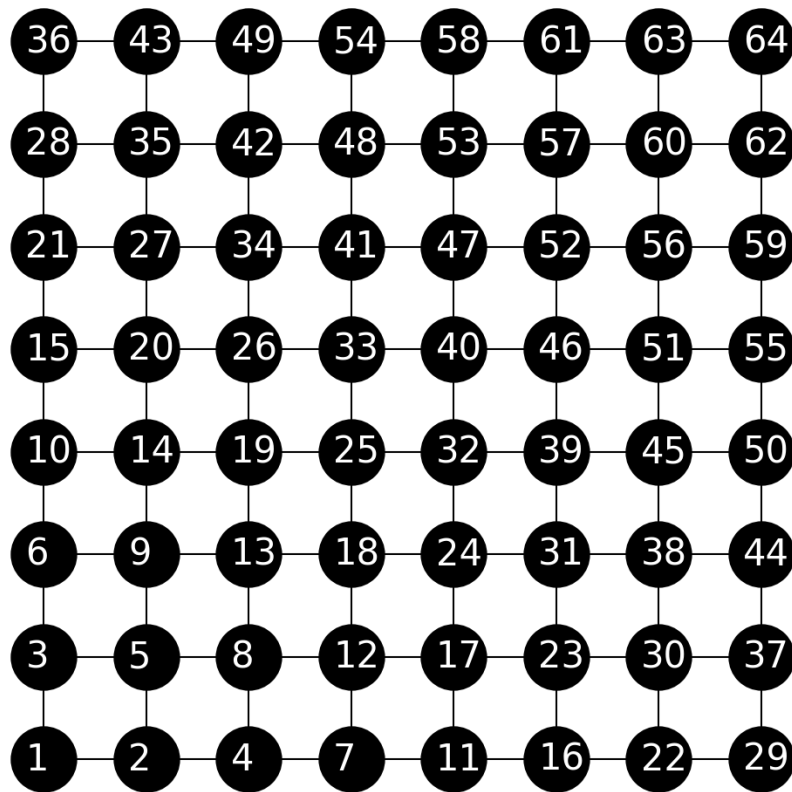


FIGURA 3.7: Red de 8×8 nodos, usada frecuentemente en la literatura para probar el desempeño de metaheurísticas en el diseño de redes

Los escenarios de prueba se arreglaron de tal manera que se pudiera comparar el desempeño de las metaheurísticas GA y ACO en la solución de la misma instancia de un problema.

Para solucionar el problema de trazo se utilizaron los algoritmos de Prim, Kruskal y de crecimiento de árbol en algunos casos para encontrar soluciones de longitud mínima. Estos algoritmos no se consideran metaheurísticas, sin embargo, se agregaron posteriormente a los escenarios de prueba en los casos donde se tuvieron problemas para encontrar soluciones factibles con Algoritmos Genéticos o en redes cuya longitud mínima es desconocida.

Los escenarios de prueba resultantes de esta combinación se muestran en el cuadro 3.8.

¹Estos algoritmos no son metaheurísticas

CUADRO 3.6: Datos de los nodos (hidrantes) de la red de riego de 12 nodos

Nodo	x	y	z	Demanda ($m^3 s^{-1}$)	Carga piezométrica (m)
1	626.33	743.17	84.20	–	84.63
2	586.33	743.17	84.20	0.180	84.53
3	261.33	743.17	83.90	0.120	84.23
4	514.74	567.16	83.80	0.120	84.43
5	176.67	542.28	83.90	0.060	84.42
6	419.91	391.07	83.70	0.060	84.27
7	0.00	462.77	82.60	0.060	82.73
8	141.18	305.57	83.20	0.060	83.92
9	360.86	216.80	83.50	0.060	83.73
10	274.27	93.29	83.10	0.060	83.62
11	157.32	0.00	82.90	0.060	83.29
12	425.24	11.67	83.00	0.060	83.32

CUADRO 3.7: Precios unitarios por diámetro de tubo

Diámetro [mm]	80	100	120	140	160	180	200	220	240	260	280	300	320
Precio [\$/m]	23	32	50	60	90	130	170	300	340	390	430	470	500

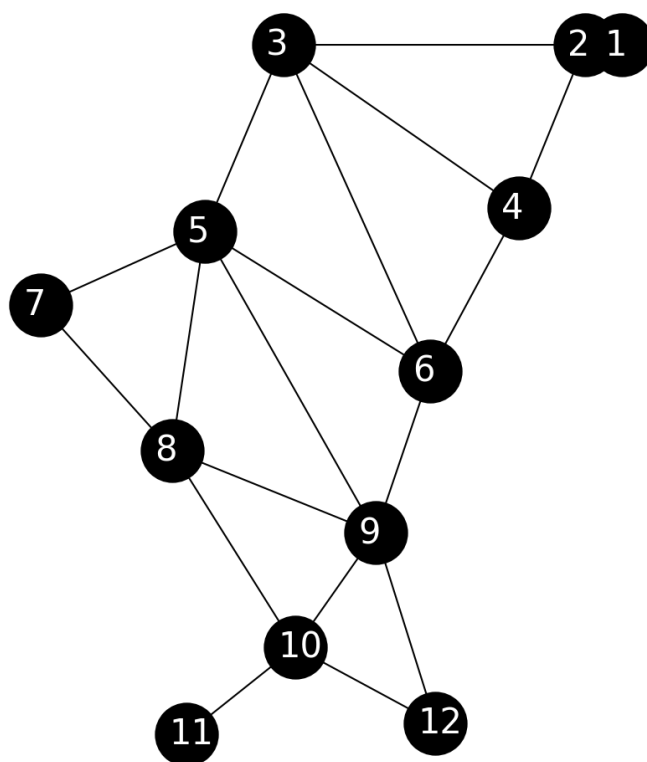


FIGURA 3.8: Distribución de los hidrantes de la red de riego de 12 nodos.

CUADRO 3.8: Escenarios de prueba de las metaheurísticas.

Escenario	Problema	Red de prueba	Metaheurística
1	Trazo	Red de 3×3	GA
2			ACO
3			GA
4		Red de 8×8	ACO
5			Kruskal; Prim ¹
6			Crecimiento de árbol ¹
7		Red de 12 nodos	GA
8			ACO
9			Kruskal; Prim ¹
10	Dimensionamiento	Red de dos circuitos	GA
11			ACO
12		Red de 3×3	GA
13			ACO
14		Red de 8×8	GA
15			ACO
16		Red de 12 nodos	GA
17			ACO

Capítulo 4

Resultados y discusión

4.1. Problema de trazo

En esta sección se presentan las soluciones obtenidas para los problemas de trazo de las redes descritas anteriormente. Los parámetros usados en los Algoritmos Genéticos se muestran en el cuadro 3.1, mientras que los parámetros para Optimización con Colonia de Hormigas se muestran en el cuadro 3.2. En los casos donde se utilizaron valores diferentes a los presentados se señalan claramente.

4.1.1. Red de 3×3

Encontrar una solución para el problema de trazo de la red de 3×3 (red de 9 nodos) como se plantea en la ecuación 3.2, es muy sencillo. Las soluciones óptimas para esta red se obtienen dibujando conexiones perpendiculares entre nodos contiguos, de manera que se forme una topología de árbol de 800 unidades de longitud. Cualquier red que incluya conexiones en diagonal o cruzadas no será una solución óptima.

En este caso se realizó el trazo de la red usando el espacio de búsqueda que resulta de todas las conexiones posibles entre los 9 nodos que es de $2^{36} = 68,719,476,736$ incluyendo las soluciones no factibles; esto con la finalidad de conocer la eficiencia de las metaheurísticas ante un problema más difícil que el descrito en el capítulo anterior.

Tanto GA como ACO fueron capaces de encontrar soluciones óptimas, ejemplos de éstas redes se muestran en la figura 4.1, que son las mejores redes obtenidas de 20 ejecuciones.

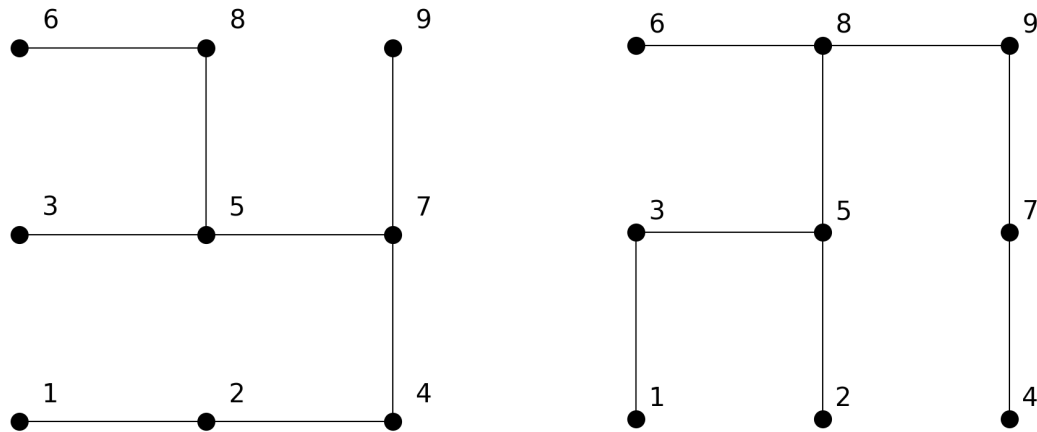


FIGURA 4.1: Ejemplos de las mejores soluciones para el trazo de la red de 3×3 encontradas con Algoritmos Genéticos y Optimización con Colonia de Hormigas. Todas las soluciones óptimas para este problema tienen un patrón similar.

En el cuadro 4.1 se presenta una comparación las soluciones obtenidas para el trazo de la red de 3×3 con 20 ejecuciones de ambas metaheurísticas. Para cada ejecución se presenta la solución mínima obtenida, su diferencia con respecto al óptimo global y el promedio de generaciones o iteraciones en las que se obtuvo dicha solución.

En el caso de GA se obtuvieron resultados aceptables usando una población de 80 individuos y 50 generaciones, por lo tanto se necesitaron 4,000 evaluaciones de la función objetivo. Con esta configuración la metaheurística obtuvo la solución óptima en 2 ejecuciones de las 20 realizadas y para ello se requirieron en promedio 20 generaciones. El promedio de la solución mínima obtenida resultó de 870.58 con una desviación estándar de 45.97. Se observa que la solución promedio está muy lejana a la óptima y su desviación estándar es muy grande, además la solución óptima se obtuvo únicamente en el 10 % de las ejecuciones. Con la finalidad de mejorar los resultados se hizo un ajuste de los parámetros del algoritmo, específicamente se incrementó el número de generaciones y el tamaño de población. De manera que se realizaron tres pruebas adicionales, la primera con una población de 100 individuos y 100 generaciones; la segunda con una población de 200 individuos y 200 generaciones; y la tercera con 200 individuos y 500 generaciones. Esta última configuración es la usada en la comparación del cuadro 4.1, en la que se obtuvo la solución óptima en 10 de las 20 ejecuciones, el promedio de

la solución mínima obtenida fue de 822.78 con una desviación estándar de 25.05, para ello se necesitó un promedio de 163 generaciones. La convergencia de GA se muestra en la figura 4.2 (*izq.*).

Para ACO se necesitó una colonia de al menos $m = 5$ hormigas artificiales, 10 iteraciones y una tasa de evaporación de $\rho = 0.5$, se necesitaron 50 evaluaciones de la función objetivo para obtener la solución óptima, misma que se obtuvo en todas las ejecuciones de ACO en un promedio de 4 iteraciones. La convergencia de ACO se muestra en la figura 4.2 (*der.*).

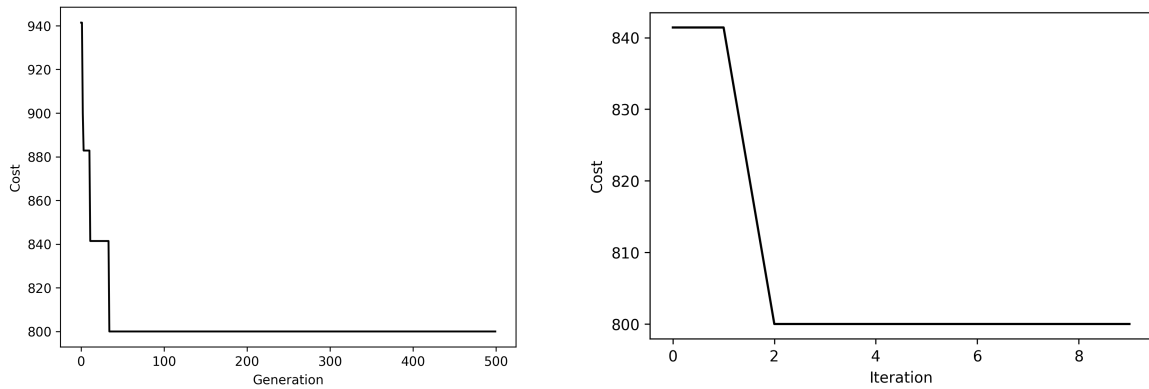


FIGURA 4.2: Convergencia de las soluciones para el trazo de la red de 3×3 obtenidas con Algoritmos Genéticos (*izq.*) y Optimización con Colonia de Hormigas (*der.*).

Una de las razones por las que la convergencia de ACO (específicamente el algoritmo *Ant System*) es tan rápida es porque el algoritmo asigna una probabilidad de selección muy alta a conexiones entre nodos muy cercanos y una probabilidad muy baja a las conexiones entre nodos lejanos. Aunque este efecto es deseable, la diferencia entre estas dos probabilidades es tan grande que el valor de la probabilidad del rastro de feromonas de las hormigas tiene un efecto mínimo en la construcción de soluciones, por lo tanto las hormigas artificiales construyen soluciones con base en las distancias más cortas. Una forma de equilibrar la importancia de la información heurística (distancias) y el rastro de feromonas, es mediante el ajuste de los parámetros α y β de ACO. Sin embargo, en este ejemplo no se observó efecto significativo al usar diferentes valores de estos parámetros.

¹Solo se incluye a las ejecuciones que encuentran la solución óptima.

CUADRO 4.1: Soluciones para el trazo de la red 3×3 obtenidas con las metaheurísticas Algoritmos Genéticos (GA) y Optimización con Colonia de Hormigas (ACO).

Ejecución	GA (200 ind., 500 gen.)			ACO		
	Mínimo [m]	Diferencia [%]	Generaciones requeridas	Mínimo [m]	Diferencia [%]	Iteraciones requeridas
1	882.84	10.36	24	800	0	3
2	800.00	0.00	51	800	0	7
3	800.00	0.00	403	800	0	5
4	800.00	0.00	304	800	0	5
5	841.42	5.18	12	800	0	5
6	800.00	0.00	35	800	0	3
7	800.00	0.00	179	800	0	4
8	841.42	5.18	67	800	0	6
9	800.00	0.00	58	800	0	3
10	841.42	5.18	16	800	0	3
11	841.42	5.18	52	800	0	4
12	841.42	5.18	32	800	0	6
13	841.42	5.18	30	800	0	4
14	841.42	5.18	23	800	0	4
15	800.00	0.00	41	800	0	5
16	800.00	0.00	63	800	0	4
17	841.42	5.18	291	800	0	3
18	841.42	5.18	40	800	0	5
19	800.00	0.00	440	800	0	5
20	800.00	0.00	54	800	0	5
Promedio:	822.78	2.85	163 ¹	800.00	0.00	4
Desv. Est.:	25.05	3.13	160.27 ¹	0.00	0.00	1.15

Las soluciones obtenidas para el trazo de la red tienen una longitud mínima y por lo tanto son aceptables. Debido a que no se considera el suministro de agua se observa que en ocasiones se obtienen recorridos muy largos para conectar la fuente de abastecimiento con los nodos de demanda. Por ejemplo, la fuente de abastecimiento de la red se encuentra en el nodo 9, entonces es deseable que los nodos más cercanos, como el 7 y 8, se conecten directamente o mediante la ruta más corta. Sin embargo, es poco probable que esto suceda ya que el algoritmo no toma en cuenta la importancia que tiene el suministro de agua hacia los nodos.

En la literatura se pueden encontrar ejemplos de optimización del trazo de esta red usando diferentes metaheurísticas (Walters y Lohbeck, 1993; Geem et al. (2000); Afshar y Mariño (2006)), sin embargo, el planteamiento del problema es diferente. El trazo de la red si considera el suministro de agua a todos los nodos, pero no se considera el dimensionamiento de las tuberías sino que el costo total de la red es una función del caudal en la tubería. Por esta razón no es posible hacer una comparación directa de las soluciones obtenidas para el trazo de esta red.

4.1.2. Red de 8×8

Las soluciones óptimas para el problema de trazo de la red de 8×8 (red de 64 nodos) como se ha planteado en la ecuación 3.2, incluyen cualquier red ramal que tenga conexiones perpendiculares entre nodos contiguos, lo que resulta en una red de 630 unidades de longitud. Por lo tanto cualquier red que incluya conexiones en diagonal o cruzadas no será una solución óptima. En el enfoque de solución original (Walters y Lohbeck (1993); Walters y Smith (1995)) se delimita un conjunto de 112 conexiones posibles entre los nodos, formando la cuadrícula que se muestra en la figura 3.7. En este caso se utilizó el espacio de búsqueda que consiste en todas las conexiones posibles entre los nodos que tiene un tamaño de 2^{2016} , ya que también incluye conexiones en diagonal entre los nodos.

Las soluciones encontradas con ACO se presentan en el cuadro 4.2 donde se observa que el óptimo global se encuentra en el 100 % de las ejecuciones realizadas, para lo que se requiere un promedio de 15 iteraciones en cada una. Estas soluciones se obtuvieron usando los parámetros mostrados en el cuadro 3.2, sin embargo en este caso se usó un tamaño de colonia de $m = 10$ hormigas artificiales, 20 iteraciones y una tasa de

evaporación de $\rho = 0.5$. La red mostrada en la figura 4.3 es la mejor obtenida de 20 ejecuciones del algoritmo. Todas las soluciones óptimas para el trazo de esta red tienen este aspecto, aunque las conexiones pueden ser diferentes.

Se puede observar que, como en el caso de la red de 3×3 , la falta de algún criterio para considerar el suministro de agua, hace posible que se obtengan redes con una configuración en la que se provee agua mediante una trayectoria muy larga a nodos que son cercanos a la fuente de abastecimiento, lo cual no es deseable para el siguiente paso que es la determinación de los diámetros de las tuberías.

CUADRO 4.2: Soluciones para el trazo de la red de 8×8 obtenidas con Optimización con Colonia de Hormigas

Ejecución	Mínimo [m]	Diferencia [%]	Iteraciones requeridas
1	630	0	11
2	630	0	13
3	630	0	14
4	630	0	16
5	630	0	17
6	630	0	12
7	630	0	15
8	630	0	18
9	630	0	12
10	630	0	16
11	630	0	15
12	630	0	19
13	630	0	17
14	630	0	12
15	630	0	13
16	630	0	13
17	630	0	16
18	630	0	12
19	630	0	17
20	630	0	14
Promedio:	630	0	15
Desv. Est:	0	0	2.33

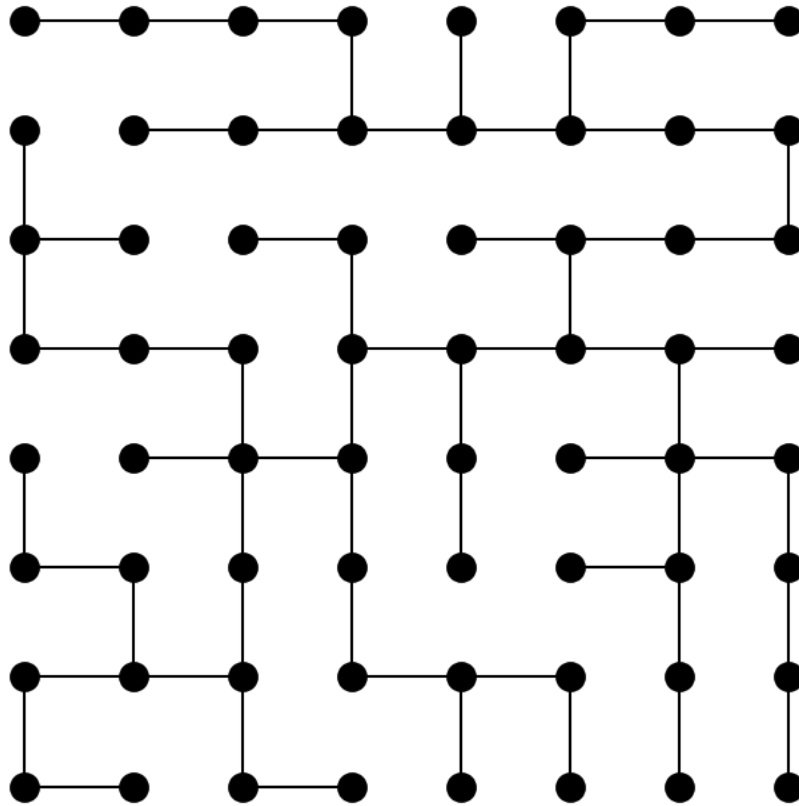


FIGURA 4.3: Solución óptima para el trazo de la red de 8×8 encontrada por la metaheurística Optimización con Colonia de Hormigas.

En la figura 4.4 se muestra la convergencia de ACO, donde el costo mostrado es la solución mínima obtenida en las 20 ejecuciones. Se observa que las soluciones formadas en las primeras iteraciones tienen un costo (longitud de la red) elevado y en pocas iteraciones el algoritmo va encontrando soluciones de menor costo hasta que finalmente se converge a la solución óptima.

Por otra parte GA fue incapaz de obtener soluciones factibles para este problema a pesar de que se usaron valores mayores a los mostrados en el cuadro 3.1 para el tamaño de población (100 y 200 individuos) y el número de generaciones (500, 1000, 10000). Usar valores altos en estos parámetros del algoritmo, hace que el tiempo de ejecución se incremente de manera proporcional. Particularmente, en el hardware descrito en el capítulo 3 cada ejecución se realizó en un tiempo promedio de 50 minutos, por lo que se

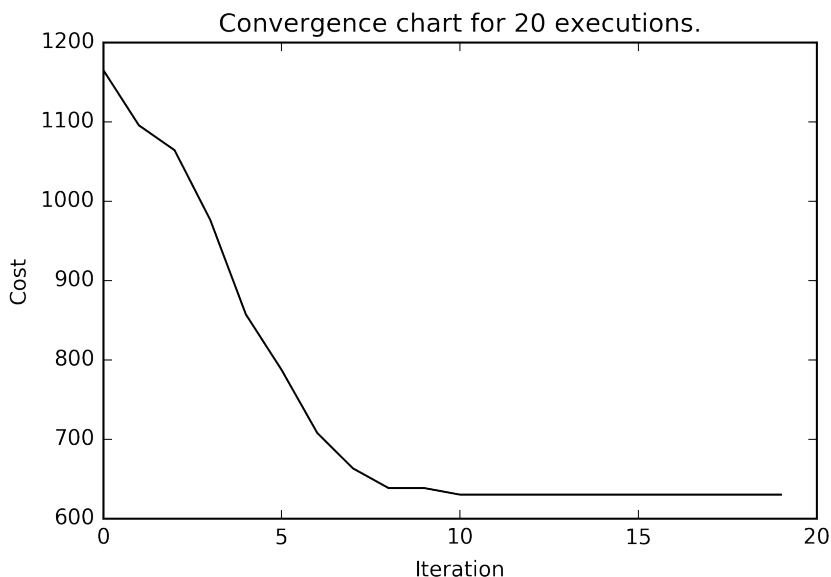


FIGURA 4.4: Convergencia de las soluciones obtenidas con Optimización con Colonia de Hormigas para el trazo de la red de 8×8 .

requirieron alrededor de 1,000 minutos (16.67 horas) para completar las 20 ejecuciones. En cambio cada ejecución de ACO requirió de un promedio de 7 minutos y un total de 140 minutos (2.3 horas) para las 20 ejecuciones del algoritmo.

Una forma de reducir el espacio de búsqueda y, por consiguiente, el tiempo de ejecución del algoritmo, es proporcionar un conjunto reducido de las conexiones permitidas entre los nodos de la red, ésto se hizo usando la gráfica base mostrada en la figura 3.7 que es una cuadrícula con 112 conexiones permitidas de manera que el espacio de búsqueda se reduce a $2^{112} = 5.19 \times 10^{33}$ combinaciones posibles.

Otra forma de agilizar la exploración del espacio de búsqueda es proponiendo únicamente soluciones factibles para su evaluación, para este caso se realizaron algunas pruebas usando un *algoritmo de crecimiento de árbol* (Walters y Smith, 1995). Éste algoritmo elige un nodo raíz de manera aleatoria y va agregando los nodos que se encuentren desconectados de manera iterativa hasta que finalmente se conectan todos en una red ramal. Todos los nodos que se van agregando se escogen mediante una función de probabilidades uniforme discreta y no se toma en cuenta la proximidad entre ellos. Por

estas razones las redes formadas por el algoritmo de crecimiento de árbol son soluciones factibles desde el punto de vista de los GA. Específicamente el algoritmo de crecimiento de árbol se utilizó para crear la población inicial de individuos y después se continuó con la ejecución del GA de manera normal.

Auxiliándose de estas dos modificaciones, el GA fue capaz de encontrar las soluciones de costo mínimo para esta red desde la primera generación usando los mismos parámetros.

4.1.3. Red de riego de 12 nodos

La solución para la red de riego propuesta por Ángeles Montiel (2002) no es tan fácil de visualizar como en las otras redes tipo cuadrícula. Debido a que la distribución de los nodos es irregular se tienen que usar algoritmos para encontrar la red de menor longitud. En este caso se usaron algoritmos de *Árbol de Expansión Mínima* para encontrar la solución óptima del trazo de la red. Los algoritmos usados fueron: *a)* Kruskal y *b)* Prim, los cuales hacen una búsqueda exhaustiva y conectan los nodos más próximos, de manera que se obtiene una red con topología de árbol de longitud mínima. La solución óptima obtenida con estos algoritmos tiene una longitud de 1,946.04 unidades.

En el cuadro 4.3 se presenta una comparación las soluciones obtenidas para el trazo de la red de 12 nodos con 20 ejecuciones de las metaheurísticas GA y ACO. Para cada ejecución se presenta la solución mínima obtenida, su diferencia con respecto al óptimo global y el promedio de generaciones o iteraciones en las que se obtuvo dicha solución.

Se obtuvo la solución óptima (igual a la encontrada con los algoritmos de Kruskal y Prim) usando la metaheurística ACO configurada con los parámetros del cuadro 3.2, un tamaño de colonia de $m = 5$ hormigas artificiales y 10 iteraciones, es decir se necesitaron 50 evaluaciones de la función objetivo para encontrar el óptimo. La solución se encontró en 9 de las 20 ejecuciones realizadas, esto es el 45 % de las veces y se obtuvo en un promedio de 9 iteraciones por cada ejecución del algoritmo. El promedio de la solución mínima obtenida resultó de 1,966.15 con una desviación estándar de 28.31, que es bastante cercano al óptimo.

Por otro lado, la metaheurística GA con los parámetros especificados en el cuadro 3.1 no pudo encontrar la solución óptima y en su lugar convergió en una solución subóptima con un valor de 1,987.72. GA tuvo un desempeño muy pobre puesto que

requirió de 200,000 evaluaciones de la función objetivo (una población de 200 individuos durante 1,000 generaciones) para llegar a la solución subóptima. El promedio de la solución mínima obtenida es de 2,424.58 con una desviación estándar de 203.39, valores que se encuentran alejados del óptimo. Además la solución subóptima se obtuvo únicamente en una de las 20 ejecuciones (5 % de las veces). En general las soluciones obtenidas en cada una de las ejecuciones se alejaron en promedio un 24.59 % con respecto a la solución óptima.

CUADRO 4.3: Soluciones para el trazo de la red de 12 nodos obtenidas con Algoritmos Genéticos (GA) y Optimización con Colonia de Hormigas (ACO).

Ejecución	GA (200 ind., 1000 gen.)			ACO		
	Mínimo [m]	Diferencia [%]	Generaciones requeridas	Mínimo [m]	Diferencia [%]	Iteraciones requeridas
1	2,619.59	34.61	56	1,946.04	0.00	10
2	2,546.37	30.85	172	1,946.04	0.00	10
3	2,391.37	22.88	807	1,946.04	0.00	8
4	2,165.37	11.27	705	2,003.03	2.93	10
5	2,503.00	28.62	410	1,946.04	0.00	10
6	2,497.63	28.34	144	1,995.51	2.54	7
7	2,291.96	17.78	147	1,989.42	2.23	10
8	2,607.94	34.01	629	1,946.04	0.00	9
9	2,374.41	22.01	276	1,964.43	0.95	7
10	2,479.81	27.43	132	1,959.66	0.70	9
11	2,381.38	22.37	235	1,946.04	0.00	7
12	2,443.90	25.58	141	2,000.65	2.81	8
13	2,481.09	27.49	891	2,052.50	5.47	5
14	2,922.94	50.20	114	1,959.66	0.70	8
15	2,416.58	24.18	212	1,964.43	0.95	10
16	1,987.72	2.14	557	1,946.04	0.00	9
17	2,479.66	27.42	317	1,946.04	0.00	10
18	2,542.22	30.64	682	1,959.66	0.70	8
19	2,093.71	7.59	364	1,959.66	0.70	10
20	2,264.88	16.38	470	1,946.04	0.00	8
Promedio:	2,424.58	24.59	373	1,966.15	1.03	9
Desv. Est.:	203.39	10.45	256.86	28.31	1.45	1.12

En la figura 4.5 se muestran las representaciones gráficas de las soluciones para el trazo de la red obtenidas con Algoritmos Genéticos (izquierda) y Optimización con Colonia de Hormigas (derecha).

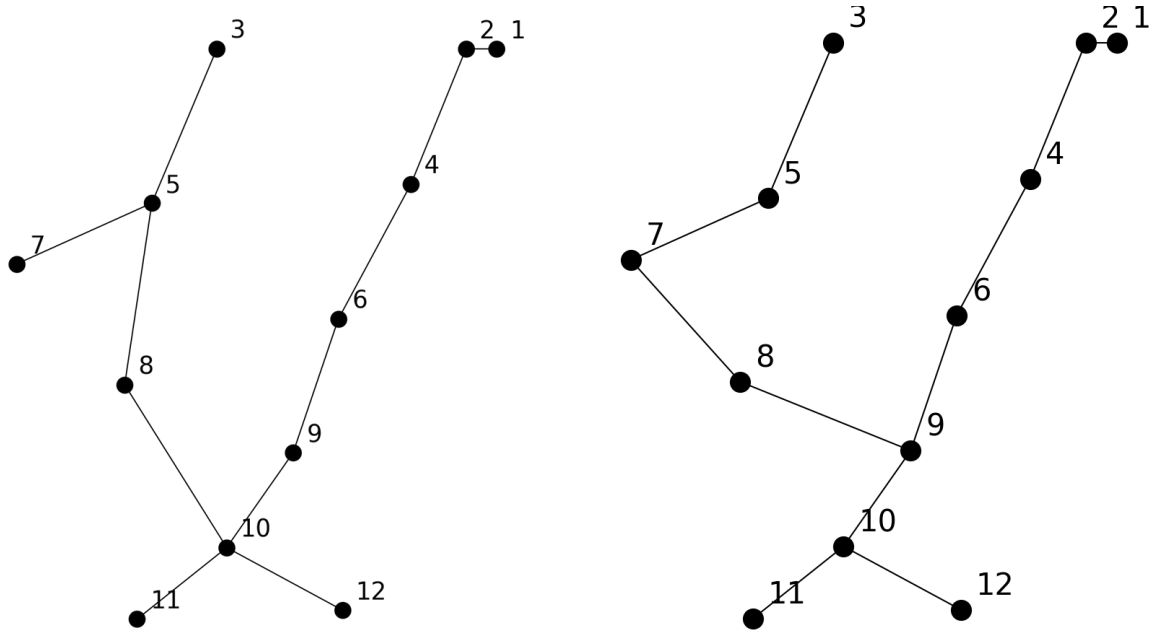


FIGURA 4.5: Mejores soluciones para el trazo de la red de riego de 12 nodos, a la izquierda se presenta la obtenida con Algoritmos Genéticos que tiene una longitud mínima de 1,987.72 unidades. Mientras que a la derecha la solución obtenida con Optimización con Colonia de Hormigas y tiene una longitud de 1,946.04 unidades.

Esta red se ha estudiado muy poco, sin embargo, se realizó una comparación con los resultados previos encontrados en la literatura. En el cuadro 4.4 se muestra el valor de las mejores soluciones reportadas y el método con el que se obtuvo dicho valor. Se puede observar que el autor original obtuvo una solución muy cercana al mínimo, sin embargo dicha solución fue modificada manualmente para ajustarse a linderos parcelarios.

Entre la variedad de métodos que se han utilizado para solucionar el trazo de esta red, se puede observar que únicamente dos metaheurísticas fueron capaces de encontrar la solución óptima: ACO y DE. Claramente ACO es la más eficiente en cuanto al número de evaluaciones de la función objetivo requeridas para encontrar la solución

CUADRO 4.4: Soluciones reportadas para el trazo de la red de riego de 12 nodos usando diferentes metaheurísticas: Programación Lineal (LP), Evolución Diferencial (DE), Colonia Artificial de Abejas (ABC), Algoritmos Genéticos (GA) y Optimización con Colonia de Hormigas (ACO).

Solución	Método	Evaluaciones	Costo [m]
Ángeles Montiel (2002)	LP	No Reportado	2, 212.00
Ponce Pacheco (2013)	DE	1, 100, 000	1, 946.04
Ponce Pacheco (2013)	ABC	1, 100, 000	2, 113.86
Este trabajo	Kruskal; Prim	No Aplica	1, 946.04
Este trabajo	GA	200, 000	1, 987.72
Este trabajo	ACO	50	1, 946.04

óptima. Sin embargo, dado que no se reporta el número de evaluaciones mínimas en las que DE encuentra la solución, no es posible hacer una comparación justa puesto que las evaluaciones se obtuvieron directamente del número de ejecuciones del algoritmo.

En lo que respecta a las metaheurísticas de este trabajo, en la figura 4.6 se muestra el comportamiento de la solución mínima con respecto a las iteraciones de ACO y las generaciones de GA. Se puede observar que la convergencia de ACO a la solución óptima es muy rápida.

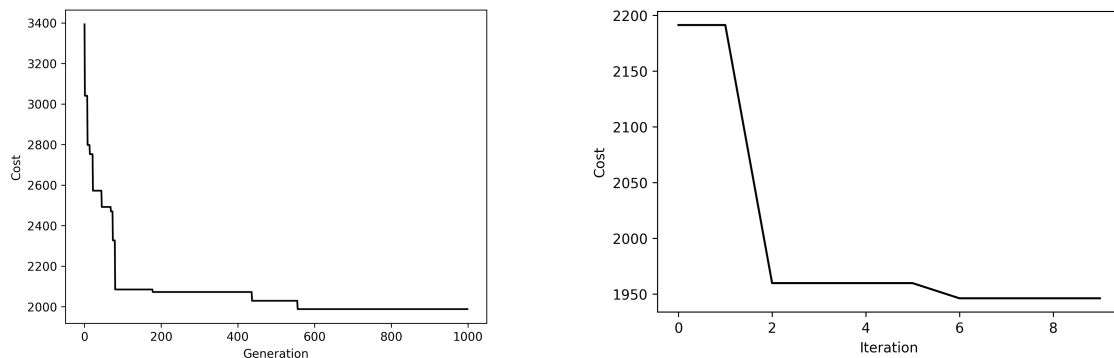


FIGURA 4.6: Convergencia de las soluciones para el trazo de la red de riego de 12 nodos obtenido con Algoritmos Genéticos (*izq.*) y Optimización con Colonia de Hormigas (*der.*).

Para instancias muy grandes de problemas de trazo de redes, el rendimiento de algoritmo *Ant System* se reduce, por lo que se recomienda probar otros algoritmos disponibles dentro de la familia de ACO, como el *MAX-MIN Ant System* (Stützle

y Hoos, 2000) que ha demostrado tener un buen rendimiento en problemas de combinatoria como el agente de viajero, asignación cuadrática e incluso para el trazo de redes.

4.2. Problema de dimensionamiento

En esta sección se presentan las soluciones obtenidas para los problemas de dimensionamiento de las redes descritas anteriormente. Los parámetros usados en GA se muestran en el cuadro 3.1 en la columna de dimensionamiento, mientras que los parámetros para ACO se muestran en el cuadro 3.2. En los casos donde se utilizaron valores diferentes a los presentados se señalan claramente.

4.2.1. Red de dos circuitos

La red de dos circuitos es una red de trazo conocido y consiste de ocho tramos de tubería acomodados de manera que se forman dos circuitos, como se puede ver en la figura 2.6. Todos los datos de red son conocidos, de tal forma que el único elemento que se puede variar son los diámetros de cada tramo de tubería. En el cuadro 3.4 se proporcionan los diámetros de tubo con sus respectivos precios unitarios disponibles para este problema.

La solución mínima encontrada por GA, consiste en una combinación de diámetros para cada tubería y se representa mediante un vector de valores numéricos, donde cada elemento es el diámetro en milímetros para cada una de las ocho tuberías de la red. Los valores de esta solución muestran en el cuadro 4.5.

Las soluciones obtenidas para el dimensionamiento de esta red usando GA se presentan en el cuadro 4.6 donde se muestran los valores mínimos obtenidos en cada una de las 20 ejecuciones, su diferencia en porcentaje con respecto a la solución mínima y las generaciones requeridas para encontrar dicha solución. Se puede observar que la solución subóptima se obtuvo en 1 de las 20 ejecuciones (5 % de las veces) y se requirieron de 437 generaciones del algoritmo para encontrarla. La solución mínima obtenida tiene un valor promedio de 480,750 con una desviación estándar de 42,270 los cuales

CUADRO 4.5: Diámetros de tuberías de la solución obtenida con Algoritmos Genéticos (GA) para la red de dos circuitos, comparados con los reportados en la literatura obtenidos con Algoritmo de Transición de Estado (STA).

Tubería	GA (Este trabajo)		STA (Zhou et al., 2015)
	Diámetro [<i>mm</i>]	Diámetro [<i>in</i>]	Diámetro [<i>in</i>]
1	457	18	18
2	356	14	10
3	356	14	16
4	152	6	4
5	356	14	16
6	102	4	10
7	356	14	10
8	254	10	1

son valores que se alejan mucho de la soluciones reportadas en la literatura, por lo tanto se puede decir que el algoritmo tiene poca robustez para obtener soluciones a este problema. Con esta configuración se necesitaron de 100,000 evaluaciones de la función objetivo para obtener esta solución.

La convergencia de GA se puede observar en la figura 4.7, en este caso se usó un valor de 500 generaciones. No se usó un número mayor de generaciones ya que el espacio de búsqueda de los problemas de dimensionamiento, como los planteados en este trabajo, es menor al espacio de búsqueda de los problemas de trazo. Esto se debe a que las opciones a elegir se limitan a un conjunto determinado de diámetros comerciales en el que todas las soluciones son factibles.

En el cuadro 4.7 se presenta una comparación de las mejores soluciones reportadas en la literatura, con la finalidad de evaluar la calidad de la obtenida en el presente trabajo. Esta comparación se basa en la realizada previamente por Zhou et al. (2015) donde se muestran las soluciones obtenidas con algoritmos como LPG y STA (fijo y variable). El costo total de la red se calculó con los mismos precios unitarios por metro de tubería manejados por dichos autores con la finalidad de poder hacer comparaciones.

Cabe señalar que, las primeras soluciones para esta red se obtuvieron con un método llamado *Linear Programming Gradient* (LPG) que está basado en técnicas de programación lineal y que fue usado por Alperovits y Shamir (1977) para obtener dicha

CUADRO 4.6: Soluciones para el dimensionamiento de la red de dos circuitos obtenidas con Algoritmos Genéticos

Ejecución	Mínimo [\$]	Diferencia [%]	Generaciones requeridas
1	509,000	19	84
2	616,000	44	448
3	429,000	0	437
4	466,000	9	214
5	515,000	20	269
6	442,000	3	450
7	550,000	28	65
8	468,000	9	86
9	454,000	6	218
10	462,000	8	211
11	480,000	12	285
12	462,000	8	485
13	461,000	7	255
14	477,000	11	312
15	454,000	6	305
16	486,000	13	95
17	462,000	8	122
18	467,000	9	248
19	453,000	6	300
20	502,000	17	215
Promedio:	480,750	12	255
Desv. Est.:	42,270	10	129

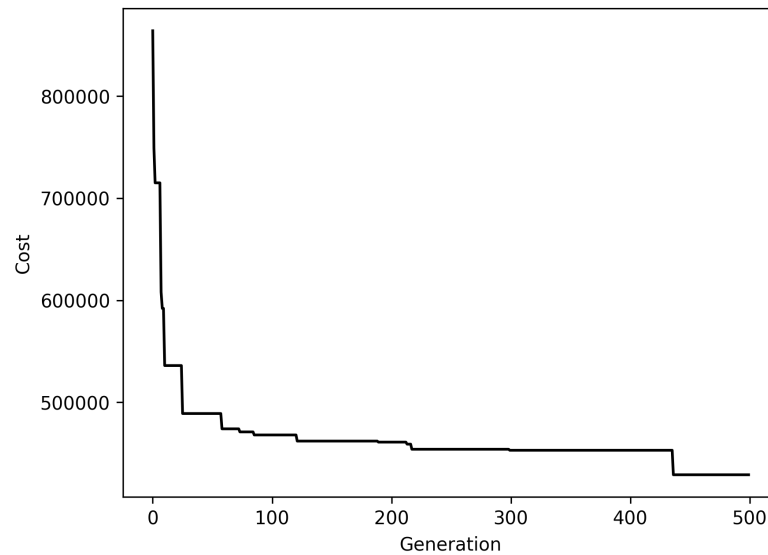


FIGURA 4.7: Convergencia de las soluciones obtenidas para el dimensionamiento de la red de dos circuitos con Algoritmos Genéticos.

solución. A partir de esto Goulter et al. (1986) y Kessler y Shamir (1989) hicieron algunas modificaciones y análisis adicionales para obtener mejores soluciones. Estas son las mejores soluciones reportadas en la literatura. Por otro lado Zhou et al. (2015) proponen una heurística llamada *State Transition Algorithm* y utilizan este problema para probar su desempeño.

Aunque se puede notar que hay mejores soluciones a la encontrada en este trabajo, especialmente la obtenida por Kessler y Shamir (1989), se debe tener presente que estas soluciones utilizan tuberías segmentadas, de esta manera en el trayecto de la tubería se pueden tener dos o tres segmentos con diferentes diámetros en lugar de mantener un solo diámetro en toda su longitud. Este enfoque tiene un planteamiento diferente de la función objetivo y permite que se puedan encontrar soluciones de menor costo. Sin embargo, el planteamiento usado en este trabajo aún puede compararse con el empleado por Zhou et al. (2015) puesto que es de los más usados en la literatura de dimensionamiento de redes.

En el caso del dimensionamiento de redes de riego es necesario hacer un análisis a detalle para verificar el cumplimiento de las condiciones hidráulicas de la red, específicamente que la carga de presión y velocidad del flujo se encuentren dentro de los

CUADRO 4.7: Comparación de la solución obtenida con Algoritmos Genéticos (GA) para la red de dos circuitos, con las reportadas usando Gradiente de Programación Lineal (LPG) y Algoritmo de Transición de Estado (STA).

Solución	Método	Costo [\$]
Alperovits y Shamir (1977)	LPG	497,525
Goulter et al. (1986)	LPG	435,015
Kessler y Shamir (1989)	LPG	417,500
Zhou et al. (2015)	STA fijo	419,000
Zhou et al. (2015)	STA variable	419,000
Este trabajo	GA	429,000

límites permisibles ya que de lo contrario se pueden tener fallas en la operación de la red. Además, en este tipo de problemas se han presentado casos donde, al realizar un análisis hidráulico de la red, se ha encontrado que las mejores soluciones reportadas en la literatura incumplen ciertos criterios y en realidad no son factibles (Maier et al., 2003). Por estas razones se realizó un análisis hidráulico de la solución reportada por Zhou et al. (2015) y la obtenida en este trabajo. El análisis se realizó con el programa de cómputo EPANET 2.

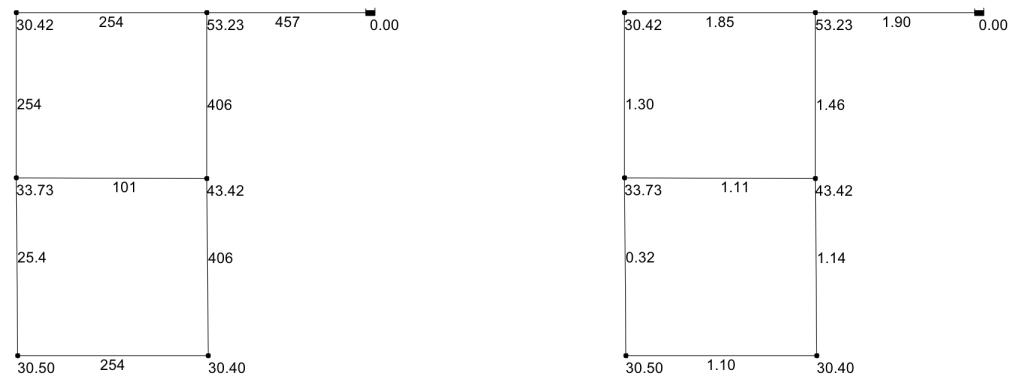


FIGURA 4.8: Análisis hidráulico de la solución obtenida por Zhou et al. (2015) para la red de dos circuitos, se muestran los diámetros en mm y las presiones en m (izq.), así como las velocidades de flujo en $m s^{-1}$ (der.).

Los resultados de dicho análisis para la solución de Zhou et al. (2015) se muestran en la figura 4.8. Se observa que en todos los casos se cumple con la carga de presión

mínima en los nodos, que debe ser mayor a 30 m , sin embargo en el nodo 2 se tiene una presión que sobrepasa por mucho el límite mínimo establecido con una presión de 53.23 m y se tienen nodos críticos en los que la presión apenas rebasa el límite teniendo una presión de 30.40 y 30.42 m . A pesar de ello, se puede considerar que la configuración no pone en riesgo el funcionamiento de la red.

Por otro lado se tiene el análisis de las velocidades de flujo permisibles que, aunque tiene bastante importancia, no es común encontrarlo en la literatura debido a que hay autores que no consideran este criterio. Para el diseño de redes de tuberías frecuentemente se toman como referencia los valores proporcionados por Richter (1962) que en general maneja un rango de velocidades permisibles entre 0.5 y 3.0 $m s^{-1}$ para este tipo de redes. En la figura 4.8 se puede observar que la tubería 8 de la red tiene una velocidad de 0.32 $m s^{-1}$, por lo tanto se puede decir que no cumple con el rango de velocidades permisibles indicado. Es necesario mencionar que el autor no tomó en cuenta el criterio de velocidad permisible en el planteamiento de su función objetivo y sus restricciones, lo que explica que la solución no cumpla con los valores permisibles utilizados en este trabajo.

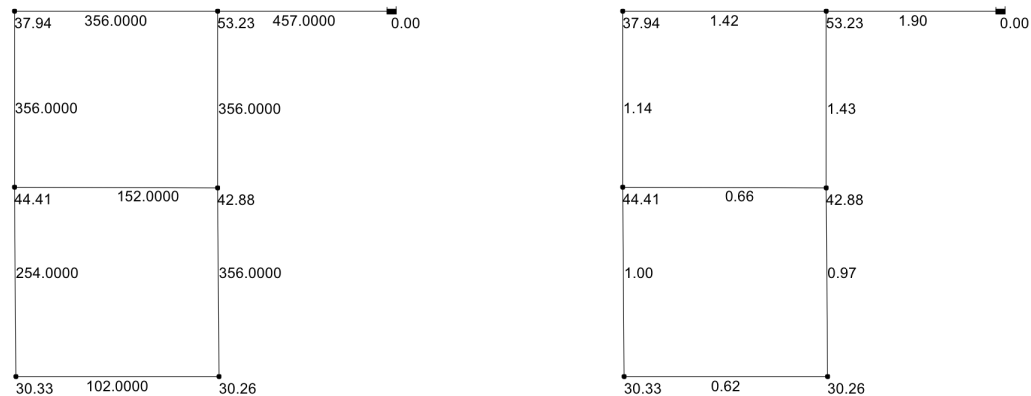


FIGURA 4.9: Análisis hidráulico de la mejor solución para la red de dos circuitos obtenida con Algoritmos Genéticos, se muestran los diámetros en mm y las presiones en m (izq.), así como las velocidades de flujo en $m s^{-1}$ (der.).

Los resultados del análisis hidráulico para la mejor solución encontrada por GA se muestran en la figura 4.9. Se observa que en todos los casos se cumple con la carga de presión mínima en los nodos que debe ser mayor a 30 m . En este caso en el nodo

2 se tiene una presión que sobrepasa el límite mínimo establecido con una presión de 53.23 m y también se tienen nodos críticos en los que la presión apenas rebasa el límite teniendo una presión de 30.26 y 30.33 m . Estrictamente estos valores son mayores al mínimo requerido y se podría considerar que la configuración no pone en riesgo el funcionamiento de la red. En cuanto a las velocidades de flujo permisibles se puede observar que todos los valores obtenidos del análisis hidráulico están dentro de los límites propuestos por Richter (1962), lo que confirma que la solución mínima obtenida en este trabajo es factible.

Al analizar una por una las tuberías de ambas soluciones con la norma mexicana NMX-O-177-SCFI-2011 (2011), se observa que se cumple sin problemas la condición de velocidad máxima de flujo, sin embargo, la norma recomienda que la velocidad mínima sea mayor a 1 $m s^{-1}$ lo cual no se cumple en ambas soluciones.

Es importante aclarar que la metaheurística GA utilizada es capaz de encontrar las soluciones de menor costo reportadas en el cuadro 4.7 cuando se proporciona holgura suficiente a los límites de velocidad de flujo permisibles en las restricciones de la función objetivo. Sin embargo, se decidió reportar la mejor solución obtenida que cumple con las condiciones de velocidad permisible.

4.2.2. Red de 3×3

Se realizó el dimensionamiento de la red de 3×3 para diferentes trazos obtenidos previamente en la literatura. Los diámetros de la solución subóptima encontrada por GA se muestra en el cuadro 4.8, esta solución se obtuvo usando los parámetros especificados en el cuadro 3.1.

Las soluciones obtenidas para el dimensionamiento de esta red usando GA se presentan en el cuadro 4.9 donde se muestran los valores mínimos obtenidos en cada una de las 20 ejecuciones, su diferencia en porcentaje con respecto a la solución mínima y las generaciones requeridas para encontrar dicha solución. Se puede observar que la solución subóptima se obtuvo en 1 de las 20 ejecuciones (5 % de las veces) y se requirieron de 108 generaciones del algoritmo para encontrarla. La solución mínima obtenida tiene un valor promedio de 62,010 con una desviación estándar de 2,409.39 los cuales son valores que se alejan mucho de las soluciones reportadas en la literatura, por lo tanto se puede decir que el algoritmo tiene poca robustez para obtener soluciones a este

CUADRO 4.8: Diámetros de tuberías en *mm* de la solución obtenida con Algoritmos Genéticos para la red de 3×3 , comparados con los reportados en la literatura.

Tubería	Afshar y Jabbari (2008)	Saleh y Tanyimboh (2013)	Este trabajo
1	100	100	80
2	120	100	120
3	120	120	120
4	100	100	140
5	120	140	140
6	140	140	160
7	140	140	180
8	140	140	180

problema. En este caso el algoritmo se configuró con una población de 200 individuos y 500 generaciones, por lo tanto se hicieron 100,000 evaluaciones de la función objetivo para obtener esta solución.

La convergencia de GA se muestra en la figura 4.10 y se puede observar que la solución mínima se mantiene durante un periodo largo, por lo tanto un valor de 500 generaciones resulta suficiente. Los problemas de dimensionamiento, como los planteados en este trabajo, no requieren de un número mayor de generaciones para encontrar soluciones aceptables, ya que el espacio de búsqueda es menor que el de los problemas de trazo.

La mejor solución reportada en la literatura es la obtenida por Saleh y Tanyimboh (2013), que se muestra la figura 4.11 (*izq.*). Cabe mencionar que en el trabajo original esta red se obtuvo mediante la solución simultánea de los problemas de trazo y el dimensionamiento. En el presente trabajo se utilizó el trazo de esta solución para obtener los diámetros de las tuberías usando GA. Una vez dimensionada la red se usaron los precios unitarios del cuadro 3.7 para obtener el costo total de la red. En el cuadro 4.10 se muestra el costo de la red de 3×3 comparado con las mejores soluciones reportadas en la literatura.

Para verificar que la solución obtenida cumpla con las condiciones hidráulicas de operación recomendadas, se realizó un análisis de la red usando el programa de cómputo EPANET 2, los valores obtenidos de presión y velocidad de flujo de la red obtenida

CUADRO 4.9: Soluciones para el dimensionamiento de la red de 3×3 obtenidas con Algoritmos Genéticos.

Ejecución	Mínimo [\$]	Diferencia [%]	Generaciones requeridas
1	65,000	11.49	442
2	61,200	4.97	109
3	64,200	10.12	223
4	65,000	11.49	239
5	59,200	1.54	219
6	62,000	6.35	135
7	59,200	1.54	327
8	60,200	3.26	481
9	62,200	6.69	381
10	61,300	5.15	100
11	65,000	11.49	217
12	65,000	11.49	86
13	59,200	1.54	286
14	61,000	4.63	325
15	59,200	1.54	178
16	61,000	4.63	323
17	65,000	11.49	144
18	65,000	11.49	147
19	58,300	0.00	108
20	62,000	6.35	243
Promedio:	62,010	6.36	236
Desv. Est.:	2,409.39	4.13	115.64

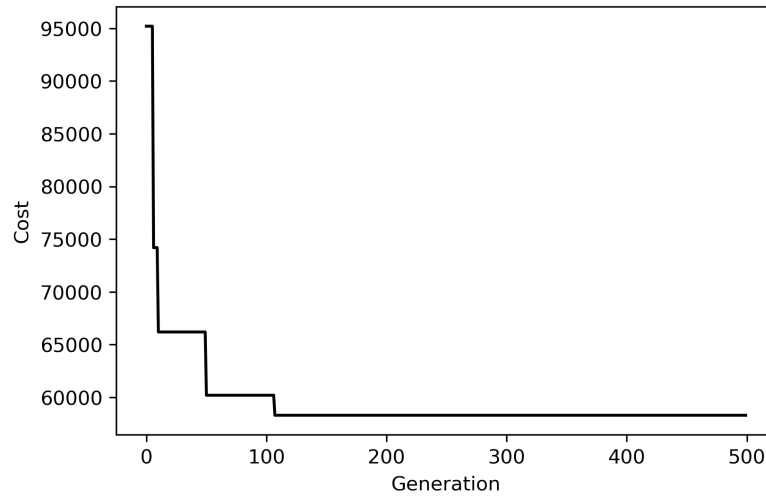


FIGURA 4.10: Convergencia de las soluciones obtenidas con Algoritmos Genéticos para el dimensionamiento de la red de 3×3 .

CUADRO 4.10: Comparación del costo de la solución obtenida con las reportadas en la literatura para el dimensionamiento de la red de 3×3 . Las soluciones se obtuvieron con Algoritmos Genéticos (GA) y con la técnica *Non-dominated Sorting Genetic Algorithm* (NSGA-II).

Solución	Método	Costo [\$]
Afshar y Jabbari (2008)	GA	39,400
Saleh y Tanyimboh (2013)	NSGA-II	38,600
Obtenida en este trabajo	GA	58,300

por Saleh y Tanyimboh (2013) se muestran en la figura 4.11.

Las dimensiones de esta solución generan una carga de presión que cumple con la mínima requerida en los nodos (30 m), aunque en los nodos 1 y 2 se tienen valores de presión críticos que apenas sobrepasan el mínimo requerido con 30.22 y 30.07 m respectivamente. Por otro lado las velocidades de flujo no cumplen con el valor máximo permisible recomendado por Richter (1962), que es de 3 m s^{-1} ; ya que en las tuberías 7 y 8 se tiene una velocidad de 3.9 m s^{-1} . Además si se utiliza la norma mexicana NMX-O-177-SCFI-2011 (2011) las tuberías 2, 4 y 6 tampoco cumplen con la recomendación de velocidad máxima, que es de 2 m s^{-1} para estos diámetros, ya que sus velocidades de flujo son de 2.55 , 2.55 y 2.6 m s^{-1} respectivamente.

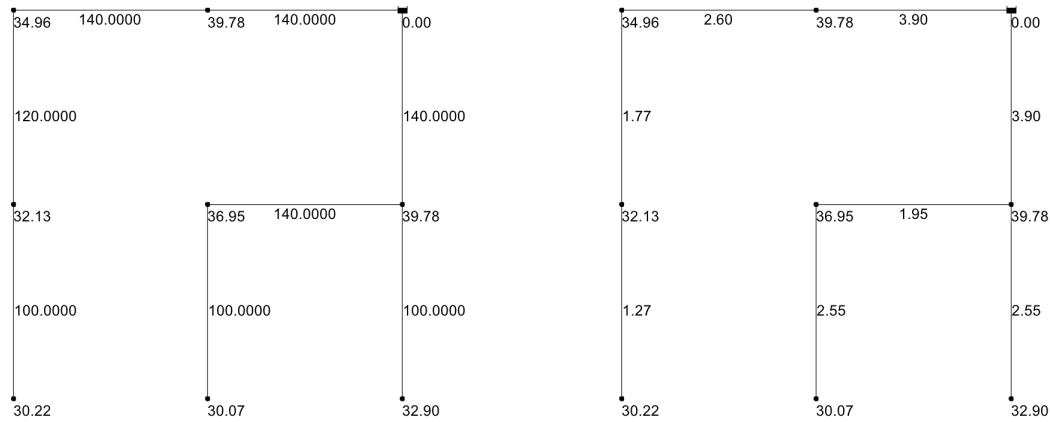


FIGURA 4.11: Análisis hidráulico de la solución obtenida por Saleh y Tan-yimboh (2013) para la red de 3×3 , se muestran los diámetros en mm y las presiones en m (izq.), así como las velocidades de flujo en m/s (der.).

Los resultados del análisis hidráulico para la mejor solución encontrada por GA se muestran en la figura 4.12. Se observa que en todos los casos se cumple con la carga de presión mínima en los nodos que debe ser mayor a $30 m$ para cumplir el caudal de demanda, de hecho los valores exceden la presión mínima requerida. Para evitar que las presiones resultantes se sobrepasen exageradamente a la mínima requerida se pueden agregar restricciones para la presión máxima en los nodos, sin embargo este tipo de análisis es poco frecuente de encontrar en la literatura ya que generalmente no se tienen problemas por esta condición. En cuanto a las velocidades de flujo permisibles se puede observar que todos los valores obtenidos del análisis hidráulico están dentro de los límites propuestos por Richter (1962), lo que confirma que la solución mínima obtenida en este trabajo es factible. Finalmente, usando la norma NMX-O-177-SCFI-2011 (2011) se puede ver que las tuberías 7 y 8 tienen una velocidad de flujo de $2.36 m s^{-1}$ y sobrepasan el valor máximo que le corresponde a sus diámetros que es de $2 m s^{-1}$.

Al igual que en el problema anterior, la metaheurística GA es capaz de encontrar las soluciones de menor costo reportadas en el cuadro 4.10 cuando se proporciona holgura suficiente a los límites de velocidad de flujo permisibles en las restricciones de la función objetivo. Sin embargo, se decidió reportar la mejor solución obtenida que cumple con las condiciones de velocidad permisible.

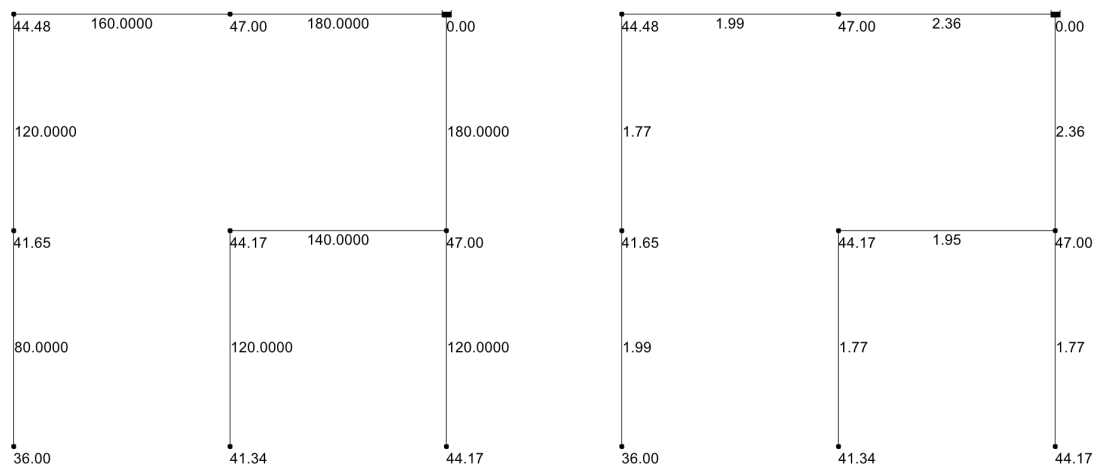


FIGURA 4.12: Análisis hidráulico de la mejor solución para la red de 3×3 obtenida con Algoritmos Genéticos, se muestran los diámetros en mm y las presiones en m (izq.), así como las velocidades de flujo en $m s^{-1}$ (der.).

Capítulo 5

Conclusiones

Se realizó el diseño de redes de riego planteando un problema para la fase de trazo y otro para el dimensionamiento. Se solucionaron ambos problemas de manera satisfactoria usando las metaheurísticas Algoritmos Genéticos (GA por sus siglas en inglés de *Genetic Algorithms*) y Optimización con Colonia de Hormigas (ACO por sus siglas en inglés de *Ant Colony Optimization*) bajo un enfoque secuencial. Los algoritmos se evaluaron mediante la solución de redes de prueba usadas en la literatura y la mayoría de las soluciones obtenidas fueron iguales o muy cercanas a las mejores reportadas hasta el momento.

En el problema de dimensionamiento, las mejores soluciones se obtuvieron con GA. Aunque las soluciones óptimas para el problema de trazo se obtuvieron con ambos algoritmos, ACO necesitó un número menor de evaluaciones comparado con las requeridas por GA. Esto refuerza la idea de que ACO es una técnica adecuada y eficiente para optimizar el diseño de redes de riego, especialmente para el problema de trazo. Aunque GA ya se ha establecido como una técnica confiable para solucionar éstos problemas, tuvo dificultad para encontrar soluciones a instancias grandes del problema de trazo en un tiempo aceptable.

Las soluciones obtenidas corroboran la hipótesis de que es necesario hacer el trazo y dimensionamiento de las redes de riego de manera simultánea para obtener soluciones aceptables, por ello es necesario hacer el planteamiento de un problema multiobjetivo y obtener las soluciones con las heurísticas adecuadas.

Apéndice A

Código fuente

El código fuente de los algoritmos se encuentra publicado en el siguiente repositorio:

`https://github.com/ecoslacker/networks\_design`

El código no se encuentra disponible en este documento ya que incrementa demasiado la extensión del mismo.

Bibliografía

- Afshar, A., F. Massoumi, A. Afshar y M. A. Mariño (2015). «State of the art review of ant colony optimization applications in water resource management». En: *Water Resources Management* 29, págs. 3891-3904. DOI: 10.1007/s11269-015-1016-9. URL: <http://link.springer.com/10.1007/s11269-015-1016-9>.
- Afshar, M. H. (2006). «Application of a max-min ant system to joint layout and size optimization of pipe networks». En: *Engineering Optimization* 38.3, págs. 299-317. ISSN: 0305215X. DOI: 10.1080/03052150500476357.
- Afshar, M. H. y E. Jabbari (2008). «Simultaneous layout and pipe size optimization of pipe networks using genetic algorithm». En: *The Arabian Journal for Science and Engineering* 33.2B, págs. 391-409. ISSN: 13198025. DOI: 10.1061/(ASCE)1076-0342(2005)11:4(221).
- Afshar, M. H. y M. A. Mariño (2006). «Application of an ant algorithm for layout optimization of tree networks». En: *Engineering Optimization* 38.3, págs. 353-369. ISSN: 0305-215X. DOI: 10.1080/03052150600557783.
- Alperovits, E. y U. Shamir (1977). «Design of Optimal Water Distribution Systems». En: *Water Resources Research* 13.6, págs. 59-67.
- Ángeles Montiel, V. (2002). *Redes abiertas de tuberías para riego (Trazo, diseño, revisión y análisis)*. Chapingo, México: Universidad Autónoma Chapingo, págs. 1-149.
- Bi, W., G. C. Dandy y H. R. Maier (2015). «Improved genetic algorithm optimization of water distribution system design by incorporating domain knowledge». En: *Environmental Modelling & Software* 69, págs. 370-381. ISSN: 1364-8152. DOI: <http://dx.doi.org/10.1016/j.envsoft.2014.09.010>. URL: <http://www.sciencedirect.com/science/article/pii/S1364815214002631>.
- Boussaïd, I., J. Lepagnot y P. Siarry (2013). «A survey on optimization metaheuristics». En: *Information Sciences* 237, págs. 82-117. ISSN: 00200255. DOI: 10.1016/j.ins.2013.02.041.

- Brabazon, A., M. O'Neill y S. McGarraghy (2015). *Natural Computing Algorithms (Natural Computing Series)*. Ed. por G. Rozenberg, T. Bäck, A. Eiben, J. Kok y H. Spaink. Springer Berlin Heidelberg, pág. 554. ISBN: 3662436302. DOI: 10.1007/978-3-662-43631-8. URL: <https://dx.doi.org/10.1007/978-3-662-43631-8>.
- Coello Coello, C. A. (2001). «A short tutorial on evolutionary multiobjective optimization». En: *Evolutionary MultiCriterion Optimization* 1993.6, págs. 21-40. ISSN: 16113349. DOI: 10.1007/3-540-44719-9_2. arXiv: arXiv:1011.1669v3.
- Coello Coello, C. A., G. B. Lamont y D. a. V. Veldhuizen (2007). *Evolutionary algorithms for solving multi-objective problems*. Ed. por D. E. Goldberg y J. R. Koza. 2.^a ed., pág. 800. ISBN: 978-0-387-33254-3. DOI: 10.1007/978-0-387-36797-2.
- CONAGUA (2016). *Estadísticas del Agua en México, Edición 2016*. Inf. téc. México, D.F.: Comisión Nacional del Agua, pág. 275.
- Dandy, G. C., A. R. Simpson y L. J. Murphy (1996). «An improved genetic algorithm for pipe network optimization». En: *Water Resources Research* 32.2, págs. 449-458. DOI: 10.1002/wrcr.20175/abstract/PERMISSIONS.
- De los Cobos Silva, S. G., J. Goddard Close, M. Á. Gutiérrez Andrade y A. E. Martínez Licona (2010). *Búsqueda y exploración estocástica*. Universidad Autónoma Metropolitana, págs. 1-260. ISBN: 978-607-477-239-5.
- Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms [en Italiano]*. Milan: PhD thesis, Dipartimento di Elettronica, Politecnico di Milano.
- Dorigo, M., V. Maniezzo y A. Coloni (1996). «Ant system: optimization by a colony of cooperating agents». En: *IEEE Transactions on Systems, Man and Cybernetics-Part B* 26.1, págs. 29-41. ISSN: 1083-4419.
- Dorigo, M. y T. Stützle (2004). *Ant Colony Optimization*. Massachusetts, United States of America: Massachusetts Institute of Technology, págs. 1 -231. ISBN: 0262042193. DOI: 10.4249/scholarpedia.1461.
- Du, K.-L. y M. N. S. Swamy (2016). *Search and Optimization by Metaheuristics*. Springer International Publishing Switzerland, pág. 437. ISBN: 978-3-319-41191-0. DOI: 10.1007/978-3-319-41192-7. URL: <http://link.springer.com/10.1007/978-3-319-41192-7>.
- Eiben, A. E. y J. E. Smith (2015). *Introduction to evolutionary computing*. Ed. por G. Rozenberg, T. Bäck, A. E. Eiben, J. N. Kok y H. P. Spaink. Second Edi. Berlin, Heidelberg:

- Springer-Verlag, pág. 287. ISBN: 978-3-662-44874-8. DOI: 10.1007/978-3-662-44874-8.
- Eiger, G., U. Shamir y A. Ben-Tal (1994). «Optimal design of water distribution networks». En: *Water Resources Research* 30.9, págs. 2637-2646.
- Farmani, R., D. A. Savic y G. A. Walters (2005). «Evolutionary multi-objective optimization in water distribution network design». En: *Engineering Optimization* 37.2, págs. 167-183. DOI: 10.1080/03052150512331303436. URL: <http://dx.doi.org/10.1080/03052150512331303436>.
- Fister, I., X. S. Yang, J. Brest y D. Fister (2013). «A brief review of nature-inspired algorithms for optimization». En: *Elektrotehniski Vestnik/Electrotechnical Review* 80.3, págs. 116-122. ISSN: 00135852. arXiv: 1307.4186.
- Fogel, L., A. Owens y M. Walsh (1966). *Artificial Intelligence through Simulated Evolution*. New York: Wiley.
- Formiga, K. T. M., F. H. Chaudhry, P. B. Cheung y L. F. R. Reis (2003). «Optimal design of water distribution system by multiobjective evolutionary methods». En: *Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003, Faro, Portugal, April 8–11, 2003*. Ed. por C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele y K. Deb. Berlin, Heidelberg: Springer Berlin Heidelberg, págs. 677-691. ISBN: 978-3-540-36970-7. DOI: 10.1007/3-540-36970-8_48. URL: http://dx.doi.org/10.1007/3-540-36970-8_{_}48.
- García Domínguez, M. Y., I. Sánchez Cohen, G. García Herrera, L. Moreno Díaz, R. Trejo Calzada y M. A. Hernández Martínez (2010). «Evaluación de la eficiencia de riego en el módulo IV del Distrito de Riego 017 Comarca Lagunera, México». En: *Revista Chapingo Serie Zonas Áridas* 9, págs. 99-106.
- Geem, Z. W., T. G. Kim y J. H. Kim (2000). «Optimal layout of pipe networks using Harmony Search». En: *Proceedings of 4th International Conference on Hydro-Science and Engineering, Seoul, South Korea*.
- Gendreau, M. y J.-Y. Potvin (2005). «Metaheuristics in combinatorial optimization». En: *Annals of Operations Research* 140, págs. 189-213. ISSN: 0254-5330. DOI: 10.1007/s10479-005-3971-7.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 412 pp.

- Gonçalves, G. M., L. Gouveia y M. V. Pato (2014). «An improved decomposition-based heuristic to design a water distribution network for an irrigation system». En: *Annals of Operations Research* 219.1, págs. 141-167. ISSN: 1572-9338. DOI: 10.1007/s10479-011-1036-7. URL: <http://dx.doi.org/10.1007/s10479-011-1036-7>.
- Goulter, I. C., B. M. Lussier y D. R. Morgan (1986). «Implications of head loss path choice in the optimization of water distribution networks». En: *Water Resources Research* 22.5, págs. 819-822. ISSN: 19447973. DOI: 10.1029/WR022i005p00819.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
- Keedwell, E. y S.-T. Khu (2006). «A novel evolutionary meta-heuristic for the multi-objective optimization of real-world water distribution networks». En: *Engineering Optimization* 38.3, págs. 319-336. DOI: 10.1080/030521505000476308.
- Kessler, A. y U. Shamir (1989). «Analysis of the linear programming gradient method for optimal design of water supply networks». En: *Water Resources Research* 25.7, págs. 1469-1480. DOI: 0043-1397/89/89WR-004285.
- Laybe, Y., M. A. Olson, A. Galand y N. Tsiourtis (1988). *Design and optimization of irrigation distribution networks*. Rome: FAO, págs. 1-247. ISBN: 92-5-102666-1.
- Maier, H. R., A. R. Simpson, W. K. Foong, K. Y. Phang, H. Y. Seah y C. L. Tan (2001). «Ant colony optimization for the design of water distribution systems». En: *World Water Congress 2001*, págs. 1-10. DOI: 10.1061/(ASCE)0733-9496(2003)129:3(200). URL: <http://ascelibrary.org/doi/10.1061/{\%}28ASCE{\%}290733-9496{\%}282003{\%}29129{\%}3A3{\%}28200{\%}29>.
- Maier, H. R., A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah y C. L. Tan (2003). «Ant colony optimization for design of water distribution systems». En: *Journal of Water Resources Planning and Management* 129.3, págs. 200-209. DOI: 10.1061/(ASCE)0733-9496(2003)129:3(200). URL: [http://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-9496\(2003\)129:3\(200\)](http://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9496(2003)129:3(200)).
- Mtolera, I., L. Haibin, L. Ye, B. F. Su, D. Xue y M. Yi (2014). «Optimization of tree pipe networks layout and size, using particle swarm optimization». En: *WSEAS Transactions on Computers* 13, págs. 219-230. ISSN: 22242872.
- NMX-O-177-SCFI-2011 (2011). *Lineamientos generales para proyectos de sistemas de riego presurizado*. Diario Oficial de la Federación: 18/08/2011. México, D.F.

- Ortega-Gaucin, D., E. Mejía Sáenz, E. Palacios Vélez, L. Rendón Pimentel y A. Exebio García (2009). «Modelo de optimización de recursos para un Distrito de Riego». En: *Terra Latinoamericana* 27.3, págs. 219-226. URL: <http://www.redalyc.org/articulo.oa?id=57313037006>.
- Ostfeld, A. (2011). «Ant Colony Optimization for Water Resources Systems Analysis-Review and Challenges». En: *Ant Colony Optimization-Methods and Applications*. InTech, págs. 147-158. ISBN: 978-953-307-157-2. DOI: 10.5772/13522. URL: <http://www.intechopen.com/books/ant-colony-optimization-methods-and-applications/ant-colony-optimization-for-water-resources-systems-analysis-review-and-challenges>.
- Osyczka, A. (1985). *Multicriteria optimization for engineering design*. In J. S. Gero, editor, *Design Optimization*. Academic Press, págs. 193-227. ISBN: 978-607-477-239-5.
- Palacios Vélez, E. (2004). «La eficiencia en el uso del agua en los distritos de riego». En: *Agua para la Agricultura*, págs. 1-9.
- Peña Peña, E. (2007). «Eficiencias del uso del agua en distritos de riego en México». En: *Gaceta del IMTA*, págs. 1-5.
- Pedroza González, E. (2014). *Manejo y distribución del agua en distritos de riego: breve introducción didáctica*. Ed. por E. Pedroza González y G. A. Hinojosa Cuéllar. Jiutepec, Mor.: Instituto Mexicano de Tecnología del Agua, págs. 1-145. ISBN: 9786077563969.
- Ponce Pacheco, M. A. (2013). «Diseño de redes de riego mediante algoritmos de evolución diferencial y colonia artificial de abejas». Universidad Autónoma Chapingo, págs. 1-147.
- Potter, M. C., D. C. Wiggert, B. Ramadan y T. I.-P. Shih (2011). *Mechanics of fluids*. 4.^a ed. USA: CENGAGE Learning, págs. 1-795. ISBN: 9788578110796. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.
- Rechenberg, I. (1965). *Cybernetic Solution Path of an Experimental Problem*. Farnborough: Royal Aircraft Establishment.
- (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag: Stuttgart.
- Richter, H. (1962). *Rohrhydraulik*. 4.^a ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg GmbH, págs. 1-377. ISBN: 9783662005668. DOI: 10.1007/978-3-662-00566-8.

- Román López, A., M. A. Inzunza Ibarra, E. A. Catalán Valencia y M. M. Villa Castonera (2013). «Indicadores de eficiencia del agua de riego aplicada en dos sistemas de riego por aspersión». En: *Agrofaz* 13.2, págs. 9-14.
- Saldarriaga, J. (2007). *Hidráulica de tuberías, abastecimiento de agua, redes, riegos*. Alfaomega, págs. 1-692. ISBN: 978-958-682-680-8.
- Saleh, S. H. A. y T. T. Tanyimboh (2013). «Coupled topology and pipe size optimization of water distribution systems». En: *Water Resour Manage* 27, págs. 4795-4814. DOI: 10.1007/s11269-013-0439-4.
- Simpson, A. R., G. C. Dandy y L. J. Murphy (1994). «Genetic algorithms compared to other techniques for pipe optimization». En: *J Water Resour Plann Manage* 120.4, págs. 423-443.
- Storn, R. M. y K. V. Price (1997). «Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces». En: *Journal of Global Optimization* 11, págs. 341-359.
- Stützle, T. y H. H. Hoos (2000). «MAX-MIN Ant System». En: *Future Generation Computer Systems* 16, págs. 889-914. ISSN: 0167739X. DOI: 10.1016/S0167-739X(00)00043-1.
- Todini, E. y S. Pilati (1987). «A Gradient Algorithm for the Analysis of Pipe Networks». En: *Computer Applications in Water Supply. Volume 1 System Analysis and Simulation*. Ed. por B. L. P. Coulbeck y O. L. P. Chun-Hou. John Wiley & Sons Inc. Cap. Section 1.
- Walters, G. A. y T. Lohbeck (1993). «Optimal layout of tree networks using genetic algorithms». En: *Engineering Optimization* 22.1, págs. 27-48. ISSN: 0305-215X. DOI: 10.1080/03052159308941324. URL: <http://www.tandfonline.com/doi/abs/10.1080/03052159308941324>.
- Walters, G. A. y D. K. Smith (1995). «Evolutionary design algorithm for optimal layout of tree networks». En: *Engineering Optimization* 24.4, págs. 261-281. ISSN: 0305-215X. DOI: 10.1080/03052159508941193. URL: <http://www.tandfonline.com/doi/abs/10.1080/03052159508941193>.
- Yang, X.-S. (2014). *Nature-inspired optimization algorithms*. First edit. London: Elsevier Inc., pág. 264. ISBN: 978-0-12-416743-8.

- Yates, D. F., A. B. Templeman y T. B. Boffey (1984). «The computational complexity of the problem of determining least capital cost designs for water supply networks». En: *Engineering Optimization* 2.7, págs. 142-155.
- Zheng, F., A. R. Simpson y A. C. Zecchin (2013). «A decomposition and multistage optimization approach applied to the optimization of water distribution systems with multiple supply sources». En: *Water Resources Research* 49.1, págs. 380-399. ISSN: 00431397. DOI: 10.1029/2012WR013160.
- Zhou, X., D. Y. Gao y A. R. Simpson (2015). «Optimal design of water distribution networks by a discrete state transition algorithm». En: *Engineering Optimization*, págs. 1-26. ISSN: 0305-215X. DOI: 10.1080/0305215X.2015.1025775. arXiv: 1304.7622. URL: <http://dx.doi.org/10.1080/0305215X.2015.1025775>.

Escudo del apellido Jiménez tomado de *Origen, Historia y Heráldica de los Apellidos Españoles e Hispanoamericanos: Apellido Jiménez* por el Instituto de Historia y Heráldica Familiar.