

TRABAJO ESPECIAL DE GRADO

**OPTIMIZACIÓN DEL COSTO DE REDES DE ACUEDUCTOS
MEDIANTE UN ALGORITMO DE BÚSQUEDA HEURÍSTICO
ADAPTATIVO MULTITETAPA**

**Presentado ante la Ilustre
Universidad Central de Venezuela
Por los Br. Guzmán C. Jorge D.,
Ramírez G. Marlon L.
para optar al título de
Ingeniero Mecánico**

Caracas, 2023

TRABAJO ESPECIAL DE GRADO

OPTIMIZACIÓN DEL COSTO DE REDES DE ACUEDUCTOS MEDIANTE UN ALGORITMO DE BÚSQUEDA HEURÍSTICO ADAPTATIVO MULTITETAPA

Tutor: Prof. Eugenio Sansone

**Presentado ante la Ilustre
Universidad Central de Venezuela
Por los Br. Guzmán C. Jorge D.,
Ramírez G. Marlon L.
para optar al título de
Ingeniero Mecánico**

Caracas, 2023



ACTA

Los abajo firmantes, Miembros del Jurado Examinador designado por el Consejo de Escuela de Ingeniería Mecánica para evaluar el Trabajo Especial de Grado presentado por los Bachilleres,

Jorge D, Guzmán C
Ci. 25525478

y

Marlon L, Ramírez G
Ci. 25607786

Titulado

“Optimización del Costo de Redes de Acueductos mediante un Algoritmo de Búsqueda Heurística, Adaptativo, Multietapa”

Consideran que el mismo cumple con los requisitos exigidos por el Plan de Estudios conducente al Título de Ingenieros Mecánicos.

Acta se levanta en la ciudad de Caracas, a los 30 ías del mes de Octubre del año dos mil veintitrés.

Prof. Cristóbal Quevedo
Jurado

Prof. Rodolfo Berrios
Jurado

Prof. Eugenio Sansone
Tutor





UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA MECANICA
DEPARTAMENTO DE ENERGÉTICA

Caracas, 30 de Octubre 2023

Profa. María Eugenia Korody

Jefe de la División de Control de Estudios
Coordinación Académica
Facultad de Ingeniería-UCV
Presente.-

Quienes suscriben, Miembros del Jurado Examinador designado por el Consejo de la Escuela de Ingeniería Mecánica, nos dirigimos a usted con la finalidad de informarle que **hemos decidido otorgarle a los Bachilleres:**

Jorge D., Guzmán C.
Ci. 25.525.478

y

Marlon L., Ramírez G.
Ci. 25.607.786

MENCIÓN HONORÍFICA
TRABAJO ESPECIAL DE GRADO

Por la excelencia demostrada en la realización del Trabajo Especial de Grado, titulado

“Optimización del Costo de Redes de Acueductos mediante un Algoritmo de Búsqueda Heurística, Adaptativo, Multietapa”

Prof. Cristóbal Quevedo
Jurado

Prof. Rodolfo Berrios
Jurado

Prof. Eugenio Sansone
Tutor



DEDICATORIA

A NUESTROS FAMILIARES,
AMIGOS Y ALLEGADOS.

AGRADECIMIENTOS

JORGE GUZMÁN

Quiero expresar mi más profundo agradecimiento a Dios, por darme la vida, la salud y la oportunidad de realizar este trabajo. A mis padres, por su amor incondicional, su apoyo constante y su ejemplo de esfuerzo y perseverancia. A mi hermano, por ser mi compañero de aventuras y mi confidente. A mis abuelos y abuelas, por sus sabios consejos y sus oraciones. A toda mi familia, por estar siempre presentes y brindarme su afecto y su ayuda.

También quiero agradecer a mis amigos, por compartir conmigo momentos de alegría, de tristeza, de diversión y de aprendizaje. Por ser mi fuente de motivación y de inspiración. Por creer en mí y en mis capacidades. A la profesora Marylin Ramos y cada uno de los miembros del DIOC por los agradables recuerdos y sus valiosos conocimientos.

Asimismo, quiero reconocer la influencia positiva que han tenido en mi vida mis ídolos, como Albert Einstein, quien me inspiró a estudiar con esta frase: “Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”. Y Jim Carrey, de quien aprendí que “Madurar no significa olvidar que alguna vez fuiste niño”. Ellos me enseñaron a valorar el conocimiento y a mantener vivo el espíritu infantil.

Finalmente, quiero expresar mi sincera gratitud a mi tutor académico, el profesor Eugenio Sansone, por su orientación, su paciencia y su dedicación. Por sus valiosas sugerencias y sus críticas constructivas. Por confiar en mi proyecto y en mi potencial. A él le debo gran parte del éxito de este trabajo. Igualmente, quiero agradecer a la Universidad Central de Venezuela, por brindarme una educación de calidad, por facilitarme los recursos necesarios para realizar esta investigación, y por formarme como un profesional competente y comprometido con la sociedad.

A todos ellos, mi más sincero reconocimiento y mi eterno agradecimiento.

Guzmán C. Jorge D.

AGRADECIMIENTOS

MARLON RAMÍREZ

En estas paginas plasmadas, existe esfuerzo, dedicación y saber, pero detrás de cada logro, siempre hay alguien que ayudó a crecer.

A la Universidad Central de Venezuela. En tus aulas aprendí y crecí como persona, me enseñaste a pensar, a investigar y a crear, gracias a tu amor y dedicación por ayudarme a crecer como profesional.

Al profesor Eugenio Sansone, por tu labor, por tus consejos y orientación. Confiaste en el proyecto, en nuestra dedicación y esfuerzo, gracias a ti este trabajo ha sido un éxito.

A cada uno de los profesores, con los que tuve el placer de crecer profesionalmente, que me brindaron las herramientas para triunfar, les agradezco por la formación que me ha permitido llegar hasta acá.

A mis padres, que en cada momento, me apoyaron y guiaron con amor, los aprecio por ser el pilar que me impulsó a seguir con fervor.

A mis amigos, que estuvieron conmigo en cada momento, apoyaron en cada alegría y cada dolor, agradezco por su compañía y por hacer más fácil esta labor.

A los autores, cuyas ideas me inspiraron y guiaron en esta investigación, les agradezco por su legado que hoy se refleja con honor.

A ti, amigo lector, el tiempo es un regalo precioso, y tu lo has compartido conmigo, gracias por leer este escrito, porque así me haces sentir vivo.

Al país que me vio nacer y al mundo entero, sin excepción, les agradezco por las oportunidades que me permitieron lograr esta misión.

Ramírez G. Marlon L.

Guzmán C., Jorge D. y Ramírez G., Marlon L.

OPTIMIZACIÓN DEL COSTO DE REDES DE ACUEDUCTOS MEDIANTE UN ALGORITMO DE BUSQUEDA HEURÍSTICO ADAPTATIVO MULTIETAPA

**Tutor: Prof. Eugenio Sansone. Tesis. Caracas, U.C.V. Facultad de Ingeniería
Escuela de Ingeniería Mecánica. Año 2023, 99 pág.**

Palabras claves: Redes de distribución de agua; Diseño de costo optimo; Evolución diferencial; Programación genética adaptativa; Algoritmo genético.

RESUMEN. El diseño de redes de distribución de agua (RDA) es un problema complejo que requiere encontrar el diámetro óptimo de las tuberías para cumplir con los requisitos hidráulicos y minimizar el costo. Los algoritmos evolutivos son capaces de encontrar la mejor solución posible o al menos una solución aceptable a problemas con un espacio de soluciones extenso y complejo, por lo que son candidatos excelentes para esta clase de dilemas, especialmente el algoritmo de evolución diferencial.

El propósito de este trabajo es presentar un algoritmo de búsqueda que particione la red completa en un conjunto de subredes que se resolverán independientemente, disminuyendo el número de variables y el esfuerzo computacional. Luego, aplicar un algoritmo de búsqueda evolutiva diferencial (ED) que permita hallar los diámetros óptimos de las tuberías en cada subred. Entonces las soluciones óptimas junto con las tuberías de partición se usan para crear una tabla de selección con un menor rango de búsqueda de la solución que empleará el algoritmo ED para optimizar el costo de la RDA original.

Los resultados obtenidos indican que la técnica de evolución diferencial fue la más eficiente en términos de costos en todos los casos estudiados. Aunque esta técnica es altamente efectiva para optimizar redes de acueductos, se requiere una investigación más profunda para mejorar los tiempos y aplicarla en redes más grandes.

ÍNDICE GENERAL

CONSTANCIA DE APROBACIÓN.....	III
MENCIÓN HONORÍFICA.....	IV
DEDICATORIA.....	V
AGRADECIMIENTOS.....	VII
RESUMEN.....	VIII
ÍNDICE GENERAL.....	IX
LISTA DE TABLAS.....	XII
LISTA DE FIGURAS.....	XIII
INTRODUCCIÓN.....	1
CAPÍTULO I.....	3
1. FUNDAMENTOS DE LA INVESTIGACIÓN.....	3
1.1 Planteamiento del problema.....	3
1.2 Objetivos.....	6
1.2.1 Objetivo General.....	6
1.2.2 Alcances.....	6
1.3 Antecedentes.....	7
CAPÍTULO II.....	12
2. CONCEPTOS Y ECUACIONES BÁSICAS.....	12
2.1 Variables y conceptos de naturaleza hidráulica.....	12
2.1.1 Tubería.....	12
2.1.2 Redes de Distribución de Agua (RDA).....	12
2.1.3 Nodos fuente o reservorios.....	12
2.1.4 Nodos de consumo.....	12
2.2 Ecuaciones generales que determinan el estado estacionario de una red.....	13
2.2.1 Carga piezométrica.....	14
2.2.2 Carga total.....	14
2.2.3 Carga de velocidad o carga cinética.....	14
2.2.4 Pérdidas de carga neta por fricción.....	15
2.2.5 Ecuación de la continuidad.....	16
2.2.6 Balance de carga neta alrededor de un lazo interno.....	17
2.2.7 Balance de carga neta en lazos cerrados.....	17
2.3 Variables y conceptos en la optimización de redes.....	17
2.3.1 Optimización.....	18

2.3.2 Optimización local y global.....	18
2.3.3 Grafo.....	19
2.3.4 Tuberías de partición.....	19
2.3.5 Problema del viajante en la teoría de grafos.....	19
2.3.6 Algoritmo.....	20
2.3.7 Algoritmo de Dijkstra.....	20
2.3.7.1 Esquema del camino más corto.....	20
2.3.8 Algoritmos Evolutivos (AE).....	21
2.3.9 Programación Lineal (PL).....	22
2.3.10 Programación no Lineal (PNL).....	22
2.3.11 Algoritmo Heurístico.....	22
2.3.12 Elitismo.....	22
2.4 Función Objetivo.....	23
2.4.1 Restricciones hidráulicas.....	24
2.4.2 Restricciones Técnico-Normativas.....	24
2.5 Algoritmo Genético (AG).....	26
2.5.1 Pseudocódigo AG.....	26
2.6 Algoritmo de Evolución Diferencial (ED).....	27
2.6.1 Pseudocódigo ED.....	28
2.6.2 Variables de inicio.....	29
2.6.3 Conjunto de Soluciones.....	32
2.6.4 Variación.....	35
2.6.5 Combinación.....	37
2.6.6 Selección.....	38
2.7 Manejo de restricciones.....	38
2.7.1 Tipo pared de ladrillo.....	39
2.7.2 Reinicialización aleatoria.....	39
2.7.3 Pivote.....	39
2.8 Estrategia para evitar reevaluar variables.....	39
2.9 Descomposición de RDA.....	40
CAPÍTULO III.....	43
3. METODOLOGÍA IMPLEMENTADA.....	43
3.1 Diseño de la investigación.....	43
3.1.1 Generación de números aleatorios.....	43

3.1.2 Datos de entrada.....	44
3.2 Casos de estudio.....	44
3.2.1 Primer caso.....	44
3.2.2 Segundo caso.....	45
3.2.3 Tercer caso. Red de Pescara, Italia.....	47
3.2.4 Cuarto caso. Red de Balerna, España.....	51
3.3 Procedimiento para optimizar RDA.....	52
3.3.1 Descomposición de RDA.....	52
3.3.2 Esquema del algoritmo de Dijkstra.....	53
3.3.3 Descomposición mediante pendiente hidráulica.....	54
3.3.4 Esquema del programa de partición de redes.....	56
3.4 Estrategia de exploración.....	56
3.5 Detección de redundancia.....	56
3.6 Algoritmo ED de primera etapa.....	57
3.7 Algoritmo ED de segunda etapa.....	58
3.8 Procedimiento para optimizar RDA con Algoritmo Genético.....	59
3.9 Análisis de datos.....	59
3.9.1 Porcentaje de eficiencia.....	60
CAPÍTULO IV.....	61
4. RESULTADOS.....	61
4.1 Primer caso de estudio.....	61
4.2 Segundo caso de estudio.....	65
4.3 Tercer caso de estudio.....	70
4.4 Cuarto caso de estudio.....	73
CAPITULO V.....	75
5. CONCLUSIONES Y RECOMENDACIONES.....	75
5.1 Conclusiones.....	75
5.2 Recomendaciones.....	76
REFERENCIAS BIBLIOGRÁFICAS.....	77
GLOSARIO.....	82
ANEXOS.....	84

LISTA DE TABLAS

Tabla 1. Algoritmos usados para optimizar RDA.....	5
Tabla 2. AE usados en la optimización de RDA.....	8
Tabla 3. Algoritmos ED en la optimización de RDA.....	10
Tabla 4. Efecto del número de iteraciones en optimización del costo.....	30
Tabla 5. Estudio de los factores F y CR.....	31
Tabla 6. Datos del primer caso de estudio.....	45
Tabla 7. Lista de diámetros comerciales de Kadu, Gupta y Bhave (2008).....	45
Tabla 8. Datos para el segundo caso de estudio.....	46
Tabla 9. Lista de diámetros comerciales para la red de Pescara, Italia.....	48
Tabla 10. Datos para la red de Pescara. Italia.....	49
Tabla 11. Cálculo de la pendiente de hidráulica.....	55
Tabla 12. Mejor ajuste con ED, primer caso.....	62
Tabla 13. Mejor ajuste con AG, primer caso.....	63
Tabla 14. Tiempo y costo de optimización según la técnica utilizada, primer caso.....	63
Tabla 15. Diámetros y presiones de la red optimizada, primer caso.....	65
Tabla 16. Mejor ajuste con ED, segundo caso.....	66
Tabla 17. Mejor ajuste con AG, segundo caso.....	67
Tabla 18. Tiempo y costo de optimización según la técnica utilizada, segundo caso.....	67
Tabla 19. Diámetros y presiones de la red optimizada, segundo caso.....	69
Tabla 20. Mejor ajuste con ED, tercer caso.....	70
Tabla 21. Tiempo y costo de optimización según la técnica utilizada, tercer caso.....	71
Tabla 22. Pendiente hidráulica en nodos 275 y 276.....	74

LISTA DE FIGURAS

Figura 1. Línea de energía hidráulica.....	15
Figura 2. Vista tridimensional de la función objetivo.....	18
Figura 3. Remoción del conjunto de partición.....	19
Figura 4. Selección elitista.....	23
Figura 5. Conjunto de soluciones después de 1, 5, 10 y 20 iteraciones.....	33
Figura 6. Diagrama para realizar ED.....	34
Figura 7. Creación del vector de variación V	36
Figura 8. Creación del vector de prueba.....	38
Figura 9. Diagrama para el diseño del costo óptimo de RDA.....	41
Figura 10. Grafo del primer caso de estudio.....	44
Figura 11. Grafo del segundo caso de estudio.....	46
Figura 12. Red de Pescara, Italia.....	48
Figura 13. Red de Balerna, España.....	51
Figura 14. Matriz de adyacencia.....	53
Figura 15. Diagrama de flujo del algoritmo de Dijkstra.....	54
Figura 16. Descomposición del primer caso de estudio.....	55
Figura 17. Diagrama de flujo del Algoritmo de Evolución Diferencial.....	58
Figura 18. Diagrama de flujo del Algoritmo Genético.....	59
Figura 19. Costo vs. número de iteraciones, primer caso.....	64
Figura 20. Costo vs. número de iteraciones, segundo caso.....	68
Figura 21. Costo vs. número de iteraciones, tercer caso.....	72

INTRODUCCIÓN

Las redes de distribución de agua (RDA) están entre las infraestructuras básicas que son esenciales para la actividad humana. Son necesarias para asegurar un suministro de agua seguro y confiable. El propósito principal de las RDA es abastecer la cantidad de agua necesaria con presión suficiente desde el reservorio hasta el usuario final. Por lo tanto, encontrar un diseño económico para su implementación es de gran interés. En la práctica, cuando se diseña el análisis hidráulico de los elementos en la RDA, la relación entre el diámetro de la tubería y el costo es no lineal. En concreto, la complejidad está presente en la ecuación de energía usada para calcular el flujo de agua en la tubería y la pérdida de carga neta por fricción. Además, debido a que la dirección del flujo a lo largo de la tubería es desconocida en redes enlazadas, realizar un diseño con costo óptimo es de mayor dificultad.

En este trabajo se busca desarrollar y aplicar un método de optimización de tipo heurístico que permita minimizar el costo de una red de acueductos, donde se conoce a priori el esquema de distribución o grafo de la red, a partir del cual se escogerán los diámetros de todas las tuberías que la componen, empleando un algoritmo de combinación de tipo adaptativo, donde se parte de un conjunto de diámetros nominales disponibles para las tuberías de la red, seleccionados aleatoriamente a través de un proceso iterativo que involucra de manera sistemática, la combinación, evaluación y perturbación de los conjuntos de diámetros seleccionados, hasta conseguir una posible solución aceptable del costo total de la red. Dicha solución debe acercarse a las condiciones de presiones mínimas requeridas en los nodos para la distribución del agua. A fin de reducir el tiempo de cálculo en redes con gran cantidad de tuberías, se empleará una estrategia de subdivisión de la red en k subredes abastecidas por un depósito de agua a la vez, donde se sugiere una metodología de búsqueda de óptimos basada en algoritmos evolutivos diferenciales (ED) que puede hallar una configuración económica admisible de diámetros para cada tubería de la subred. Finalmente, la solución óptima se obtiene de implementar el algoritmo de ED en la RDA original, con la ventaja de conocer los posibles diámetros candidatos del proceso anterior y por lo tanto reducir de manera significativa el tiempo de cálculo.

La implementación del algoritmo de optimización se hará utilizando un código previo desarrollado por Velásquez (2022), bajo la tutoría del Prof. Eugenio Sansone en lenguaje FORTRAN, que permite la solución estacionaria de redes de acueductos de cualquier tipo, mediante el Método del Elemento Finito, obteniendo los resultados del caudal en los elementos de la red y las cargas piezométricas en los nodos, por lo que el trabajo se realizará sobre un código académico realizado en la Escuela de Ingeniería Mecánica de la Universidad Central de Venezuela y donde no será necesario utilizar en ningún momento código libre o comercial para calcular las variables del flujo de la red, tales como EPANET. En tal sentido, se desarrollarán los módulos de optimización necesarios que permitirán, una vez que sean ingresados los datos iniciales, minimizarla en costo durante una etapa intermedia, para luego proceder con la segunda etapa de optimización, con un número de variables reducida.

La ventaja del modelo propuesto permitirá impulsar el diseño de redes de distribución de agua basadas en la optimización, consiguiendo emplear un algoritmo que permita lograr resultados satisfactorios en la búsqueda de redes confiables en su servicio (comportamiento hidráulico) para un costo viable, con un tiempo computacional reducido.

CAPÍTULO I

1. FUNDAMENTOS DE LA INVESTIGACIÓN

1.1 Planteamiento del problema

Los sistemas de distribución de agua son obras de infraestructura pública costosas ya que requieren un alto nivel de inversión de capital para la construcción, además de una inversión continua para el mantenimiento. Por lo tanto, la investigación sobre el diseño óptimo de redes de distribución de agua (RDA) está motivado por la posibilidad de ahorros sustanciales en los costos. El proceso de optimización implica identificar la red de tuberías de menor costo que pueda proporcionar la demanda y la carga hidráulica requerida en cada nodo de suministro.

El diseño de RDA es un problema complejo, ya que la formulación del problema es no lineal y supone un problema matemático de gran complejidad, incluso para redes de tamaño moderado. Hallar un diseño óptimo tomaría mucho tiempo mediante métodos deterministas convencionales debido a las restricciones hidráulicas, y la necesidad de considerar múltiples variables en su instalación (topología, tanques, accesorios, bombas, velocidad, presión, etc.) en conjunto con las diferentes soluciones locales. Para este tipo de casos, el comportamiento hidráulico y sus variables suelen ser analizados con la ayuda de herramientas computacionales, a su vez se les conoce como códigos de dinámica de fluidos computacional (DFC) que ofrecen un gran potencial para ahorrar tiempo en el proceso de cálculo.

Con el fin de hallar un diseño de costo óptimo en redes de acueductos, se propone una estrategia que funcione en conjunto con métodos de análisis DFC, se modela una RDA que funcione mediante gravedad, sin accesorios. Que a partir de unos datos de entrada como la topología, altura geodésica, carga de presión requerida en los nodos, diámetros de tuberías comerciales disponibles, entre otros; sea capaz de llevar a cabo el proceso de optimización automáticamente sin la necesidad de realizar ajustes específicos, además de

cumplir con las restricciones en cuanto a la presión del servicio y caudal en los nodos de alimentación y suministro.

En las últimas décadas, se han realizado importantes investigaciones que emplean diferentes métodos para optimizar el diseño de RDA. En la mayoría de estos métodos, la estrategia de búsqueda se basa en usar métodos heurísticos para obtener resultados aceptables. Para este caso, se ha encontrado que los algoritmos evolutivos (AE) son capaces de resolver problemas de diseño de redes con buenas probabilidades en encontrar soluciones cercanas al óptimo, manejar restricciones hidráulicas y tamaños de tubería discretos con facilidad. Sin embargo, un inconveniente importante con el uso de AE es que requieren un gran número de evaluaciones de la RDA para encontrar soluciones óptimas (especialmente para casos relativamente grandes), lo que resulta en una costosa carga computacional.

Se han realizado muchas investigaciones en un intento de mejorar la eficiencia de los AE aplicados a problemas de optimización. A este respecto, en las últimas dos décadas se ha vuelto popular el uso de algoritmos de evolución diferencial (ED) por ser muy efectivo en resolver todo tipo de problemas de optimización, el cual es un tipo de AE más particular. En cuanto a optimización en redes de acueductos, Vasan y Simonovic (2010) y Suribabu (2010) concluyeron que el rendimiento de los algoritmos ED era al menos tan bueno, si no mejor, que el de otros AE que se han empleado para resolver esta clase de problema. En la tabla 1 se muestra diferentes algoritmos empleados en la optimización de redes de acueductos, en comparación a otros algoritmos, la ED posee un bajo número de evaluaciones necesarias de la función costo para hallar un diseño óptimo.

Tabla 1. Algoritmos usados para optimizar RDA. Obtenido de Suribabu (2010)

Autores	Técnica usada	Promedio de evaluaciones de función objetivo
Suribabu (2010)	Evolución Diferencial	4750
Suribabu y Neelakantan (2006)	Optimización por Grupo de Partículas	5138
Liong y Atiquzzaman (2004)	Algoritmo de Transposición	1019
Neekantan y Suribabu (2005)	Algoritmo Genético Modificado	2440
Eusulf y Lansey (2003)	Algoritmo Salto Aleatorio	11155
Cunha y Sousa (1999)	Algoritmo de Transformación	25000
Savic y Walters (1997)	Algoritmo Genético	65000

Como seguimiento de esta actividad, la descomposición de la RDA original que utiliza la teoría de grafos para facilitar el proceso de optimización es una de estas líneas de investigación, Zheng, Simpson y Zecchin (2013) sugieren realizar una optimización multietapa donde primero se aplica el algoritmo DE en cada una de las subredes con el fin de encontrar los primeros posibles diámetros, luego estos resultados se toman en cuenta para una segunda etapa del algoritmo de ED aplicado en la red original, contando con la ventaja de una menor selección de diámetros. Todo esto con el fin de reducir en gran medida el área de búsqueda y las iteraciones necesarias para la convergencia.

Muchos de los modelos basados en AE presentados en la literatura incorporan todas las variables de decisión durante todo el proceso de búsqueda. Por lo tanto, los modelos derivados podrían no tener en cuenta las implicaciones en tiempo de computación de realizar el proceso de optimización considerando todos los diámetros posibles en una sola implementación. Se requiere de metodologías especiales que haciendo uso de un tiempo computacional razonable permita obtener una configuración de diámetros admisible con las

limitaciones de velocidad, caudal y presión, así como en alcanzar valores aceptables en los objetivos donde se buscan mejores costos y mayor confiabilidad.

En este estudio se detalla el fundamento teórico y la metodología a seguir en la estrategia de ED para desarrollar un modelo aceptable y eficiente que pueda proveer diseños de RDA factibles y económicas que puedan facilitar el proceso de la toma de decisiones.

1.2 Objetivos

1.2.1 Objetivo General

Optimizar el costo de redes de acueductos mediante el desarrollo de un algoritmo de búsqueda heurístico, adaptativo, multietapa.

1.2.2 Alcances

- Realizar una investigación básica sobre los métodos existentes más recientes que se están empleando en la optimización del costo de redes de acueductos, que involucran procesos de combinación heurística adaptativa y seleccionar el más conveniente.

- Establecer el algoritmo para realizar la partición de una red de acueductos en un conjunto de subredes que tienen una fuente o depósito de agua a la vez, utilizando un algoritmo similar al de Dijkstra en la Teoría de Grafos, que permite encontrar las rutas óptimas a cualquier nodo de la red, partiendo de un depósito en particular.

- Establecer el algoritmo de optimización heurística adaptativa que será aplicado a cada subred, con el propósito de encontrar los diámetros que mejor se adapten a los requerimientos de presiones mínimas establecidas en todos los nodos de la red.

- Programar los módulos de generación de subredes y el de optimización, que serán insertados en el código desarrollado por el Tutor.

- Probar distintas redes de acueductos analizando los costos luego del algoritmo desarrollado.

- Redactar el informe con la conclusión y las recomendaciones que se deriven de este trabajo.

1.3 Antecedentes

A continuación, se describe una serie de investigaciones previas que dan sustento a esta investigación, aportando conceptos que orientan el desarrollo teórico de este trabajo. Con el fin de orientar el enfoque metodológico de la investigación.

Se han venido realizado importantes investigaciones con el objetivo de conseguir un diseño económico en redes de acueductos (RDA), para ello se dispone de múltiples técnicas de optimización que hacen uso de métodos tradicionales, como el de prueba y error utilizado en Gómez (2013), y programación lineal García (1977) aunque esta tiene sus limitaciones, ya que el problema de diseño de la RDA es no lineal por naturaleza, se utiliza el enfoque en programación no lineal (PNL) en redes de acueductos de Shamir (1974). Recientemente son cada vez mas usados los algoritmos evolutivos (AE) ya que han demostrado ser eficientes en encontrar buenas soluciones (resumidos en la Tabla 2) y pueden manejar redes complejas con múltiples opciones de diámetro de tubería con mayor facilidad que sus contrapartes.

Mediante el uso de los AE el problema se puede modelar de manera más precisa debido a la facilidad de manejar relaciones no lineales del modelo formulado. Los AE empiezan con soluciones iniciales aleatorias para cubrir todo el espacio de búsqueda y estrategias para generar mejores propuestas, que están sujetas a una función objetivo con el fin de minimizar el costo capital. Consecuentemente se evalúan todas las soluciones hasta cumplir con el criterio de finalización del algoritmo. Por lo tanto, la solución final depende en gran medida de como se generan los vectores, el proceso de búsqueda, perturbación,

combinación y selección de diámetros para generar redes económicas que puedan satisfacer las demandas y presiones de consumo.

Tabla 2 - AE usados en la optimización de RDA. Elaboración Propia.

Algoritmo de Transformación	Gonzalez M. 2013
Búsqueda armónica	Mora D. 2012
Evolución Diferencial (ED)	Suribabu 2010
AG Modificado	Kadu, Gupta y Bhave 2008
Optimización por Grupo de Partículas	Kumar y Reddy 2007
Colonia de Hormigas	Zecchin y otros 2007
Algoritmo Genético (AG)	Montesinos y otros 1996

Numerosos trabajos se han realizado en un intento de mejorar la eficiencia del algoritmo y algunas de ellas también consideraron ciertos aspectos de la confiabilidad, Simonovic, P. y Vasan, A. (2010) optimizó la red de Nueva York y Hanoi mediante ED y recomendó el maximizar la resiliencia de la red, Sirsant, S. y Janga Reddy, M. (2022) señalan que usualmente se usan números aleatorios para abarcar la mayor cantidad de posibles soluciones, por ello diseñó una “población” o conjunto de soluciones (el cual se usará de ahora en adelante) generado de números pseudoaleatorios para una convergencia más rápida en regiones de mayor interés. También toma en cuenta la confiabilidad mecánica de la RDA en el análisis de diseño óptimo, de la misma manera que Pino E. y otros (2017) que además se fundamentaron en la optimización multiobjetivo para obtener el menor costo de la red y la mayor confiabilidad del servicio. Es de notar en el trabajo de Pino, la posibilidad de contar con un conjunto de soluciones optimizadas, pudiendo elegirse el diseño que tenga el menor costo posible con la contrapropuesta de una baja confiabilidad.

En los últimos años se han ampliado los requerimientos de optimización y se han formulado nuevas estrategias para reducir el costo computacional. La ED ha conseguido buenos resultados en la optimización de RDA, estudios más recientes como son los de

Mansouri y Mohamadizadeh. (2017) y Bilal Pant (2022) lograron optimizar las RDA de Ismail Abad y los túneles de Nueva York, Mansouri y otros (2015) empleando el algoritmo DE logró reducir el costo en 10,66% en comparación con métodos clásicos empíricos como el de prueba y error, estudiando la RDA de Ismail Abad. Por otra parte, el valor de las constantes de “mutación y cruce” (las cuales se llamarán variación y combinación durante el desarrollo de la investigación) juegan un rol importante en el resultado del proceso de optimización, por ello investigadores como Sirsant y Janga Reddy (2022) y Zheng, Simpson y Zecchin (2013b) diseñaron una auto-adaptación del algoritmo para hallar la mejor configuración de estas constantes. En relación a esto, Angira y Babu (2005) señalan que los parámetros de control y las estrategias que funcionan bien para un problema dado pueden no funcionar bien para otros problemas. Por lo tanto, es necesario ajustarlos o diseñarlos de nuevo para cada problema específico.

La siguiente fase de la investigación se centrará en los algoritmos evolutivos diferenciales para el proceso de optimización. La cual es una técnica de optimización propuesta por Storn y Price (1995). Estudios previos han demostrado que es un método de optimización más rápida que otros AE, con pocas variables de control, fácil de usar, eficiente en dominios continuos y se puede aplicar en cualquier campo. Desde entonces se han venido empleado distintas estrategias de ED para encontrar mejores soluciones.

En la evolución diferencial (ED), los operadores de variación se utilizan para explorar el espacio de búsqueda. Estos operadores se pueden interpretar como variación y combinación darwiniano, donde los vectores representan “individuos” en un conjunto de soluciones. La ED utiliza la selección para conservar los mejores vectores en el conjunto de soluciones para la siguiente iteración. De esta forma, las estrategias de variación determinan cómo se debe explorar el espacio de búsqueda. Fister y otros (2022).

En el pasado reciente, algunos investigadores han dado grandes avances prácticos y teóricos en la ED. Autores como Chakraborty (2008), Feoktistov (2006), Kumar, Oliva y Suganthan (2022), Onwubolu y Davendra (2009) y Price, Storn y Lampinen (2005) incluyeron en sus libros varias aplicaciones exitosas de la ED para varios problemas de

ingeniería. Estas investigaciones han encontrado que el algoritmo de búsqueda evolutivo diferencial es bastante apropiado para la optimización combinatoria y que es en efecto, competitivo con otros enfoques en este dominio. También en algunos avances recientes los investigadores han mejorado la convergencia del algoritmo probando modificaciones en el método de ED. Aquellas utilizadas en la optimización de RDA se listan en la Tabla 3.

Tabla 3 - Algoritmos ED en la optimización de RDA. Elaboración Propia.

Sirsant, S., Janga Reddy, M. 2022
Bilal Pant, M. 2022
Pino, E. y otros 2017
Mansouri, R. y Mohamadizadeh, M. 2017
Ho Min Lee y otros 2016
Mansouri, R. y otros 2015
Zheng F, Simpson A, Zecchin A. ED Autoadaptativo 2013
Zheng. F., Simpson, A., Zecchin, A. 2013
Suribabu, C. 2010
Vasan, A. y Simonovic, P. 2010

Las investigaciones relevantes en este campo, enfatizaron en encontrar un método capaz de reducir el costo computacional debido a la dificultad de resolver la red hidráulica. Kadu, Gupta y Bhave (2008) basándose en el método de la ruta crítica encontró que al reducir el espacio de búsqueda era posible generar soluciones factibles con mayor rapidez y obtener una solución admisible en un menor número de iteraciones.

Atendiendo a estas consideraciones, se ha buscado implementar conceptos de la teoría de grafos para dirigir la búsqueda de soluciones alrededor del área más prometedora del espacio de soluciones, y de igual manera se puede abordar el problema combinatorio de posibles soluciones de la RDA, que es análogo al problema del viajante vendedor (*Travel Salesman Problem* o TSP por sus siglas en inglés) que se trata en la teoría de grafos Deo

(2017). Como seguimiento de esta actividad, en el trabajo de Zheng, Simpson y Zecchin (2013) y H. M. Lee y otros (2016) investigaron el diseño del costo óptimo mediante la subdivisión de redes según la cantidad de reservorios de agua. En el trabajo de Zheng y otros se propone el criterio de la pendiente hidráulica que se emplea para asignar cada nodo a una de las subredes. Por otra parte en el trabajo de H M Lee y otros, se estudia más a fondo la subdivisión de las RDA de múltiples fuentes empleando la herramienta de rastreo de fuente de EPANET que permite cuantificar el porcentaje de agua que reciben los nodos para determinar las subredes. Los enfoques de ambos trabajos se analizan más adelante en el diseño de la metodología a seguir.

CAPÍTULO II

2. CONCEPTOS Y ECUACIONES BÁSICAS

2.1 Variables y conceptos de naturaleza hidráulica

2.1.1 Tubería

Conducto que cumple con la función de transportar agua u otros fluidos entre diferentes localidades de un sistema. Múltiples cargas actúan sobre las tuberías durante su funcionamiento. La presión interna y el peso propio del tubo y del agua, son las principales cargas que las tuberías deben soportar, aunque cargas puntuales externas, sísmicas, térmicas y del viento pueden influir también.

2.1.2 Redes de Distribución de Agua (RDA)

Una red de distribución de agua o RDA contiene una serie de componentes, que se conectan apropiadamente entre sí, con el fin de suministrar el agua requerida a una presión determinada. Esta se compone por tuberías, válvulas, bombas, depósitos, zanjas, ventosas, hidrantes, y otros componentes.

2.1.3 Nodos Fuente o Reservorios

Reciben una entrada de caudal que es suministrada al resto de componentes de la red.

2.1.4 Nodos de Consumo

Puntos de la RDA que tienen una demanda de caudal y presión.

2.2 Ecuaciones generales que determinan el estado estacionario de una red

Como es sabido, la carga total neta que fluye en un sistema de acueductos se cuantifica habitualmente, en metros de columna de agua (mca) Suponiendo la incompresibilidad del flujo, la carga total neta del flujo se expresa como:

$$Z + \frac{P}{\gamma} + \frac{u^2}{2g} \quad (2.1)$$

Donde:

Z : Cota geométrica o elevación. Representa la energía potencial respecto de una cota de referencia o datum.

$\frac{P}{\gamma}$: Carga de presión, usualmente se considera como el valor de la presión manométrica o absoluta, de modo que la presión atmosférica toma el valor de cero.

$\frac{u^2}{2g}$: Carga cinética, generalmente despreciable en relación a la carga piezométrica ($\frac{P}{\gamma} + z$ en el caso de flujos incompresibles) mientras que los desniveles en la red y la presión llegan a alcanzar varias decenas de metros de columna de fluido.

γ : Peso específico del fluido (en el caso del agua, 9800 N/m^3 a 15°C).

g : Aceleración de gravedad.

La ecuación de Bernoulli afirma que la energía total del flujo, se mantiene constante a lo largo de una línea de corriente si este se considera como un flujo incompresible y admitiendo que no existen pérdidas por fricción ni aportes de energía. Ello significa que la energía del flujo puede sufrir transformaciones de una forma a otra a lo largo de la conducción, pero permaneciendo constante la energía total. Cuando entre dos secciones 1 y

2 de la conducción existen pérdidas por fricción, es más conveniente utilizar la Ecuación de la Energía que toma en cuenta dicha pérdida y que se puede escribir como:

$$Z_1 + \frac{P}{\gamma} + \frac{u_1^2}{2g} = Z_2 + \frac{P_2}{\gamma} + \frac{u_2^2}{2g} + h_{12} \quad (2.2)$$

En relación a la energía del fluido se suele operar con los siguientes conceptos:

2.2.1 Carga piezométrica

$$H = Z + \frac{P}{\gamma} \quad (2.3)$$

Corresponde a la suma de los términos de carga de presión y geodésica. Lo que resulta más práctico en aquellos sistemas donde la carga cinética presenta un valor despreciable.

2.2.2 Carga total

$$H_t = Z + \frac{P}{\gamma} + \frac{u^2}{2g} \quad (2.4)$$

Se representa por la suma de los tres términos de altura. Estos se describen en la figura 1.

2.2.3 Carga de velocidad o carga cinética

Es la diferencia entre la carga total y la carga piezométrica.

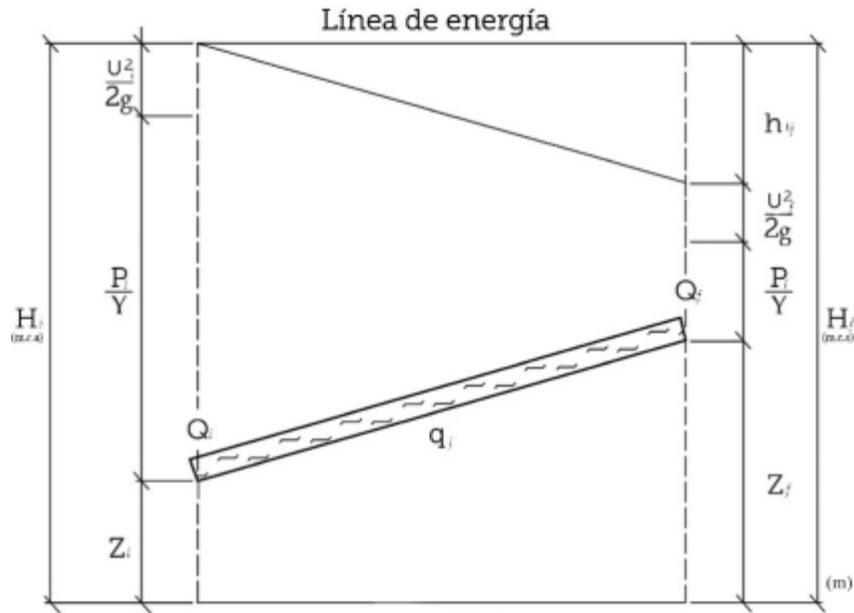


Figura 1. Línea de energía hidráulica. Obtenido de Corvera y otros (2021)

2.2.4 Pérdidas de carga neta por fricción (h_L)

Se producen a lo largo de las tuberías debido al rozamiento mecánico del flujo en su recorrido a través de la misma. En cada tubo la energía total disponible se disipa en pérdidas por fricción y pérdidas menores debidas a recirculación o flujos secundarios.

$$H_t = h_f + \sum k_m \quad (2.5)$$

Para determinar las incógnitas del sistema se diseña un análisis en régimen permanente, considerando flujo turbulento, uniforme, estacionario, unidimensional e incompresible.

En general el problema de análisis en régimen permanente de una RDA puede resumirse en la determinación de los caudales en cada tuberías, así como de las cargas piezométricas en los nodos del sistema, a partir de la información disponible sobre las características de todas las conducciones, elementos especiales, los consumos y aportes de la red y al menos, una altura piezométrica de referencia conocida. Según Potter y Wiggert

(2012) el mayor problema radica en la elección de los diámetros como variables de decisión; puesto que en este caso las restricciones son funciones implícitas de estos mismos diámetros (Ecuaciones 6, 7 y 8).

$$h_L = 1.07 \frac{LQ^2}{gD^5} \left(\ln \left[\frac{e}{3.7D} + 4.62 \left(\frac{vD}{Q} \right)^{0.9} \right] \right)^{-2} \quad (2.6)$$

$$\dot{Q} = -0.965 \left(\frac{gD^5 h_L}{L} \right)^{0.5} \ln \left[\frac{e}{3.7D} + \left(\frac{3.17v^2 L}{gD^3 h_L} \right)^{0.5} \right] \quad (2.7)$$

$$D = 0.66 \left[e^{1.25 \left(\frac{LQ^2}{gh_L} \right)^{4.75}} + vQ^{9.4} \left(\frac{L}{gh_L} \right)^{5.2} \right]^{0.04} \quad (2.8)$$

Se hace uso de tres ecuaciones generales, que se cumplen independientemente de la configuración y los elementos de los que consta la red y que constituyen una particularización de las ecuaciones generales de conservación de la masa y la energía aplicadas a un fluido que fluye a través de un sistema de tuberías a presión. Las condiciones del método son las siguientes:

2.2.5 Ecuación de la continuidad

$$\sum (\pm)_j \dot{Q}_j - \dot{Q}_e = 0 \quad (2.9)$$

Donde:

Subíndice j: representa el caudal saliente de una conducción.

Subíndice e: representa el caudal entrante a la conducciones.

\dot{Q} : Caudal externo a la red (Demanda externa)

La convención de signo \pm depende de la dirección del flujo supuesto para cada elemento, si el flujo sale del nodo es positivo y negativo si el flujo es entrante al mismo.

2.2.6 Balance de carga neta alrededor de un lazo interno

$$\sum (\pm)_i H_{Ti} \quad (2.10)$$

Donde:

H_{Ti} : carga neta

Subíndice i : representa los elementos que componen el lazo.

Se supone que no hay bombas en el lazo, la convención de signo utilizada expresará de forma positiva el recorrido del lazo en sentido horario y en forma negativa para el recorrido antihorario.

2.2.7 Balance de carga neta en lazos cerrados

En el caso donde se aplican métodos de corrección progresiva de caudales residuales, como el de Hardy-Cross, el balance a lo largo de una trayectoria única o pseudo lazo que conecta dos nodos de nivel fijo, se hace mediante la siguiente ecuación:

$$\sum \pm [h_L - H_p]_i + \Delta H_T = 0 \quad (2.11)$$

Donde:

1. Subíndice i : representa el lazo
2. ΔH_T : Diferencia de carga total entre los nodos.
3. H_{pi} : Carga neta de bombeo.

La convención de signo sigue el mismo argumento de la ecuación 2.2.10.

2.3 Variables y conceptos en la optimización de redes

Los problemas de optimización son cotidianos en la ingeniería y usualmente es necesario un algoritmo robusto capaz de resolver aquellos dilemas fundamentales en el trabajo diario. Este proceso no debería ser difícil en sí mismo, lo más adecuado es que sea fácil de usar, poderoso, confiable y rápido.

2.3.1 Optimización

Empleando las palabras de Price, Storn y Lampinen (2005), se puede definir como: “el intento de maximizar las propiedades deseables de un sistema y, al mismo tiempo, minimizar sus características indeseables”.

2.3.2 Optimización local y global

El proceso de optimización implica hallar el conjunto de entradas para una función objetivo que resulte en la salida de un mínimo o máximo valor o conjunto de valores. En esta investigación es de interés conocer el conjunto de parámetros que resultan en el costo mínimo de la RDA mediante el proceso de búsqueda del algoritmo de optimización, a su vez este puede terminar en un óptimo local o global.

La optimización local trata de acercarse a la mejor solución de una región determinada donde un proceso de búsqueda realiza de forma iterativa ligeros cambios en una nueva solución propuesta y la evalúa para ver si conduce a una mejora; la cual termina en un nuevo pico local e idealmente localiza una solución aceptable al problema propuesto si este se encuentra en la región de estudio. La figura 2 muestra un caso en el que se pueden observar varios mínimos locales. La optimización global intenta encontrar el extremo mínimo o máximo de la función objetivo para todo el espacio de búsqueda.

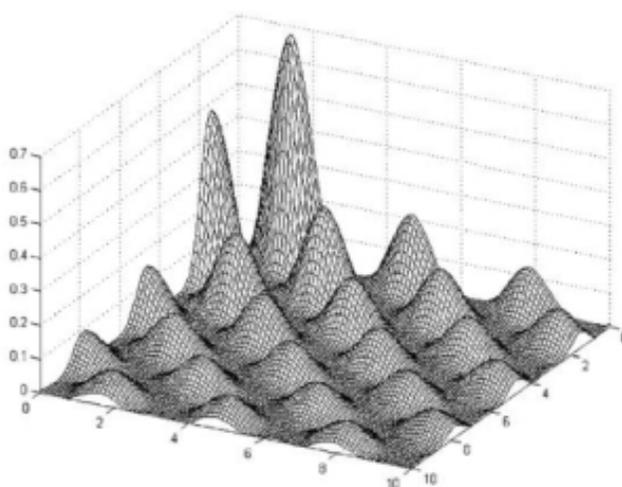


Figura 2. Vista tridimensional de la función objetivo.

Obtenido de Feoktistov, V. (2006)

2.3.3 Grafo

Está formado por puntos o nodos y líneas o aristas. Cada arista tiene un nodo en sus extremos. A fines de modelar la red de acueductos, se consideran las líneas como tuberías y los nodos como lugares donde las tuberías se interconectan, se diferencian entre nodos de consumo y nodos fuente.

2.3.4 Tuberías de partición

Es un conjunto de tuberías representadas en el grafo como aristas, que al ser eliminadas de éste, quedan separados en conjuntos independientes sin elementos en común. En la figura 3 se puede observar cómo se divide un grafo removiendo un conjunto de tuberías.

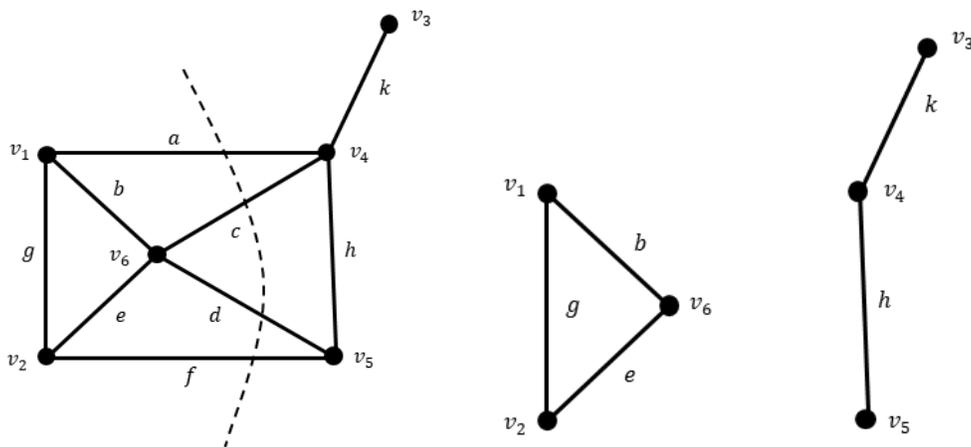


Figura 3. Remoción del conjunto de partición $\{a,c,d,f\}$ del corte de un grafo.

Obtenido de Deo N. (2017)

2.3.5 Problema del viajante en la teoría de grafos

Una característica frecuente en los problemas de optimización es la necesidad de conocer la configuración más eficiente entre una gran variedad de opciones. Este se encuentra estrechamente relacionado con el caso del vendedor viajante planteado de la siguiente manera: un vendedor debe visitar varias ciudades durante un viaje, conociendo las distancias entre ellas ¿en qué orden debe viajar para visitarlas todas exactamente una vez,

con la mínima distancia recorrida? Este problema se cataloga como un problema combinatorio debido a la gran variedad de soluciones posibles. Con el fin de aplicar ordenadamente un algoritmo, se puede establecer un vector con todas las distancias, representar las ciudades por vértices y las carreteras por aristas, obteniendo un grafo que puede ser estudiado con mayor facilidad.

2.3.6 Algoritmo

Se define como una serie de operaciones organizadas de manera lógica y sistemática, que permite llevar a cabo un objetivo determinado.

2.3.7 Algoritmo de Dijkstra

También llamado algoritmo de camino más corto, busca la menor distancia entre un nodo especificado y el resto de nodos pertenecientes a un grafo con pesos en cada arista. En el caso de un grafo como una RDA, la distancia de una ruta es la suma de los pesos de las aristas que lo conforman. El algoritmo trabaja construyendo un conjunto de nodos que tienen una mínima distancia a un nodo fuente.

2.3.7.1 Esquema del camino más corto

1- Se crea un vector que lleva el registro de los nodos visitados. Los nodos que ya fueron analizados no se vuelven a evaluar.

2- Se utiliza un vector que lleva el recorrido desde el nodo fuente hasta los demás nodos. El nodo fuente se marca con 0 y la distancia hasta los demás nodos en infinito.

3- Se considera como nodo actual, aquel nodo con la menor distancia del vector recorrido. Como el nodo fuente tiene recorrido cero, será el nodo donde comience el análisis.

4- Se calcula la longitud del camino hacia los nodos adyacentes, añadiendo el peso que existe hasta el siguiente. En esta etapa solo se analizan los nodos vecinos al nodo actual.

5- Ahora, si los nuevos caminos encontrados hasta los nodos adyacentes, son menores que aquellos almacenados en el vector recorrido, entonces se reemplazan. De no ser así, se ignoran.

6- Se marca el vértice actual como procesado después de visitar los nodos vecinos.

7- Selecciona el nodo con el menor recorrido como el nuevo nodo actual y de regreso hacia el paso cuatro.

8- Repetir el proceso hasta que todos los nodos son marcados y procesados.

Una vez completado el algoritmo, se puede construir el camino hasta el nodo fuente para encontrar el camino más corto.

2.3.8 Algoritmos Evolutivos (AE)

Se consideran AE las técnicas de búsqueda de soluciones basadas en la tesis de Darwin. A un conjunto de vectores de diámetros se define como el conjunto de soluciones, que a menudo se inicializa al azar para abarcar el mayor espacio en la zona de búsqueda donde se sabe se encuentra el óptimo, esta fase se conoce como inicialización. En el caso de RDA, el espacio de búsqueda incluye todos los diámetros permitidos, donde solamente una configuración de diámetros determinada tendrá el menor costo, cumpliendo con los requerimientos de la red. A continuación, se establece el criterio de optimización, que en general es una evaluación para cada vector del conjunto de soluciones, esto se realiza para puntuar a cada vector y conocer cuales están mas cerca de ser el óptimo. Así, el algoritmo inicia su ciclo evolutivo con iteraciones, hasta que se alcanza la condición de parada Feokstítov (2006). Otra característica de los AE es que son considerados como los mejores métodos de optimización estocástica, ya que su resultado no es determinado, y varía en el tiempo en función de elementos aleatorios, con el fin de proveer distintas soluciones en cada ejecución, Mora (2009).

El ciclo evolutivo consiste de los siguientes pasos: 1- Selección del vector padre o jerárquico de base para reproducirse. 2- Variación aleatoria del vector seleccionado mediante operadores como combinación y selección. 3- Reemplazo del conjunto de soluciones mediante selección elitista.

2.3.9 Programación Lineal (PL)

En la programación lineal, las variables de decisión se consideran que tienen un comportamiento lineal, lo que significa que su contribución a la función objetivo y a las restricciones del problema es directamente proporcional a su valor Mora (2012). En general la técnica permite una buena aproximación de la realidad, además cuenta con métodos como el Gradiente de Programación Lineal que permite mejorar la solución obtenida ajustando el valor de las variables.

2.3.10 Programación no Lineal (PNL)

Según describe Mora (2012), la programación no lineal es un enfoque para resolver problemas de optimización que se pueden expresar como un sistema de igualdades y desigualdades. Las restricciones de la PNL pueden ser lineales o no lineales. Un aspecto a considerar es que no es posible garantizar que la solución obtenida sea el óptimo global.

2.3.11 Algoritmo Heurístico

Se plantean para hallar una solución aproximada de un problema específico haciendo uso de sus características. Un algoritmo heurístico se diferencia de los algoritmos aproximados porque estos encuentran soluciones aproximadas, en un tiempo aceptable, así como indicar el grado de discrepancia con la supuesta solución óptima. Este simplemente encuentra una solución lo suficientemente buena, dentro de un tiempo aceptable.

2.3.12 Elitismo

Es una estrategia que se usa para seleccionar la mejor solución. En la selección elitista, los vectores con el mejor valor de aptitud se conservan para la próxima iteración. Esto garantiza que los mejores parámetros de un conjunto de soluciones nunca se pierdan Feokstiov (2009). Esta operación se ilustra en la figura 4.

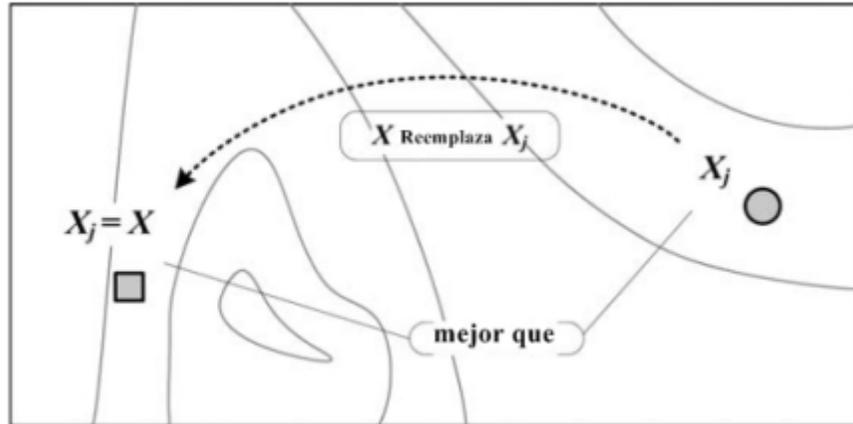


Figura 4. Selección elitista: si el vector de prueba es mejor que el vector actual X_j , entonces es reemplazado. El conjunto de soluciones siempre se actualiza.

Obtenido de Feokstítov (2009).

2.4 Función Objetivo

La idea central de la función objetivo es evaluar las “aptitudes” de los vectores y partir de una solución que se intenta mejorarlas explorando en direcciones que resulten en una minimización de dicha función. Para ello se definen las reglas de exploración y cambio entre cada conjunto de soluciones. En general el problema de optimización de RDA manejadas por descenso requiere hallar la combinación de tamaños de tubería que otorga el mínimo costo de construcción; sujeto a las restricciones de continuidad de flujo, considerando pérdidas por fricción y carga de presión mínima que pertenecen al rango de diámetros de tuberías comerciales disponibles.

Matemáticamente, se puede expresar como:

$$\text{Minimizar } f(D_1, \dots, D_x) = \sum u(D_x)L_x; \quad x = 1, \dots, X \quad (2.12)$$

Donde:

$u(D)$: unidad de costo de dicho diámetro D

L_x : longitud de la tubería

Sujeto a:

2.4.1 Restricciones Hidráulicas

Se utilizó el Método del Elemento Finito para calcular los caudales en los tubos y las cargas piezométricas de los nodos que satisfagan las ecuaciones de conservación de masa y energía de la red, de acuerdo al trabajo de Velásquez (2022) bajo la tutoría del Profesor Eugenio Sansone.

2.4.2 Restricciones Técnico-Normativas

La investigación se apoya en la Norma Venezolana vigente para cálculo de redes de acueductos, junto con otras normativas que serán de utilidad para definir aspectos que se deben considerar en el diseño de redes de acueductos. La ley venezolana está hecha con el fin de regir las dotaciones en viviendas y construcciones con consumo de agua, a diferencia de este trabajo que está enfocado en proveer un diseño competente con las demandas hidráulicas, en este caso un nodo de consumo puede representar un complejo industrial, residencial, comercial o cualquier actividad que requiera un consumo específico de caudal a una presión dada. Es importante realizar el diseño adecuado de las RDA, ya que el no hacerlo trae como consecuencia las fugas de agua.

(a) Límites comerciales: Los diámetros solo pueden tomar valores discretos dados por la disponibilidad del material y diámetros según el fabricante de tuberías.

$$D_x \in \{D_{\min}, \dots, D_{\max}\}; \quad x = 1, \dots, X \quad (2.13)$$

D_{\min} y D_{\max} = diámetro mínimo y máximo de tuberías comerciales disponibles, respectivamente.

(b) Presión requerida: La red debe prestar un servicio eficiente y continuo, por lo cual el diseño de las presiones debe atender a la condición más desfavorable del sistema, a fin de no provocar deficiencias en el sistema. Considerando la necesidad de hidrantes para combatir el fuego, los lineamientos establecidos por la ISO* establecen un mínimo de presión de 14 mca (1,4 Kg/cm² o 20 psi) en los nodos para abastecer de la altura suficiente en dichos casos.

* Cita de la Revista El Maquinista. Cap. (B). FPE. Manuel García Cruz. Instituto Autónomo Cuerpo de Bomberos del Estado Miranda, Venezuela. Nro 3.

No se ha encontrado una normativa clara en lo que a rangos de máxima presión operativa se refiere. Por tal motivo se refiere a la normativa peruana para el diseño de RDA para consumo humano dada por el Reglamento Nacional de Edificaciones en su norma OS.050. Establece que la presión mínima y máxima en cualquier nudo que conforme una red de distribución de agua potable debe estar entre 10 y 50 mca respectivamente, aunque usualmente los límites de presión varían según la RDA y tienen requerimientos específicos de presión.

$$H_j \geq H_j^{\min} \leq H_{\max}; \quad j = 1, \dots, N \quad (2.14)$$

Donde:

H_j^{\min} : carga de presión mínima permisible en nodo j.

H_{\max} : carga de presión máxima.

(c) Velocidad en los tramos: De acuerdo con la Gaceta Oficial 4044 de la República de Venezuela, se recomienda una velocidad mínima de 0,6 m/s para asegurar el arrastre de partículas y una máxima de 3,00 m/s para evitar ruidos en las tuberías. Si la velocidad no entra en el rango establecido, se debe variar el diámetro hasta lograr obtener valores que cumplan con la normativa. En casos justificados de 5,00 m/s para evitar la erosión excesiva.

(d) Restricciones adicionales: El método idóneo debe ser capaz de encontrar una solución apropiada para una amplia variedad de topologías, demandas en los nodos,

topografía, entre otros. Igualmente, debe ser capaz de encontrar una solución efectiva de forma rápida y eficiente.

2.5 Algoritmo Genético (AG)

Los AG se enmarcan dentro de los algoritmos evolutivos, que incluyen también las estrategias evolutivas y todas sus formas. Su funcionamiento se basa en la teoría de adaptación al entorno de un organismo según la tesis darwiniana. Por imitación de este proceso, los AG son capaces de crear variaciones de posibles soluciones ejecutando acciones aleatorias, parecidas a variaciones, selección y recombinación genética que siguen un criterio que permite la evolución de vectores hacia valores óptimos. El cual converge hacia el vector con mejor valor de aptitud, que representa el óptimo o en muchos casos una solución muy cercana al óptimo del problema y es obtenido de acuerdo a la función objetivo del problema a resolver.

2.5.1 Pseudocódigo AG

Aunque existe gran cantidad de variaciones, se puede describir la estructura de un AG en términos generales, de acuerdo con Amat (2019), para resolver problemas de optimización con algoritmos genéticos se describen las siguientes etapas:

- 1- Se codifica un conjunto de soluciones inicial de P vectores al azar.
- 2- Calcular la aptitud de cada vector del conjunto de soluciones. La aptitud está relacionada con el valor de la función objetivo para cada vector.
- 3- Reproducir un nuevo conjunto de soluciones manipulando los vectores de acuerdo a los siguientes pasos hasta generar una nueva iteración.
 - 3.1 Seleccionar dos vectores del conjunto de soluciones existente, donde la probabilidad de selección es proporcional a la aptitud de los vectores.
 - 3.2 Combinar los dos vectores seleccionados para generar un nuevo vector.
 - 3.3 Aplicar un proceso de variación aleatorio sobre el nuevo vector.
 - 3.4 Añadir el nuevo vector al nuevo conjunto de soluciones propuestos.
 - 3.5 Calcular la aptitud de los nuevos vectores.

4- Reemplazar el antiguo conjunto de soluciones por uno nuevo mediante selección elitista.

5- Si no se cumple un criterio de parada, volver al paso 2 hasta completar el número determinado de iteraciones.

2.6 Algoritmo de Evolución Diferencial (ED)

La técnica de ED presenta un marco algorítmico simple que básicamente sigue la estructura de un AE, pero difiere en la manera que se aplica la variación y combinación para dirigir la búsqueda hacia regiones con mejores soluciones. Los AG utilizan el mismo mecanismo de búsqueda, valiéndose además de la prueba probabilística, la selección y el cambio de información entre soluciones. La principal diferencia con los AG, es que la ED simplifica el proceso de selección; solamente el vector de solución que posea el menor costo se le permite avanzar a la siguiente iteración, sin necesidad de realizar más operaciones de selección, Suribabu (2010).

El conjunto de vectores que representan una solución se denomina conjunto de soluciones, para cada iteración el número del conjunto de soluciones no cambia durante el proceso de optimización. El conjunto de soluciones de diámetros candidatos se elige aleatoriamente, posteriormente se calcula el costo de cada vector del conjunto de soluciones mediante la función objetivo. En el siguiente paso, se generan nuevos vectores agregando la diferencia entre dos vectores del conjunto de soluciones a un tercer vector. Esta operación se llama “variación”. El vector variación se recombina a continuación con un vector predeterminado, el “vector objetivo” para ser combinados a continuación y conseguir el “vector de prueba”. Finalmente, el vector de prueba se compara con el vector objetivo y el que tenga el mejor costo pasa a la siguiente iteración si reduce el valor de la función objetivo. Esta operación puede denominarse selección. Se repite el proceso mencionado hasta cumplir con la condición de parada. Cada vector del conjunto de soluciones tiene que servir una vez como vector objetivo para que haya transcurrido una iteración Plagianakos, Tasoulis y Vrahatis (2008) y Suribabu (2010).

Storn y Price (1995) probaron diferentes variantes de ED, las cuales presentamos posteriormente en el apartado de estrategias de variación. Mediante un banco de pruebas extensivo se demostró que el algoritmo converge más rápido y con mayor certeza que el método de transformación adaptativo y sus variantes, las cuales poseen reputación de ser métodos muy poderosos. Este logra buenos resultados, especialmente en resolución de problemas de optimización en condiciones continuas, pero se puede ampliar su aplicabilidad para resolver problemas discretos y problemas del mundo real. La ED se ha utilizado en varias aplicaciones de ciencia e ingeniería para descubrir soluciones efectivas a problemas muy intrincados y no lineales sin apelar al conocimiento experto o algoritmos de diseño complejos, debido a esto se les conoce por ser de naturaleza heurística. Se destacan en el diseño de ingeniería, control, procesos químicos y biosistemas, procesamiento de imágenes, bioinformática, biología molecular, redes neuronales, toma de decisiones, planificación, etc... Todo esto hace que el algoritmo sea una herramienta de diseño de gran utilidad, permaneciendo accesible para aplicaciones prácticas, simple, poderoso, confiable y rápido Feoktistov (2006) y Price, Storn y Lampinen (2005).

2.6.1 Pseudocódigo ED

Notación: (Np : dimensión del conjunto de soluciones, F : factor de variación, Cr : Factor de combinación, NG : Condición de parada, P_G : Vector del conjunto de soluciones, $X_{i,G}$ vector de diámetros, $V_{i,G}$: vector variación, $u_{i,G}^j$: vector de prueba)

La estrategia básica se puede describir de la siguiente manera:

Paso 1: Generar vectores de prueba (vectores del conjunto de soluciones) aleatoriamente asignando a cada vector los diámetros comerciales disponibles.

$$P_G = \{X_{1,G}, \dots, X_{NP,G}\} \text{ con } X_{i,G} = \{X_{i,G}^1, \dots, X_{i,G}^D\}, i = 1, \dots, NP; G = 0;$$

Paso 2: Calcular la función objetivo para N vectores iniciales

$$\{f(X_{1,G}), \dots, f(X_{NP,G})\}$$

Contador = 1

Repetir hasta que Contador $\geq NG$

Paso 3: Realizar la variación diferencial para generar Np soluciones variantes V . Cabe señalar que los vectores seleccionados al azar deben ser distintos entre sí.

Para $i = 1, \dots, N$

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$$

Paso 4: Realizar el operador de combinación para generar soluciones de prueba $U_{i,G}$

Para $i = 1, \dots, N$

Para $j = 1, \dots, D$

Si Aleatorio(0,1) \leq CR,

$$u_{i,G}^j = v_{i,G}^j$$

Si No

$$u_{i,G}^j = x_{i,G}^j$$

Paso 5: Discretizar el diámetro continuo de la tubería a el diámetro más cercano de cada variable y entonces evaluar la función objetivo en el conjunto de vectores solución.

Paso 6: Tomar para la siguiente iteración el vector con el menor precio.

Contador = Contador + 1

2.6.2 Variables de inicio

Los parámetros necesarios para emplear la ED estándar, incluyen el tamaño del conjunto de soluciones, Np ; el factor de variación, F ; tasa de combinación, Cr ; la condición de parada o cantidad máxima de iteraciones, NG ; y los límites estipulados para el problema. Múltiples investigaciones han demostrado que el comportamiento de búsqueda de la ED es especialmente sensible a las variables de inicio y la estrategia de variación. Price, Storn y Lampinen (2005); Bilal Pant (2022). Estos parámetros deben ajustarse para diferentes problemas de optimización porque generalmente dependen de la tarea a resolver. Zheng, Simpson y Zecchin (2013b). Por ello conviene en este sentido conseguir una exploración eficiente del espacio de búsqueda y su explotación efectiva. Debido a que no todos los operadores tienen el mismo impacto en la fase de búsqueda, encontrar el operador de variación y combinación que consigue en el mejor ajuste puede resultar problemático.

Aunque las propiedades de convergencia de la ED normalmente no dependen en gran medida de sus parámetros de control, A esto conviene contraponer la explicación de Plagianakos, Tasoulis y Vrahatis (2008), que señala si las tasas de combinación y variación son demasiado altas, el algoritmo explora el espacio de búsqueda de manera demasiado amplia. Esto puede resultar en la pérdida de buenas soluciones y en la dificultad para converger al mínimo global.

Respecto al número de iteraciones, se indica en la tabla 4 los resultados investigación de Mansouri y Mohamadizadeh (2017). Un número de iteración de 200 proporciona resultados aceptables para la ED.

Tabla 4. Efecto del número de iteraciones en optimización del costo.
Obtenido de Mansouri y Mohamadizadeh (2017).

Nro. Iteración	Costo Optimo (\$)	Tiempo de cálculo (s)
30	55293902	629
40	2065347	780
50	1222823	1005
100	786416	1950
200	737920	4024
300	737931	6164
500	737920	9324
1000	737924	20163
2000	737920	39826
3000	737920	53911

Chakraborty, (2008) recomienda usar como regla general las variables dadas por Storn y Price en una publicación del 1997* para los valores de F , C_r y N .

$$0,5 \leq F \leq 1$$

$$0,8 \leq C_r \leq 1$$

$$Np = 10 D \tag{2.15}$$

Donde D es el número de variables de decisión

[*] Storn, R., Price, K.V.: Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)

Teniendo en cuenta a Mansouri y Mohamadizadeh (2017) que examinaron 18 combinaciones diferentes para los coeficientes F y Cr. (Los resultados se encuentran en la Tabla 5). Encontrando que los valores medianos para dichos factores proporcionan la situación óptima y hacen que el algoritmo ED no llegue a la convergencia en un óptimo local. Estos valores coincidieron con los resultados de Suribabu (2010).

Tabla 5. Estudio de los factores F y CR.
Obtenido de Mansouri y Mohamadizadeh (2017)

Nro. De combinación	F	Cr	Costo Óptimo (\$)
1	0.1	0.1	115427393
2	0.5	0.3	832628
3	0.5	0.4	768561
4	0.5	0.5	758917
5	0.5	0.6	740000
6	0.6	0.3	738039
7	0.6	0.4	737931
8	0.6	0.5	737920
9	0.6	0.6	737992
10	0.7	0.3	737924
11	0.7	0.4	737988
12	0.7	0.5	740588
13	0.7	0.6	758028
14	0.8	0.3	786416
15	0.8	0.4	824850
16	0.8	0.5	832628
17	0.8	0.6	833455
18	1	1	55293902

Los parámetros apropiados varían para los diferentes problemas de optimización y normalmente se ajustan por ensayo y error. Por lo tanto, el uso efectivo del algoritmo ED requiere mucho tiempo porque se deben determinar los valores con el mejor ajuste para cada nuevo estudio de RDA. Zheng, Simpson y Zecchin (2013b) Debido a esto, las investigaciones para estrategias de autoadaptación de parámetros de control como Angira y

Babu (2005) desarrollaron una estrategia de ED autoadaptable, evitando así el ajuste por ensayo y error.

Es recomendable usar técnicas que empleen diferentes variables de control, debido a que la ED al igual que muchos AE están sujetos al sesgo estructural de preferencia. En donde el algoritmo conduce a una preferencia por regiones particulares en el espacio de búsqueda, independientemente de la función objetivo, Vermetten (2022). Debido a que es difícil de detectar, reconocer el tipo de sesgo involucrado y el componente que puede ser la causa de ello. Una buena manera de lidiar esta situación es emplear distintas técnicas de optimización y comparar sus efectos, junto con suficientes pruebas para los parámetros de control. De esta manera se puede contar de diferentes resultados para el desempeño del algoritmo en diferentes condiciones que permite encontrar patrones, características positivas o negativas para el análisis de la red y finalmente determinar el óptimo global de la RDA.

2.6.3 Conjunto de Soluciones

La ED opera en un conjunto de soluciones $P_{x,G}$ que contiene N soluciones candidatas sujetas a cambiar en cada iteración G . La matriz de posición del conjunto de soluciones de la iteración G se puede representar como:

$$P_{X,G} = (X_{i,G}), \quad i = 0,1,\dots,N-1, \quad G = 0,1,\dots,NG \quad (2.16)$$

$$X_{i,G} = (x_{i,G}^j), \quad j = 0,1,\dots,D-1. \quad (2.17)$$

El índice, $G = 0, 1, NG$, indica la iteración a la que un vector pertenece. En adición, a cada vector es asignado un índice, i , el cual corre desde 0 a $N-1$. Los vectores cuentan con parámetros que se indexan con j el cual corre desde 0 hasta $D-1$.

Para generar la conjunto de soluciones inicial, usualmente se escogen vectores al azar con parámetros entre los límites mínimos y máximos del problema, de modo que pueda cubrir todo el espacio de parámetros. También se puede usar el método de la

inicialización no aleatoria del conjunto de soluciones inicial, porque este es capaz de acelerar la convergencia a óptimos globales si es implementada correctamente.

Una vez inicializados los vectores de diámetros candidatos que servirán para explorar el espacio de búsqueda, se perturba aleatoriamente cada uno de los vectores para producir un conjunto de soluciones de vectores variados, $V_{i,G}$:

$$P_{V,G} = (V_{i,G}), \quad i = 0,1,\dots,N-1, \quad G = 0,1,\dots,NG \quad (2.18)$$

$$V_{i,G} = (v_{i,G}^j), \quad j = 0,1,\dots,D-1. \quad D \text{ indica la cantidad de parámetros.} \quad (2.19)$$

La manera en que se lleva a cabo la estrategia de variación es clave para el éxito de la ED, Plagianakos y otros, señalan que algunos operadores de variación tienen la tendencia a concentrar subconjuntos del conjunto de soluciones en la región de atracción de diferentes minimizadores de la función objetivo. En la figura 5 se ilustra esta tendencia según el número de iteraciones.

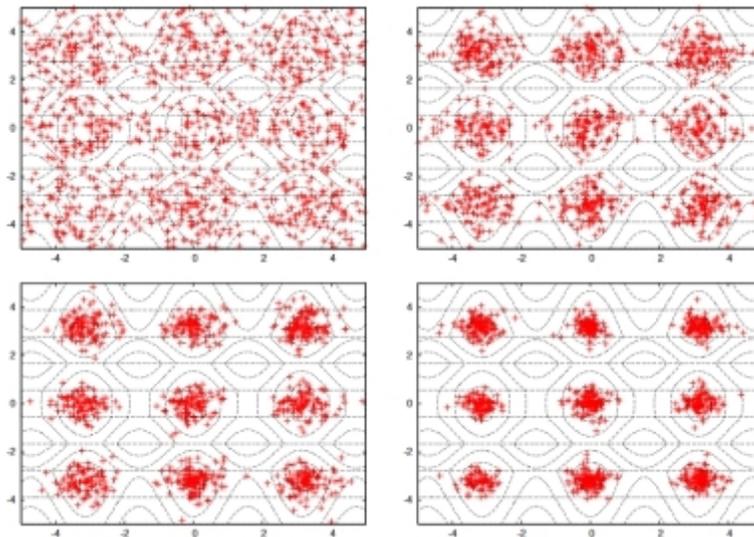


Figura 5. Conjunto de soluciones después de 1, 5, 10 y 20 iteraciones.

Obtenido de Plagianakos, Tasoulis y Vrahatis (2008)

En la etapa de combinación, el vector objetivo es recombinado con el vector de variación para producir el vector de prueba $U_{i,G}$ de N vectores y así generar la matriz de vectores de prueba P_U :

$$P_U = (U_{i,G}), \quad i = 0, 1, \dots, N-1, \quad G = 0, 1, \dots, NG \quad (2.20)$$

$$U_{i,G} = (u_{i,G}^j), \quad j=0, 1, \dots, D-1. \quad (2.21)$$

Finalmente, cada vector objetivo se somete a prueba hasta recorrer todo el conjunto de soluciones. En el siguiente diagrama (Figura 6) se puede ver en mayor detalle como interactúan las variables del algoritmo ED.

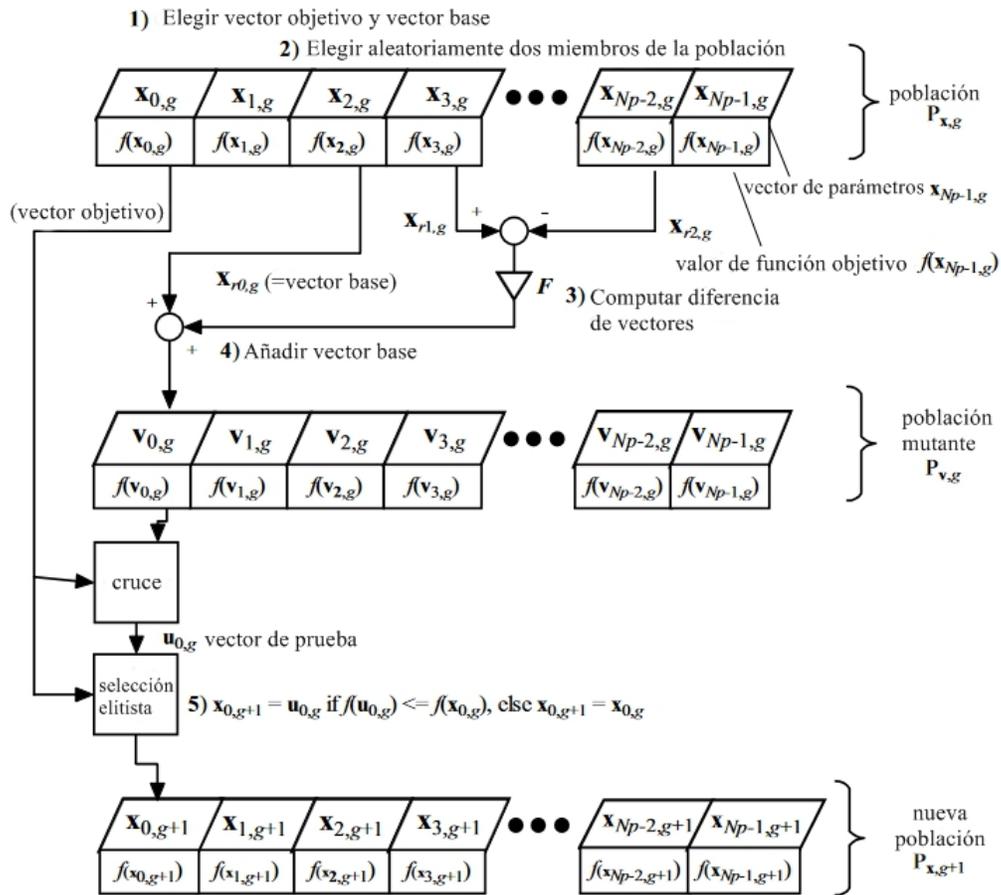


Figura 6. Diagrama para realizar ED.
Obtenido de Price, Storn y Lampinen (2005)

Para ir de una iteración a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, combinación y variación.

2.6.4 Variación

El operador de variación en la ED, se puede considerar como un método que añade diversidad al proceso de búsqueda, utilizando de manera aleatoria los vectores del conjunto de soluciones, que emplea los mejores vectores, los actuales o ambos (refiriéndose al vector objetivo), etc., para generar un nuevo vector. Esto permite que el algoritmo explore otras regiones del espacio de soluciones, posibilitando una mejora hacia un óptimo global. De esta manera se evita un estancamiento ocasionado por la repetición de vectores similares. Fister y otros (2022)

La ED admite muchas estrategias de variación que se aplican de acuerdo con el problema a resolver, reconocer cual estrategia de variación es la mejor para un problema particular es clave para el buen desempeño del algoritmo. Onwubolu y Davendra (2009) describen más a fondo las ventajas de emplear diferentes estrategias de variación para controlar la magnitud del cambio que esta puede provocar.

La estrategia de variación básica de la ED es simple pero poderosa, toma tres vectores X elegidos al azar del conjunto de soluciones, la diferencia entre X_{r1} y X_{r2} , define la dirección y la longitud del paso de búsqueda. La constante F , gestiona el balance entre la explotación y la exploración del espacio. El vector variación V se obtiene de sumar $F \cdot (X_{r1} - X_{r2})$ a un tercer vector X_{r3} , que sirve como punto base de aplicación o punto de referencia actual. Tal como se muestra en la figura 7.

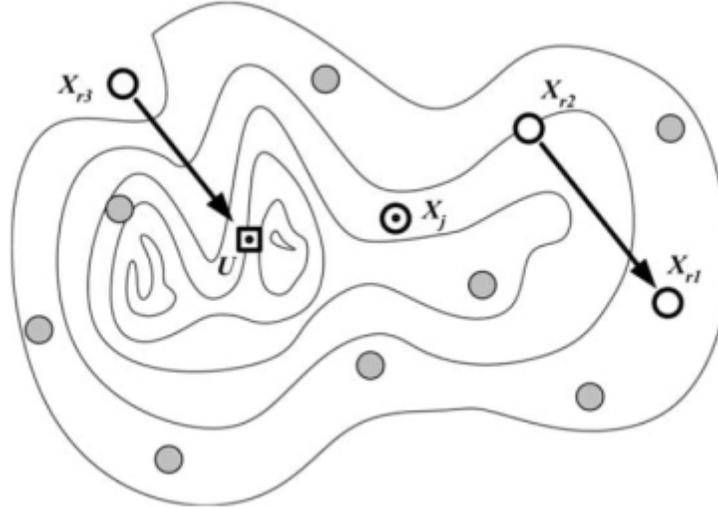


Figura 7. Creación del vector de variación V . Obtenido de Feoktitov (2006)

Debido a este proceso eventualmente la longitud entre el vector de aplicación y los pasos se va disminuyendo hasta alcanzar la convergencia. Al principio de la evolución, los vectores están muy dispersos en el espacio de búsqueda. A medida que la ED avanza, los vectores se acercan entre sí y la longitud del paso se reduce. La aleatoriedad de las direcciones de búsqueda y de los puntos base puede ayudar a encontrar el óptimo global, pero puede retrasar ligeramente la convergencia Feokstítov (2006).

A continuación, se muestran 6 estrategias clásicas de variación sugeridas por Fister y otros (2022), para emplear la estrategia que mejor se adapte en la optimización de RDA.

$$\text{Azar-1:} \quad V = X_{r1} + F(X_{r2} - X_{r3}) \quad (2.22)$$

$$\text{Azar-2:} \quad V = X_{r1} + F(X_{r2} - X_{r3}) + F(X_{r4} - X_{r5}) \quad (2.23)$$

$$\text{Mejor-1:} \quad V = X_{best} + F(X_{r1} - X_{r2}) \quad (2.24)$$

$$\text{Mejor-2:} \quad V = X_{best} + F(X_{r1} - X_{r2}) + F(X_{r3} - X_{r4}) \quad (2.25)$$

$$\text{Azar-a-mejor-1:} \quad V = X_{r1} + F(X_{best} - X_{r2}) + F(X_{r3} - X_{r4}) + F(X_{r5} - X_{r6}) \quad (2.26)$$

$$\text{Actual-a-mejor-1:} \quad V = X_i + F(X_{best} - x) + F(X_{r1} - X_{r2}) \quad (2.27)$$

En este mismo orden de ideas, el objetivo fundamental de las estrategias de ED es seleccionar la configuración de vector base y longitud de paso que proporcionen un avance

efectivo en la exploración del espacio de búsqueda durante el análisis de RDA. Tal como expresa Feokstítov, el éxito de la ED reside en la creación de los vectores de prueba, donde la etapa de variación también juega un factor determinante para la ED. Por ejemplo, al seleccionar la estrategia de variación clásica Azar-1 se obtienen en general buenos resultados, sin embargo, en el caso de RDA grandes con múltiples tuberías, es altamente probable que el nuevo vector no sea capaz de cumplir con las demandas hidráulicas, debido a que el diámetro de una sola tubería afecta a la distribución de toda la red causando la incapacidad de surtir la demanda de caudal. Por este motivo, para que el algoritmo de ED sea capaz de generar buenas propuestas, es preferible asegurar que los vectores de variación sean soluciones factibles, tratar de direccionar la variación hacia los mejores parámetros y de esa manera reducir la rata de vectores indeseados que podrían retrasar la convergencia y aumentar el esfuerzo computacional.

2.6.5 Combinación

Con el fin de incrementar la diversidad de los parámetros del vector variación, se introduce la operación de combinación, que tiene la función de copiar parámetros de una pareja de vectores (en este caso el vector objetivo y el vector variación) para crear el vector de diámetros de prueba $U_{i,G}$, que hereda las características de sus antecesores. En la figura 8 se muestra gráficamente este proceso.

En orden de prevenir el caso donde $u_{i,G} = x_{i,G}$, se ajusta Cr para controlar la fracción de parámetros que se copian del vector mutante y garantizar que al menos un componente de este se incluye en el vector de prueba $U_{i,G}$. Se utiliza un número aleatorio generado en el rango de cero a uno.

$$U_{i,G} = (u_{i,G}^j) = \begin{cases} v_{i,G}^j & \text{sí (aleatorio [0,1])} \leq Cr \\ x_{i,G}^j & \text{de otra manera.} \end{cases} \quad (2.28)$$

Como describe Chakraborty, se consideró que la variación por sí sola es demasiado restrictiva como método de perturbación, y requieren alguna variación aleatoria además del

mecanismo de variación dominante para que la optimización funcione correctamente. Por ello tanto la ED como los AG tienen una estrategia adicional de recombinación que distorsiona ligeramente las tendencias de la perturbación dominante.

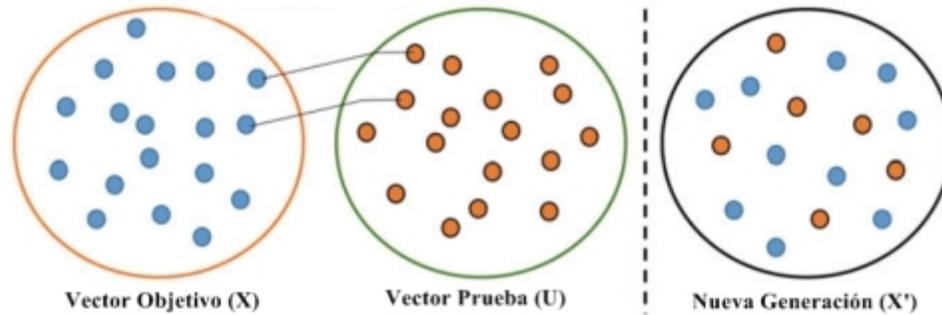


Figura 8. Creación del vector de prueba. Obtenido de Bilal Pant (2022).

2.6.6 Selección

La ED usa una selección elitista uno a uno donde el vector de prueba $U_{i,G}$, compete contra el vector objetivo $X_{i,G}$. Si el vector $U_{i,G}$ proporciona un costo menor de la función objetivo que $X_{i,G}$ este se conserva en la siguiente iteración, de otra manera $X_{i,G}$ se mantiene sin cambios. Este proceso se llama selección.

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{sí } f(u_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{de otra manera.} \end{cases} \quad (2.29)$$

Finalmente, este proceso continúa con cada vector del conjunto de soluciones. Por lo tanto, la aptitud del conjunto de soluciones está siempre mejorando o al menos manteniendo los mismos valores. Entonces el mismo proceso es repetido hasta alcanzar la condición de terminación.

2.7 Manejo de restricciones

A continuación, se presentan 3 métodos de penalización para el manejo de restricciones mencionados por Price, Storn y Lampinen (2005). Estos métodos permiten

rechazar las soluciones no factibles, y redirigir la búsqueda hacia mejores soluciones en el espacio de búsqueda.

2.7.1 Tipo pared de ladrillo

Si algún parámetro de prueba excede un límite, se establece el valor de la función objetivo del vector infractor a un número suficientemente alto capaz de garantizar que no será seleccionado. No obstante, cuando el óptimo se encuentra cerca de los límites, la penalización es capaz de retrasar el proceso optimización, ya que es más probable generar soluciones con parámetros fuera de los límites.

2.7.2 Reinicialización aleatoria

El método reemplaza un vector que ha excedido uno o más de sus límites hasta que un nuevo vector aleatorio pueda satisfacer todas las restricciones. La mayor ventaja de este método es la garantía que siempre se van a proponer soluciones factibles, aunque el tiempo de computación será bastante mayor.

2.7.3 Pivote

A diferencia de la reinicialización aleatoria, la estrategia de rebote tiene en cuenta el progreso hacia el óptimo al seleccionar un valor de parámetro que se encuentra entre el valor del parámetro base y el límite que se está violando. A medida que el conjunto de soluciones se mueve hacia sus límites, el método de pivote genera vectores que se ubicarán aún más cerca de los límites.

2.8 Estrategia para evitar reevaluar variables

Durante el proceso de optimización, los cambios entre estas fases de exploración y explotación ocurren a menudo y, en consecuencia, muy probablemente se vuelven a examinar vectores ya evaluados anteriormente. Esta situación tiene un gran impacto en el

tiempo de optimización del algoritmo, a menudo en las etapas finales del algoritmo sucede la ocurrencia de duplicados cuando el proceso está explorando alrededor del óptimo y se torna aún más laborioso cuando se llevan a cabo múltiples evaluaciones del proceso de optimización para encontrar el mejor ajuste de la red. Lo que puede conducir a la situación de que la mayor parte del tiempo de optimización se dedique a la evaluación de duplicados, Črepinšek (2022). Especialmente en el caso de RDA el costo de la evaluación para determinar si un vector es factible, naturalmente termina elevando el costo computacional de todo el proceso en gran medida.

Por lo expuesto anteriormente es recomendable evitar las evaluaciones innecesarias de duplicados. Črepinšek propone dos propuestas relativamente fáciles de emplear, la primera es aplicar un criterio de parada adicional cuando los duplicados aparecen con mucha frecuencia y la segunda es aplicar la reinicialización aleatoria en fases donde se generan muchos duplicados. La segunda opción permite un mayor abanico de opciones para afrontar el trabajo que no aporta beneficios, para aplicar esta estrategia básicamente se requiere crear una base de datos que guarde los distintos vectores generados durante el algoritmo, para posteriormente revisar si ya han sido evaluados y decidir que hacer a continuación. Asumiendo que se lleve a cabo la evaluación del vector duplicado, este últimamente termina siendo descartado porque no mejora el costo de la función objetivo, por este motivo una buena forma de manejar la aparición de vectores duplicados es descartarlos apenas son reconocidos y continuar con el proceso de búsqueda del algoritmo. Tampoco es estrictamente necesario generar alternativas distintas a los vectores duplicados ya que los AE naturalmente continúan con el proceso de búsqueda y evaluación para encontrar mejores resultados.

2.9 Descomposición de RDA

Se ha encontrado que se pueden separar las RDA en distintas subredes según la cantidad de reservorios de agua del sistema, y analizarlas individualmente con el propósito de mejorar el tiempo de análisis durante cada etapa del proceso de análisis de costo óptimo. Zheng y otros (2013) toman como referencia el algoritmo de descomposición propuesto por Deuerlein (2008), que usando RDA complejas descompuestas en nodos, bloques y puentes

utilizando la teoría de grafos, simplifica la RDA original en varias partes con el fin de mejorar la gestión de la red en su posterior análisis. Este enfoque ha sido estudiado para la descomposición de las redes de agua mediante el cálculo de la ruta de flujo más efectiva entre el nodo de demanda y la fuente de agua.

El enfoque de descomposición propuesto por H. M. Lee y otros (2016) que utiliza la técnica de rastreo de fuente de EPANET, permite la descomposición eficiente de una red real a gran escala, tomando en cuenta todas las tuberías de la RDA. Sin embargo, debido a que el programa de análisis hidráulico está programado en FORTRAN, se hará uso de este para analizar la RDA por el método de Zheng basado en la pendiente hidráulica para descomponer el grafo en subredes. En el enfoque propuesto por Zheng y otros, la RDA original se optimiza de acuerdo al diagrama de flujo en la figura 9:

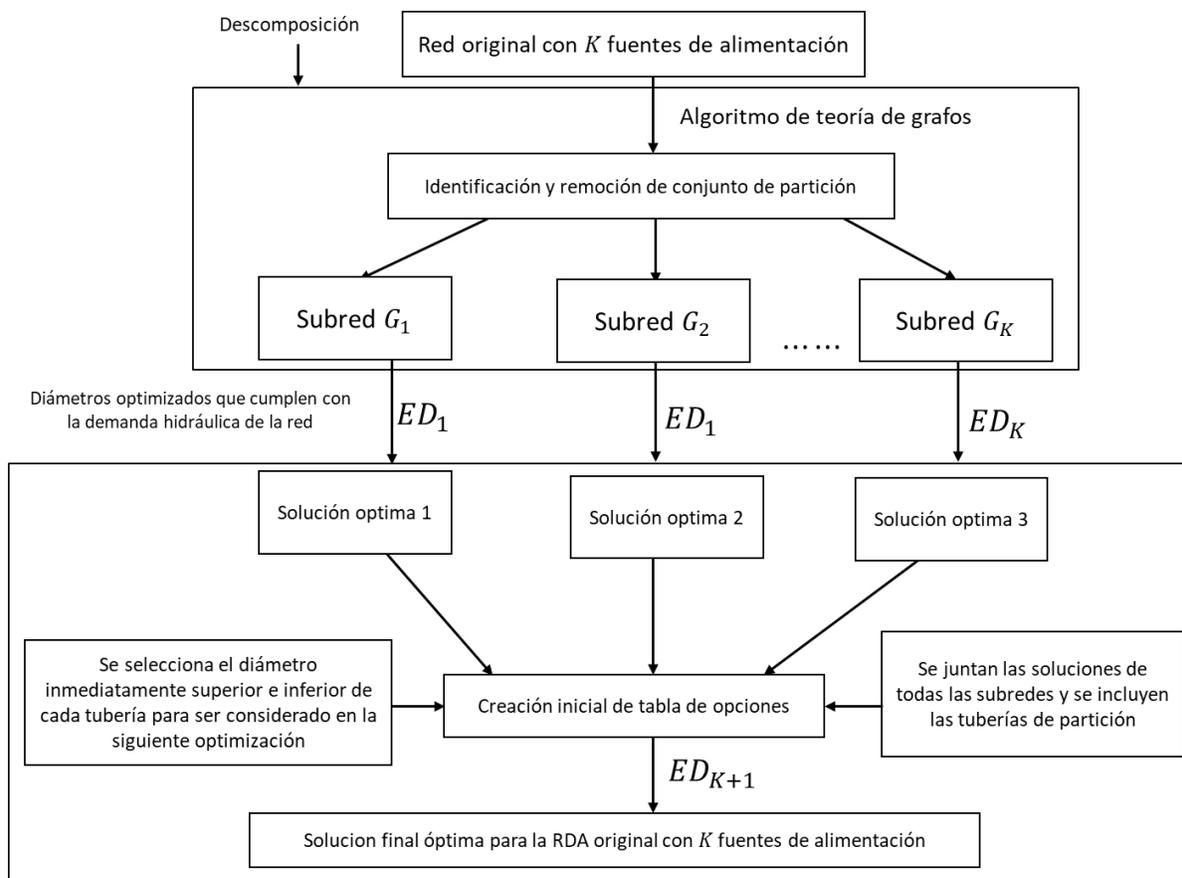


Figura 9. Diagrama para el diseño de costo óptimo de RDA.

Obtenido de Zheng, Zecchin y Simpson (2013)

Se desarrolla un algoritmo para identificar el conjunto de tuberías de partición óptimo para K reservorios de agua, empleando el concepto de pendiente hidráulica y algoritmo de Dijkstra. Al eliminar el conjunto de corte de partición, la RDA original se descompone en K subredes, cada una con una fuente de suministro. Luego, cada subdivisión se optimiza mediante ED de forma independiente, que es la primera etapa de optimización. Los diámetros óptimos para todas las subredes se combinan para generar una solución óptima aproximada de toda la red. Sin embargo, este resultado se debe mejorar debido a que las tuberías dentro del conjunto de partición no se toman en cuenta en la primera etapa de optimización. Por lo tanto en una segunda etapa, se vuelve a emplear la ED para analizar el espacio de búsqueda alrededor de la solución óptima proveniente de la primera etapa, esperando encontrar soluciones de menor costo para toda la red. La metodología de la descomposición propuesta y el algoritmo multietapa se detalla más adelante.

CAPÍTULO III

3. METODOLOGÍA IMPLEMENTADA

3.1 Diseño de la investigación

En este documento, se desarrolló un programa que obtiene los diámetros de tubería que producen un diseño de menor costo para una RDA efectiva empleando principalmente Evolución Diferencial. Considerando los siguientes aspectos.

- 1) Mejorar la calidad de los resultados obtenidos y el tiempo de computación, empleando optimización multietapa, que se basa en la descomposición de sistemas complejos y de gran escala en subsistemas independientes.
- 2) Aunque el algoritmo diferencial evolutivo se utiliza en este estudio, otros Algoritmos Evolutivos, como el Algoritmo Genético, también podrían implementarse en el marco de optimización propuesto. Por este motivo, se lleva a cabo la comparación del rendimiento de un AG estándar, con la ED.
- 3) Tener en cuenta la influencia que ejerce el ajuste de los distintos parámetros de cada técnica para el buen hacer de la misma y la velocidad de resolución del algoritmo.

El código del programa de computador ha sido escrito para emplear la ED y AG usando FORTRAN 95. El programa realiza un análisis en cada evaluación para determinar la carga de presión en los nodos. En caso de no cumplir con la presión requerida, se emplea la penalización por pared de ladrillo explicada en el apartado 2.7.1, incrementando el precio de un vector para evitar que sea seleccionado como una solución factible.

3.1.1 Generación de números aleatorios

Se consigue generando un nuevo valor dentro del rango permitido para dicha variable, con este fin una función adiciona el valor del límite inferior a la diferencia entre el límite inferior y superior más uno, multiplicado por un numero generado con una distribución uniforme entre cero y uno.

$$R = A + (B-A+1)*\text{aleatorio}(0-1) \quad (2.30)$$

3.1.2 Datos de entrada

En primer lugar, se da lectura a la red que consta de número de fuentes de alimentación (K), número de nodos (N_{poin}), número de tuberías que conforman la red (NT), altura geodésica de los nodos (Z), presión mínima requerida en cada nodo (PR), lista con costo de diámetros comerciales por metro ($DCOST$) y topología de la red en forma de matriz de adyacencia $G_{N,N}$ y las matrices de adyacencia de cada subred (G_k).

3.2 Casos de estudio

3.2.1 Primer caso

Consta de dos reservorios con 6 nodos y 6 tuberías. Los nodos 5 y 6 son fuentes de suministro. Las tuberías están fabricadas con fundición de hierro con una rugosidad de 0.30. La distribución se muestra en la figura 10.

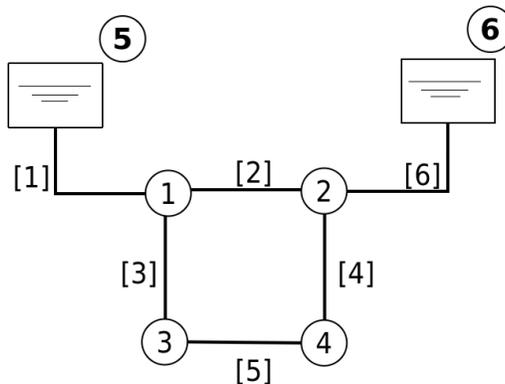


Figura 10. Grafo del primer caso de estudio. Obtenido de Zheng, Zecchin y Simpson (2013)

A continuación, se detallan en la tabla 6 los valores respecto a la elevación, presión requerida (PR), demanda de caudal en el nodo (\dot{Q}) y el largo de la tubería. Para esta red el espacio de búsqueda consta de $14^6 = 7529536$ posibles redes.

Tabla 6. Datos del primer caso de estudio. Obtenido de Zheng, Zecchin y Simpson (2013a)

i : Número de nodos	Z : Elevación (m)	PR : Carga de presión requerida (mca)	Q̇ : Demanda de caudal (L/s)	NT : Número de tuberías	L : Largo de la tubería (m)
1	27	20	50	1	800
2	29	20	60	2	800
3	31	20	75	3	650
4	33	20	90	4	700
5	54	-	-	5	550
6	56	-	-	6	400

Tabla 7. Lista de diámetros comerciales de Kadu, Gupta y Bhawe (2008).

Elaboración propia.

Diámetro (mm)	Costo Unitario (\$/m)
150	24,53
200	35,2
250	47,388
300	61,16
350	76,45
400	93,61
450	113,784
500	134,024
600	180,158
700	234,74
750	261,228
800	291,742
900	355,322
1000	426,69

3.2.2 Segundo caso

Una RDA de dos reservorios con 26 nodos y 34 tuberías usada anteriormente por Kadu y Zheng debido a su potencial para demostrar la utilidad del método propuesto. La distribución se muestra en la figura 10. Los nodos 1 y 2 son marcados como fuentes de suministro. En la tabla 8, se detallan los valores respecto a la elevación, presión requerida (PR), demanda de caudal en el nodo (\dot{Q}) y el largo de la tubería. Cuenta con 14 diámetros

comerciales de tuberías de fundición de hierro, con los mismos precios indicados en la tabla 7. Para esta red el espacio de búsqueda consta de $14^{34} = 7,48 \times 10^{37}$ posibles redes.

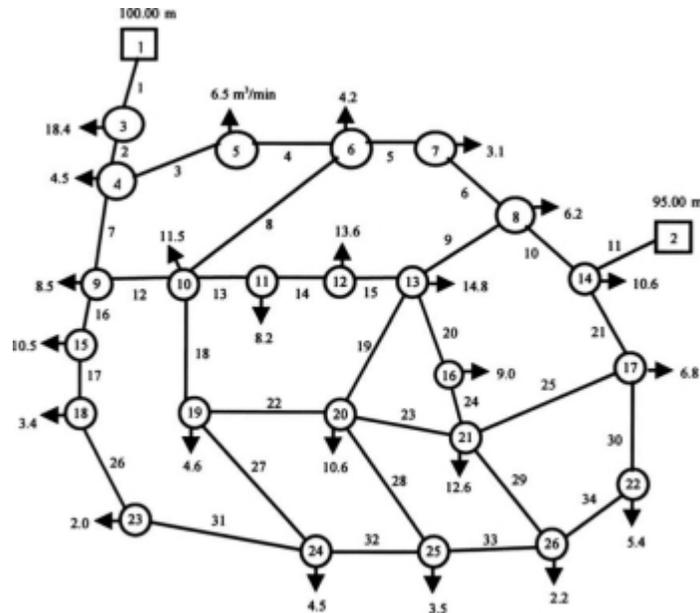


Figura 11. Grafo del segundo caso de estudio. Obtenido de Kadu, Gupta y Bhawe (2008)

Tabla 8. Datos para el segundo caso de estudio.
Obtenido de Kadu, Gupta y Bhawe (2008)

i: Número de nodos	Z: Elevación (m)	PR: Carga de presión requerida (mca)	Q̇: Demanda de caudal (m ³ /min)	NT: Número de tuberías	L : Largo de la tubería (m)
1	100	20	-	1	300
2	95	20	-	2	820
3	85	20	18.4	3	940
4	85	20	4.5	4	730
5	85	20	6.5	5	1620
6	85	20	4.2	6	600
7	82	20	3.1	7	800
8	82	20	6.2	8	1400
9	85	20	8.5	9	1175
10	85	20	11.5	10	750
11	85	20	8.2	11	210
12	85	20	13.6	12	700
13	82	20	14.8	13	310
14	82	20	10.6	14	500

15	85	20	10.5	15	1960
16	82	20	9.0	16	900
17	82	20	6.8	17	850
18	85	20	3.4	18	650
19	82	20	4.6	19	760
20	82	20	10.6	20	1100
21	82	20	12.6	21	660
22	80	20	5.4	22	1170
23	82	20	2.0	23	980
24	80	20	4.5	24	670
25	80	20	3.5	25	1080
26	80	20	2.2	26	750
				27	900
				28	650
				29	1540
				30	730
				31	1170
				32	1650
				33	1320
				34	3250

3.2.3 Tercer caso. Red de Pescara, Italia.

La red de Pescara corresponde a una versión reducida del sistema de distribución de agua de una ciudad en Italia. La red está compuesta de 99 tuberías, 71 nodos y 3 reservorios de agua, el grafo de la red se muestra en las figuras 12 y 13. Las tuberías son fabricadas de hierro fundido y el costo del metro por diámetro se muestra en la tabla 9. La presión se debe mantener en un rango de 20 a 55,9 mca y una velocidad máxima de 3 m/s. Para esta red el espacio de búsqueda consta de 13^{99} posibles casos.

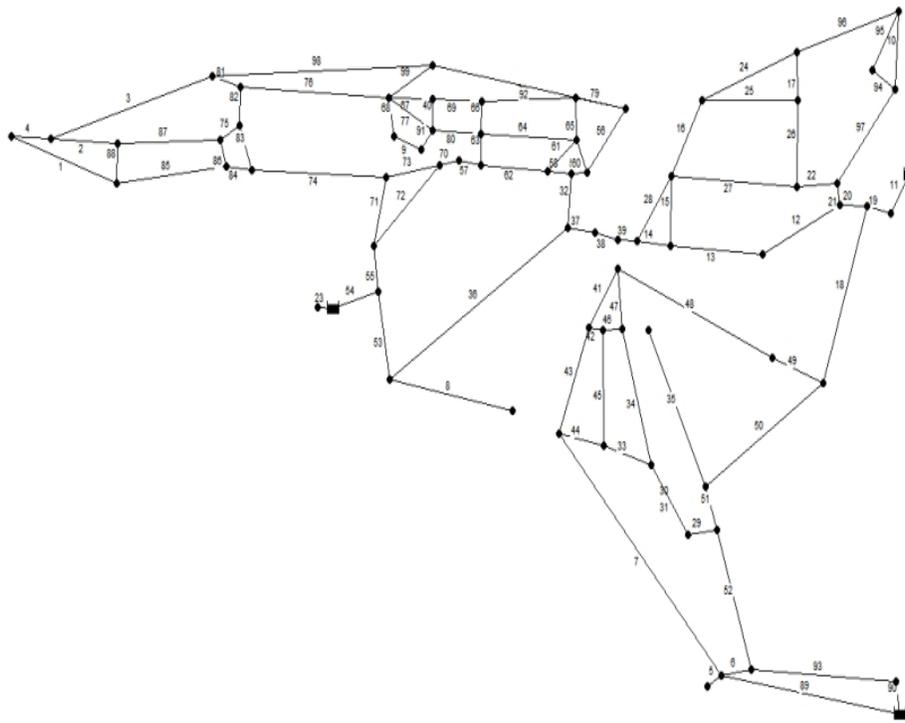


Figura 12. Red de Pescara, Italia. Elaboración propia.

Tabla 9. Lista de diámetros comerciales para la red de Pescara, Italia.

Obtenido de Saldarriaga y otros (2020)

Diámetro (mm)	Costo Unitario (€/m)
100	27,7
125	38
150	40.5
200	55.4
250	75
300	92.4
350	123.1
400	141.9
450	169.3
500	191.5
600	246
700	319.6
800	391.1

Tabla 10. Datos para la red de Pescara, Italia. Obtenido de Saldarriaga y otros (2020)

i: Número de nodos	Z: Elevación (m)	PR: Carga de presión requerida (mca)	Q̇: Demanda de caudal (L/S)	NT: Número de tuberías	L: Largo de la tubería (m)
1	2,9	20	10,01	1	977,36
2	5	20	12,43	2	443,17
3	3,5	20	8,5	3	1410,77
4	3,8	20	8,5	4	269,61
5	2,2	20	12,12	5	116,52
6	7	20	1,63	6	203,06
7	6,5	20	0	7	2309,14
8	5,2	20	33,6	8	1479,98
9	4,3	20	4,38	9	180
10	5,2	20	16,4	10	240
11	28	20	29,77	11	43,97
12	3,2	20	0,49	12	589,16
13	3,2	20	7,49	13	618,53
14	19,2	20	6,06	14	341,1
15	57	-	6,46	15	341,92
16	1,5	20	2,77	16	434,24
17	4	20	5,03	17	240,59
18	3,5	20	4,81	18	957,44
19	2,8	20	21,04	19	171,04
20	2,7	20	16,94	20	235,57
21	1,8	20	10,41	21	3,59
22	2,1	20	2,17	22	324,36
23	2	20	6,55	23	10
24	18,7	20	10,41	24	749,74
25	3,2	20	2,78	25	639,7
26	1,6	20	5,32	26	422,95
27	3,8	20	3,63	27	843,52
28	4,3	20	33	28	609,23
29	3,6	20	6	29	179,04
30	3,2	20	6,4	30	570,9
31	2,8	20	2,96	31	575,73
32	1,8	20	3,09	32	250,28
33	3,8	20	3,09	33	442,11
34	2,9	20	8,79	34	762,35
35	3,7	20	1,68	35	1124,9
36	2,1	20	12,64	36	1903,75
37	6,7	20	12,64	37	125,51
38	6,7	20	12,64	38	121,27

39	4,2	20	5,56	39	55,32
40	2,9	20	2,6	40	150,82
41	2,9	20	4,57	41	424,86
42	28,5	20	7,56	42	72,8
43	53,08	-	6,21	43	554,08
44	27,3	20	3,09	44	305,53
45	1,1	20	0,59	45	568,35
46	2,5	20	12,63	46	63,81
47	2,8	20	3,41	47	310,76
48	3,3	20	13,21	48	1217,17
49	1,8	20	3,87	49	365,82
50	1,8	20	6,43	50	1408,33
51	1,6	20	2,55	51	33,25
52	1,8	20	0	52	720,9
53	1,6	20	11,3	53	441,69
54	3,3	20	3,99	54	332,99
55	2,5	20	7,75	55	229,85
56	2,8	20	1,91	56	500,92
57	3,2	20	0,88	57	232,07
58	3,6	20	5,14	58	5,43
59	3,8	20	5,14	59	66,24
60	3,1	20	7,44	60	232,13
61	3	20	0,37	61	154,78
62	4,2	20	0,54	62	645,1
63	2,1	20	0	63	157,87
64	3,3	20	0,01	64	645,79
65	55	-	0	65	206,69
66	2,1	20	25	66	156,75
67	2,1	20	2,15	67	179,54
68	4,2	20	1,75	68	152,81
69	19,2	20	10,01	69	331,4
70	2,1	20	12,43	70	52,37
71	1,5	20	8,5	71	394,67
				72	774,69
				73	435,58
				74	899,67
				75	198,5
				76	1116,15
				77	332,35
				78	631,8
				79	350,97
				80	325,03

81	226,03
82	186,77
83	337,24
84	173,09
85	784,45
86	86,39
87	732,93
88	193,78
89	1744,27
90	154,18
91	113,35

3.2.4 Cuarto caso. Red de Balerma, España.



Figura 13. Red de Balerma, España. Elaboración propia.

La red de riego de Balerna suministra agua para cultivos en Almería, España. Está compuesta por 454 tuberías, 447 nodos de demanda y 4 embalses. La presión mínima permitida es de 20 mca, y las tuberías disponibles comercialmente tienen un rango de diámetro de 125 a 600mm. El material de las tuberías es PVC con una rugosidad de 0.0025mm. El espacio de búsqueda para el diseño de la red es de 10^{454} . Los parámetros de la red, como el largo de las tuberías, el caudal de demanda, tabla de precios y la topología, fueron obtenidos de Saldarriaga y otros (2020).

3.3 Procedimiento para optimizar RDA

3.3.1 Descomposición de RDA

Se genera la matriz $RECOR(K)(Npoin)$ para guardar la distancia más corta hasta cada nodo, donde K es el índice del respectivo reservorio de agua y $Npoin$ señala el nodo respectivo.

Luego es calculada la pendiente de hidráulica $P_{h(K,Npoin)}$ donde K es el número del reservorio y el segundo índice i representa el valor de $P_{h(k,i)}$ desde el reservorio K hasta el nodo i . Para cada reservorio se compara el valor de Ph en cada nodo, y aquel con la mayor pendiente hidráulica pertenece a dicho reservorio de agua, de esta manera se puede dividir la RDA original en K subredes independientes con una sola fuente de alimentación.

$$P_{h(k,i)} = \frac{H_k - H_i^{min}}{\sum L} \quad (2.31)$$

El grafo y los subgrafos se leen a partir de matrices de adyacencia de dimensiones $N \times N$, que sirven para representar las conexiones entre nodos y tuberías mediante unos y ceros.

$$\begin{bmatrix} 0 & 800 & 650 & 0 & 800 & 0 \\ 800 & 0 & 0 & 700 & 0 & 400 \\ 650 & 0 & 0 & 550 & 0 & 0 \\ 0 & 700 & 550 & 0 & 0 & 0 \\ 800 & 0 & 0 & 0 & 0 & 0 \\ 0 & 400 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figura 14. Matriz de adyacencia. Elaboración propia.

En el caso de las tuberías se puede conocer el largo de una tubería y nodos a los que está conectada mediante la matriz de adyacencia. La posición de cada fila indica una conexión con un determinado nodo, la primera fila es el primer nodo; la segunda fila es el segundo nodo y así hasta el último nodo de la red, lo mismo se aplica para las columnas; como consecuencia la matriz tiene la propiedad de ser simétrica.

3.3.2 Esquema del algoritmo de Dijkstra

Finalmente, el algoritmo Dijkstra se ejecuta una vez por cada reservorio de agua para encontrar el camino más corto desde cada fuente de suministro al resto de los nodos. La figura 15 ilustra el diagrama de flujo del algoritmo.

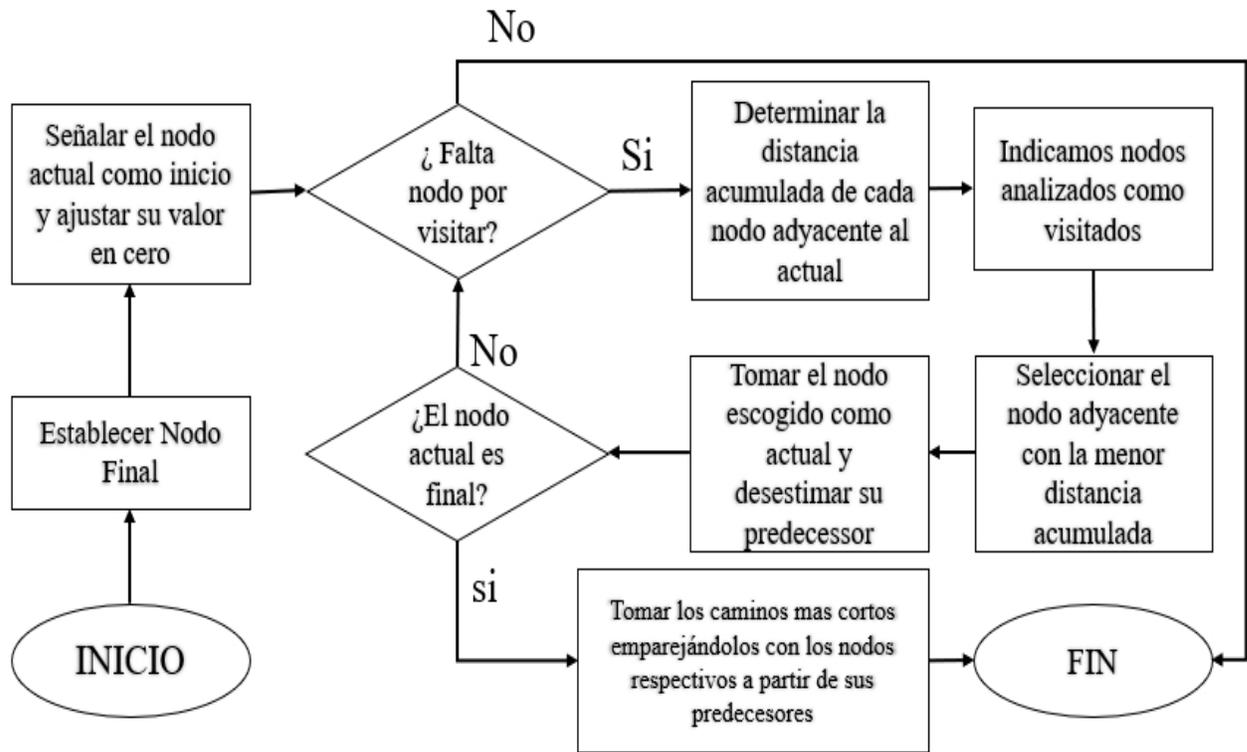


Figura 15. Diagrama de flujo del algoritmo de Dijkstra. Elaboración propia.

3.3.3 Descomposición mediante pendiente hidráulica

Se calcula la pendiente hidráulica desde cada reservorio hasta el resto de los nodos para identificar los nodos que pertenecen a cada subred. Como ya se conoce la ruta más corta para cada fuente de suministro K a un nodo i dentro de la RDA, entonces no hay necesidad de considerar otras rutas.

Tabla 11. Cálculo de la pendiente hidráulica. Elaboración propia.

Rutas de flujo	Longitud (m)	Diferencia de carga (m) $H_k - H_i^{\min}$	Pendiente hidráulica (adimensional)
5-1	800	54-47	0.0088
6-2-1	1200	56-47	0.0075
5-1-2	1600	54-49	0.0031
6-2	400	56-49	0.0175
5-1-3	1450	54-51	0.0021
6-2-4-3	1650	56-51	0.0030
5-1-2-4	2000	54-53	0.0005
6-2-4	1100	56-53	0.0027

Luego de encontrar los nodos que pertenecen a cada subred, una subrutina se encarga de identificar las tuberías de partición evaluando los extremos de cada tubería. Si ambos extremos de la tubería pertenecen al mismo reservorio, entonces dicha conexión también es perteneciente a la subred donde se encuentra. Si ambos nodos son distintos, entonces se asigna al conjunto de partición óptimo Ω , debido a que ambos extremos pertenecen a subredes distintas. Las subredes se obtienen después de la eliminación del conjunto de corte. Estas subredes son independientes y pueden optimizarse por separado. Para el primer caso de estudio el conjunto Ω consta de las tuberías 2 y 3.

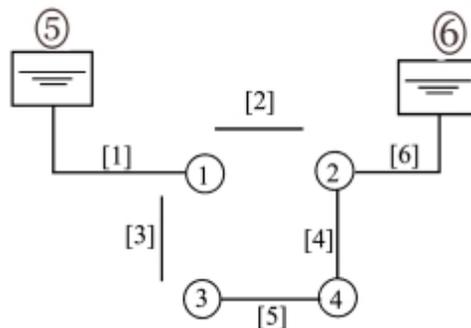


Figura 16. Descomposición del primer caso de estudio.

Obtenido de Zheng, Zecchin y Simpson (2013)

3.3.4 Esquema del programa de partición de redes

Los detalles del algoritmo propuesto para identificar el conjunto de corte de particionamiento óptimo Ω para una RDA con $G_{N,N}$, de NT tuberías, en K subredes G_k se indican en el anexo 2.

3.4 Estrategia de exploración

Es recomendable elaborar múltiples ensayos para los coeficientes F y Cr , debido a que un ajuste óptimo para estos parámetros proporciona el mejor costo para el estudio de la RDA. De esta manera se propone una rutina que examine diferentes combinaciones de los parámetros de control automáticamente, y de esa manera garantizar el uso efectivo del algoritmo.

Básicamente la estrategia consiste en ejecutar 16 combinaciones de los parámetros de control durante el proceso de optimización de la red. Tal como se ilustra en el anexo 3.

3.5 Detección de redundancia

Existen diferentes métodos para evitar reevaluar parámetros. Uno de los métodos más sencillos es mantener un registro de los parámetros que han sido evaluados en anteriormente. Este registro se puede utilizar para verificar si un parámetro han sido analizado antes de realizar la evaluación actual.

El número de configuraciones posibles para la red de acueductos depende del número de tuberías y diámetros comerciales disponibles. Tomando en cuenta que el diámetro de cada tubería se puede repetir y es necesario considerar el orden de las configuraciones. El número de formas en que se puede extraer un multiconjunto con k elementos de un conjunto con n elementos corresponde al número de k -permutaciones con repetición tomadas de un conjunto con n elementos. Por lo que el número total de permutaciones de una red es n^k .

La dimensión máxima que puede conseguir la base de datos se obtiene con la ecuación 12. Es necesario calcular la dimensión de la base de datos para definirla como una variable en FORTRAN.

Empleando la detección de redundancia se evita la duplicación de esfuerzos y se garantiza que la evaluación sea efectiva. Esta operación se debe tener en cuenta durante la creación de lal conjunto de soluciones inicial, ya que son evaluados los vectores para determinar si estos cumplen con la demanda hidráulica, así como en el paso de la selección elitista.

Antes de realizar el análisis el programa verifica si el vector a analizar ya se encuentra en la base de datos, y de ser así, se evita el análisis y se devuelve la información requerida para el proceso de optimización. Si es una nueva propuesta entonces se procede con el análisis y almacenado de la nueva información. La estructura de la base de datos se especifica en el anexo 4.

3.6 Algoritmo ED de primera etapa

La primera etapa del proceso de optimización inicia generando el conjunto de soluciones de vectores solución P_G . Para ello son seleccionados para cada tubería uno de los diámetros comerciales disponibles.

Teniendo en cuenta además las restricciones del problema, cada vector X_i debe cumplir con las demandas de carga de presión en los nodos. Por este motivo se utiliza la estrategia de recreación de vectores para generar todos los vectores que conforman el conjunto de soluciones candidatas P_g . En la figura 17 se describe el diagrama de flujo para el algoritmo ED.

El resto del proceso de optimización es el mismo descrito para el algoritmo clásico de Evolución Diferencial, con pocas diferencias. En el anexo 5 se detalla el algoritmo.

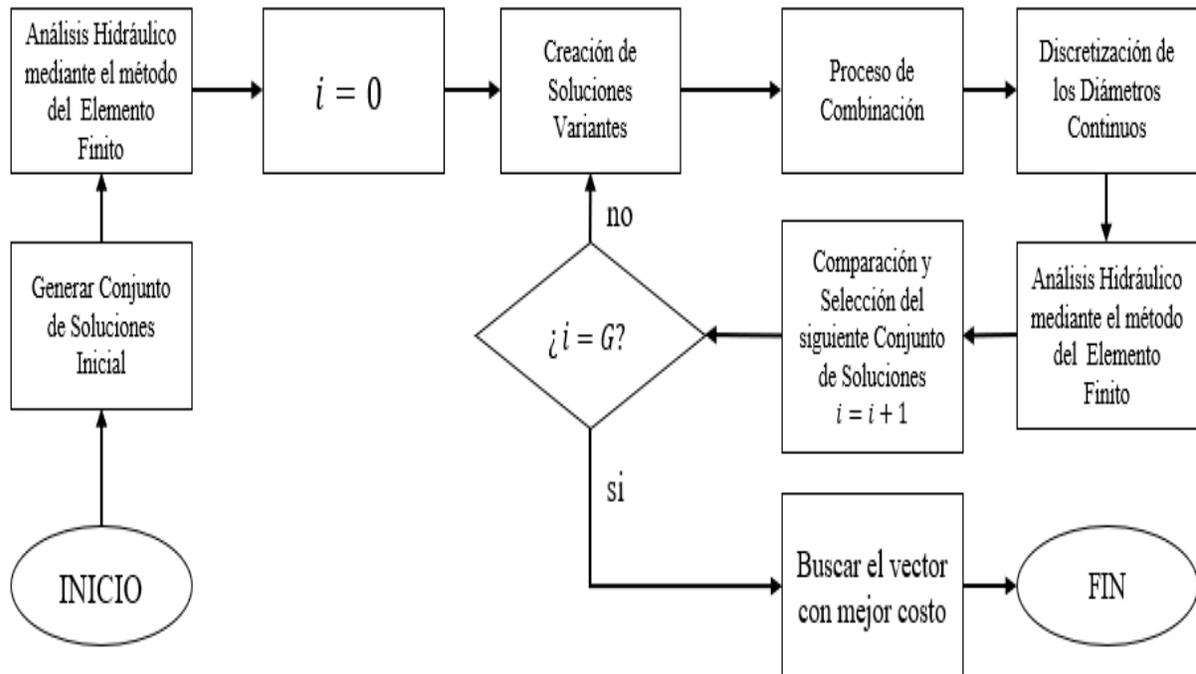


Figura 17. Diagrama de flujo del Algoritmo de Evolución Diferencial. Elaboración propia.

3.7 Algoritmo ED de segunda etapa

Con el diámetro de tubería óptimo en todas las subredes, se procede ahora a elaborar una tabla de opciones que se usa como espacio de búsqueda en la segunda etapa de optimización. Se juntan los resultados de la optimización de primera etapa y las tuberías de partición.

A cada enlace de la red completa se le da un diámetro de tubería combinando las soluciones óptimas de las subredes y el conjunto de tuberías de partición Ω , a las tuberías de partición se les asigna el menor diámetro disponible y los dos subsiguientes como opciones a tomar en cuenta, para el resto de tuberías se toman en consideración los diámetros inmediatamente inferiores y superiores. Esto, por lo tanto, crea una solución aproximada para la red completa con menor esfuerzo computacional. El resto del proceso de optimización es el mismo descrito que en la pagina anterior.

3.8 Procedimiento para optimizar RDA con Algoritmo Genético

El algoritmo se describe en detalle en el anexo 6.

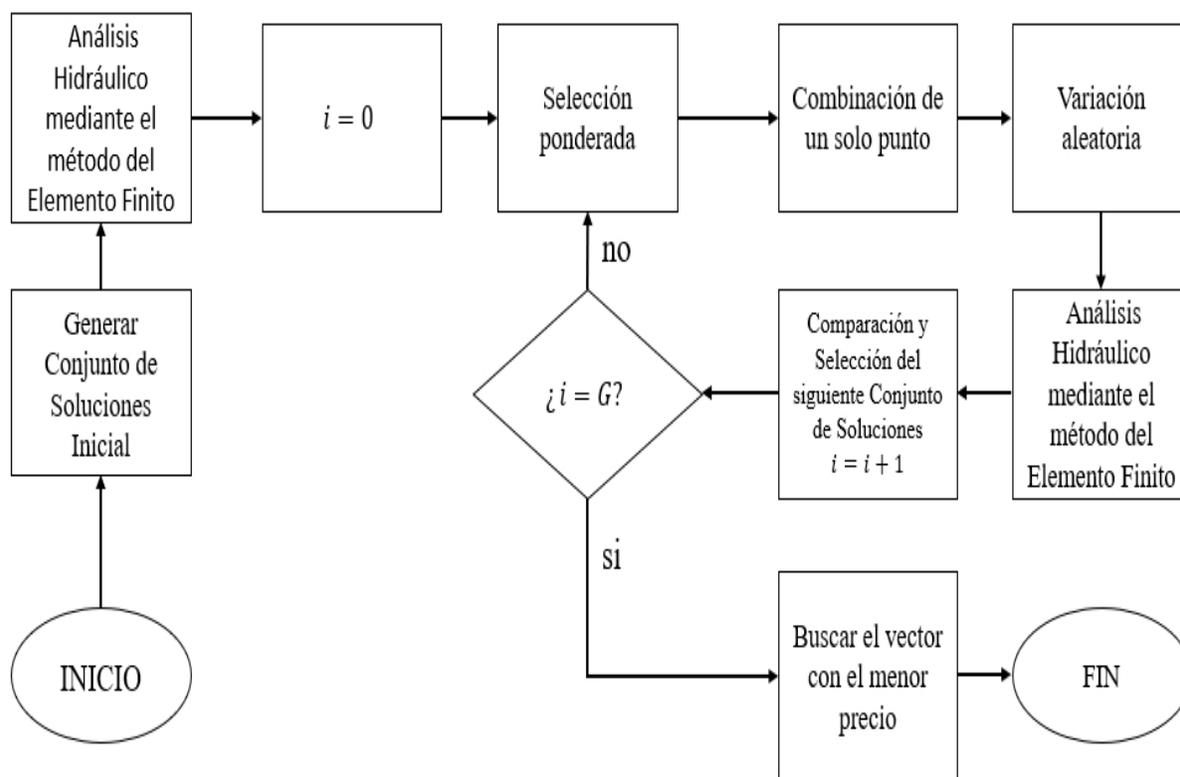


Figura 18. Diagrama de flujo del Algoritmo Genético. Elaboración propia.

3.9 Análisis de datos

Básicamente, el conocimiento sobre el comportamiento de un AE aplicado al diseño de RDA consiste en seguir ciertas pautas durante el modelo de optimización y tratar de obtener la mejor solución posible mediante un método que podría calificarse como de prueba y error. Así la mayoría de trabajos que pueden encontrarse en la literatura, usualmente no profundizan en otros aspectos de interés como la influencia de los parámetros de control, la repetitividad durante el análisis de costo óptimo, el costo computacional requerido para alcanzarlo, la configuración adecuada que deben tener los operadores asociados a la técnica de optimización, comparar distintos algoritmos de optimización con la técnica desarrollada, presentar alternativas y resultados que no sean

simplemente la obtención de un diseño de costo mínimo. De esta manera se busca conocer el comportamiento de las técnicas y sus variables, que permitan extrapolar los resultados a otros casos de diseño que puedan plantearse. Para este fin, se obtiene la siguiente información del programa para su posterior análisis:

- Crear el marco de datos con los resultados del proceso de optimización de la siguiente manera:
 - Tabla con los resultados del mejor ajuste obtenido por la ED según el parámetro empleado.
 - Tabla con información del tiempo de computación, eficiencia, precio y técnica utilizada.
 - Calcular el porcentaje de eficiencia del algoritmo.

3.9.1 Porcentaje de eficiencia

El índice de eficiencia es una medida de la calidad de una solución en relación con los recursos empleados para obtenerla. Se calcula como el número de buenas soluciones dividido por el número total de soluciones generadas.

$$\eta_{eficiencia} = \frac{\text{nro de soluciones factibles}}{\text{nro de soluciones generadas}} \quad (2.32)$$

Un índice de eficiencia más alto indica una mejor relación entre calidad y recursos. Este índice se puede utilizar para comparar diferentes algoritmos de optimización en base a distintos criterios. En cada red de distribución, la mejor combinación de parámetros para un algoritmo tendrá un índice de eficiencia de 1.

CAPÍTULO IV

4. RESULTADOS

A lo largo de este capítulo se determinó el mejor ajuste de parámetros para cada una de las técnicas de diseño planteadas en base a distintos criterios. Este sub apartado muestra los principales resultados obtenidos para cada uno de los casos simulados.

Se tomó en cuenta que para tener un adecuado diámetro en las tuberías el flujo de agua en estas debe tener una velocidad máxima de 3 m/s y una presión entre 10 y 50 mca. Los cuales se cumplen internamente en el algoritmo.

Después unas pruebas preliminares se determinó que la Ecuación 2.26 es más adecuada para explorar el espacio de búsqueda de redes de acueductos en comparación a la estrategia de variación aleatoria clásica (ecuación 2.24).

4.1 Primer caso de estudio

Los algoritmos de optimización tienen parámetros que pueden ajustarse para mejorar su rendimiento. Una configuración adecuada de estos parámetros puede reducir la aleatoriedad de los algoritmos evolutivos, lo que aumenta su probabilidad de encontrar la solución óptima. En primera instancia estos parámetros para la red son desconocidos, para ello es necesario un alto número de simulaciones con la finalidad de encontrar los valores adecuados para explorar el área de búsqueda a la vez que se disminuye el tiempo de cálculo, sin embargo esto último no siempre se puede lograr ya que una mayor exploración requiere una cantidad superior de análisis a soluciones encontradas, y esto conlleva un tiempo de ejecución más elevado.

La influencia de los distintos parámetros de ajuste en la velocidad de cálculo de cada metodología puede seguirse con detalle en la tabla 12.

Tabla 12 - Mejor ajuste con ED, primer caso. Elaboración propia.

Nro. De combinación	F	Cr	Costo Optimo promedio (\$)	Eficiencia %	Tiempo (s)
1	0.2	0.3	222093,30	35,13	4,2777
2		0.4	226279,77	30,83	3,1368
3		0.5	222543,11	22,67	3,0451
4		0.6	222543,11	28,13	2,6182
5	0.3	0.3	222169,86	32,57	6,3656
6		0.4	222137,55	28,14	5,2435
7		0.5	222825,73	28,64	6,3244
8		0.6	222086,70	33,88	3,9177
9	0.5	0.3	222128,31	28,36	4,5536
10		0.4	222102,10	28,87	6,8374
11		0.5	222247,48	30,41	8,1222
12		0.6	222091,10	34,27	9,3534
13	0.8	0.3	222088,90	32,81	17,4605
14		0.4	222218,13	33,02	24,7521
15		0.5	222155,56	30,49	15,3961
16		0.6	222232,01	28,46	18,5417

Ella muestra los resultados obtenidos al buscar el mejor ajuste del primer caso de estudio, que se representa en la Figura 10. Al observar el menor costo promedio, determinamos el ajuste más conveniente. Este dato comienza a obtenerse luego de la primera etapa donde se seleccionan las subredes óptimas. Esto indica que la zona de búsqueda más reducida y con una rápida convergencia es la combinación número 8, ya que los costos obtenidos en la red promedian resultados más bajos.

Para el análisis del algoritmo genético simple la configuración numero 1 de la tabla 13 fue la que produjo los mejores resultados en términos de tiempo y costo promedio. Esto se debe a que el algoritmo permite cierta aleatoriedad, lo que ayuda a evitar la excesiva generación de vectores inviables. Sin embargo, aumentar este factor puede provocar una mayor exploración de la diversidad de vectores, lo que conduce a un aumento del tiempo de cálculo y una disminución de la eficiencia del proceso.

Tabla 13 - Mejor ajuste con AG, primer caso. Elaboración propia.

Nro. De combinación	F	Costo(\$)	Eficiencia %	Tiempo (s)
1	0.1	222127,18	33,64	8,9003
2	0.2	232035,27	34,11	19,8091
3	0.3	222190,74	32,81	34,4293
4	0.4	222192,30	29,26	63,0608
5	0.5	222542,21	29,97	59,6521
6	0.6	222417,74	28,45	102,8230
7	0.7	222324,30	28,18	130,5237

Una vez obtenidos las mejores combinaciones de ajuste en cada uno de los algoritmos se procede a comparar los métodos estudiados con los parámetros obtenidos, así la Tabla 14 se muestra los resultados. El subíndice Ω indica que el algoritmo emplea la estrategia de subdivisión y base de datos, el subíndice S se refiere a que el algoritmo empleado no emplea la partición de redes y la base de datos.

Tabla 14 - Tiempo y costo de optimización según la técnica utilizada, primer caso.

Elaboración propia.

Técnica utilizada	Costo (\$)	Tiempo total (seg)
DE_{Ω}	222084,50	2,86
DE_S	241717,00	52,72
AG_{Ω}	222084,50	8,77
AG_S	222524,50	72,27

La estrategia con partición de redes obtuvo un 8,84% de reducción de costo en comparación con la estrategia sin partición. El algoritmo genético obtuvo un 0,198% de reducción de costo en comparación con la estrategia sin partición.

La partición de redes dota al código de una notable ventaja en comparación con las técnicas que no las utilizan, los métodos con ausencia de este recurso comienzan con costos elevados antes de converger en un mínimo local, esta primera etapa permite el descarte de una gran cantidad de soluciones que no son factibles, la tabla de elección usa la

información obtenida para disminuir el espacio de búsqueda tomando en cuenta las soluciones optimas de cada subred.

La Evolución Diferencial (ED) usando partición de redes y una base de datos fue la técnica que obtuvo los mejores resultados en cuanto a precio y tiempo. En el gráfico de la Figura 19 se muestra el desarrollo de cada método.

Los algoritmos obtuvieron o un costo similar, ya que la red particionada tiene un número menor de tuberías a considerar. Esto facilitó la generación de vectores de prueba y redujo el tiempo total de ejecución. Además, por la simplicidad del problema la repetitividad de las soluciones ocurre con mucha frecuencia y la base de datos se encarga de evitar analizar estos casos duplicados, por ello los tiempos de las técnicas que cuentan con la base de datos son muy inferiores en comparación a sus homólogos simples.

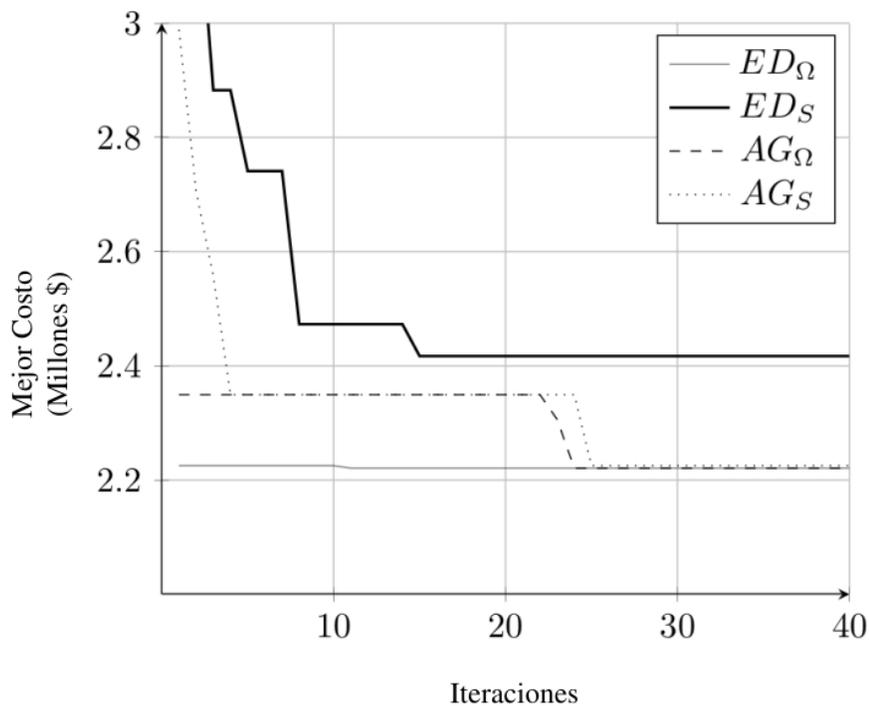


Figura 19. Costo vs. número de iteraciones, primer caso.

Elaboración propia.

El costo obtenido usando evolución diferencial es de 222084,50 dólares. Por otro lado, Zheng, Zecchin y Simpson lograron un valor de 239034 dólares, lo que representa una diferencia del 7,1% en comparación con nuestra investigación.

Luego de los análisis realizados los resultados obtenidos para el primer caso de estudio son los que se muestran en la Tabla 15.

Tabla 15 - Diámetros y presiones de la red optimizada, primer caso. Elaboración propia.

Tuberías	Diámetro optimizado (mm)	Nodos	Presión obtenida (m)	Presión mínima requerida (m)
1	150	1	49,13	47,00
2	150	2	55,04	49,00
3	150	3	51,40	51,00
4	450	4	53,63	53,00
5	300	5	54,00	54,00
6	500	6	56,00	56,00

Se denota los requerimientos de la red son cumplidos sin mayores dificultades; el algoritmo Diferencial Evolutivo fue ejecutado un total de 50 veces para garantizar el óptimo global de la misma, el precio mostrado en la Tabla 16 fue obtenido en 45 ejecuciones.

4.2 Segundo caso de estudio

El estudio se realizó utilizando la misma metodología que el caso anterior, con la diferencia de que esta vez se centró en el costo óptimo individual de cada configuración de ajuste, en lugar del costo óptimo promedio debido al aumento en la complejidad de la red. Los resultados de los algoritmos mostraron que el ajuste número 5 de la tabla 16 obtuvo el menor costo.

Tabla 16 - Mejor ajuste con ED, segundo caso. Elaboración propia.

Nro. De combinación	F	Cr	Costo Optimo (\$)	Eficiencia %	Tiempo (s)
1	0.2	0.3	2836734,77	47,51	692,1883
2		0.4	2653885,64	46,62	709,3041
3		0.5	2934211,89	51,90	384,9660
4		0.6	2798936,95	53,45	233,7895
5	0.3	0.3	2621237,33	36,73	911,6245
6		0.4	2642994,55	33,74	850,1825
7		0.5	2726872,50	38,99	529,4544
8		0.6	2707992,86	45,26	414,8731
9	0.5	0.3	2681199,53	21,98	1314,4316
10		0.4	2631746,39	30,15	1137,7920
11		0.5	2672030,07	16,42	1013,2292
12		0.6	2637242,26	22,01	880,4953
13	0.8	0.3	2813124,58	15,69	1358,8543
14		0.4	2794598,39	13,55	1335,1486
15		0.5	2866371,03	8,50	1374,8569
16		0.6	2950604,12	4,80	1406,9075

El mejor ajuste de parámetros para la red se obtiene con $F = 0.3$ y $Cr = 0.3$

Al igual que en el caso anterior, los resultados de los distintos ajustes del algoritmo genético simple arrojaron la misma configuración de parámetros, ver Tabla 17.

Tabla 17 - Mejor ajuste con AG, segundo caso. Elaboración propia.

Nro. De combinación	F	Costo Optimo Promedio (\$)	Eficiencia %	Tiempo (s)
1	0.1	2827831,49	11,195	1197,704
2	0.2	2843545,94	6,715	1727,494
3	0.3	2917908,12	4,745	2027,189
4	0.4	3023668,08	2,347	1893,489
5	0.5	3104196,47	1,254	1946,641
6	0.6	3130415,99	0,668	1967,325
7	0.7	3380435,32	0,317	1983,330

Luego de ejecutar cada una de las técnicas desarrolladas con los distintos recursos desarrollados a lo largo de la investigación, se presentan los resultados de estas ejecuciones en la Tabla 18. El subíndice Ω indica que el algoritmo emplea la estrategia de subdivisión y base de datos, el subíndice S se refiere a que el algoritmo no emplea la partición de redes y la base de datos.

Tabla 18 - Tiempo y costo de optimización según la técnica utilizada, segundo caso.

Elaboración propia.

Técnica utilizada	Costo (\$)	Tiempo total (seg)
ED_{Ω}	2599390,61	1260,34
ED_S	2798912,21	789,37
AG_{Ω}	2716147,72	1065,39
AG_S	2863270,88	711,89

El algoritmo de evolución diferencial obtuvo el costo más bajo, a pesar de tener un tiempo de cálculo mayor que los demás algoritmos. Lo que podría deberse a una mala iteración inicial. Sin embargo, fue capaz de encontrar un precio menor que las otras técnicas. La aleatoriedad inicial del algoritmo puede causar este fenómeno en cualquiera de las técnicas evaluadas en la investigación. La partición de redes tiene un significativo impacto en los tiempos, puesto la cantidad de optimizaciones es mayor ya que debe minimizar los costos en cada una de las subdivisiones de la red antes de realizarlo con la red completa.

El proceso de variación que debe realizar el algoritmo de evolución diferencial es de mayor complejidad que aquel que ejecuta el algoritmo genético simple, por este motivo posee tiempos mayores. La ED con partición de redes obtuvo un 4,70% de reducción de costo en contraste al resultado reportado por Zheng, Zecchin y Simpson, este fue la mejor solución que el algoritmo pudo encontrar luego de 50 iteraciones. En cambio el AG con partición de redes obtuvo un 5,42% de reducción de costo en comparación con el AG sin partición. El código de evolución diferencial obtuvo un 7,68% de ahorro al medirlo con el de ED simple.

A continuación se muestra en la gráfica de la Figura 20 se observan el desarrollo de las metodologías evaluadas.

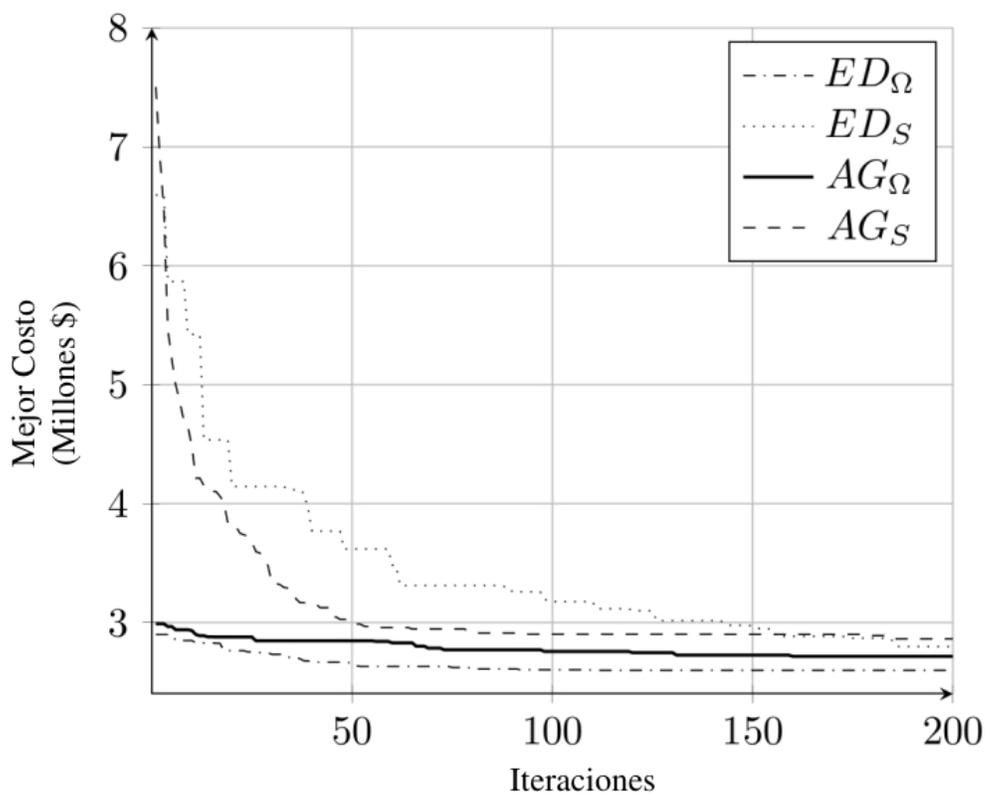


Figura 20 - Costo vs. número de iteraciones, segundo caso.

Elaboración propia.

Tabla 19 - Diámetros y presiones de la red optimizada, segundo caso.

Elaboración propia.

Tuberías	Diámetro Óptimo Tesis (mm)	Nodos	Presión obtenida Tesis (m)	Presión mínima requerida (m)
1	900	1	100	100
2	900	2	95	95
3	350	3	98,60	85
4	250	4	96,00	85
5	150	5	89,60	85
6	200	6	85,42	85
7	800	7	84,35	82
8	150	8	89,80	82
9	400	9	92,85	85
10	500	10	88,27	85
11	800	11	86,77	85
12	600	12	85,05	85
13	500	13	82,25	82
14	450	14	93,79	82
15	150	15	87,99	85
16	450	16	82,15	82
17	400	17	91,06	82
18	350	18	86,44	85
19	150	19	85,84	82
20	150	20	82,26	82
21	700	21	85,51	82
22	150	22	82,51	80
23	450	23	82,03	82
24	350	24	80,18	80
25	600	25	80,75	80
26	250	26	80,28	80
27	200			
28	300			
29	200			
30	250			
31	250			
32	150			
33	150			
34	150			

4.3 Tercer caso de estudio

La mejor configuración para la red se obtuvo con la configuración 5 de la Tabla 20.

Tabla 20 - Mejor ajuste con ED, tercer caso.

Elaboración propia.

Nro. De combinación	F	Cr	Costo Optimo (€)	Eficiencia %	Tiempo (s)
1	0.2	0.3	2151611,30	63,45	2918,08
2		0.4	2309068,24	55,70	3646,46
3		0.5	2220792,31	55,79	1846,41
4		0.6	2436192,72	61,07	1610,54
5	0.3	0.3	1884523,01	47,69	4939,60
6		0.4	1915417,32	41,56	5030,87
7		0.5	2025050,86	45,75	2829,39
8		0.6	1980652,17	41,90	3163,74
9	0.5	0.3	1950454,91	23,83	5724,35
10		0.4	1913553,63	27,90	5988,47
11		0.5	1960843,77	19,57	5692,13
12		0.6	1986476,59	13,60	5330,47
13	0.8	0.3	2296840,21	14,21	6539,75
14		0.4	2309421,97	12,66	7041,84
15		0.5	2293561,73	8,16	6709,56
16		0.6	2349495,54	4,30	6475,80

El comportamiento del algoritmo evolutivo diferencial estudiado es similar al del algoritmo genético simple visto en las tablas 13 y 18. A medida que aumenta la aleatoriedad del método, el tiempo de cálculo aumenta debido a los cambios excesivos en las soluciones propuestas. Estos cambios pueden causar desviaciones que alejan la solución del óptimo. Esto explica el aumento considerable del tiempo de cálculo a medida que

aumentan los factores de variación y combinación, así como los resultados altamente costosos en comparación con el ajuste elegido, debido a la excesiva variación de las soluciones generadas.

Debido a que el algoritmo genético simple tiene un comportamiento repetitivo, se utilizó la misma configuración que los dos casos anteriores para este tercer caso. Los mejores ajustes obtenidos que se muestran en la tabla 21 tabla. El subíndice Ω indica que el algoritmo emplea la estrategia de subdivisión y base de datos, el subíndice S se refiere a que el algoritmo empleado no emplea la partición de redes y la base de datos.

Tabla 21- Tiempo y costo de optimización según la técnica utilizada, tercer caso.

Elaboración propia.

Técnica Utilizada	Costo (€)	Tiempo Total (seg)
ED_{Ω}	1875622,87	5096,20
ED_S	2189598,27	3413,48
AG_{Ω}	2213996,63	5514,54
AG_S	2418264,76	3667,41

El código de partición de redes mediante ED logró una reducción del costo del 16,74% después de 50 iteraciones en comparación con su homólogo simple. Por otro lado, la partición de redes en el AG logró una reducción del costo del 9,23% al compararlo con el AG sin partición.

Por otra parte, el mejor costo obtenido para la red de Pescara resultó es 3,52% mayor que el reportado por H. M. Lee y otros (2016), el cual fue de 1809241 €. En la figura 21 se muestra el costo por iteración de cada algoritmo.

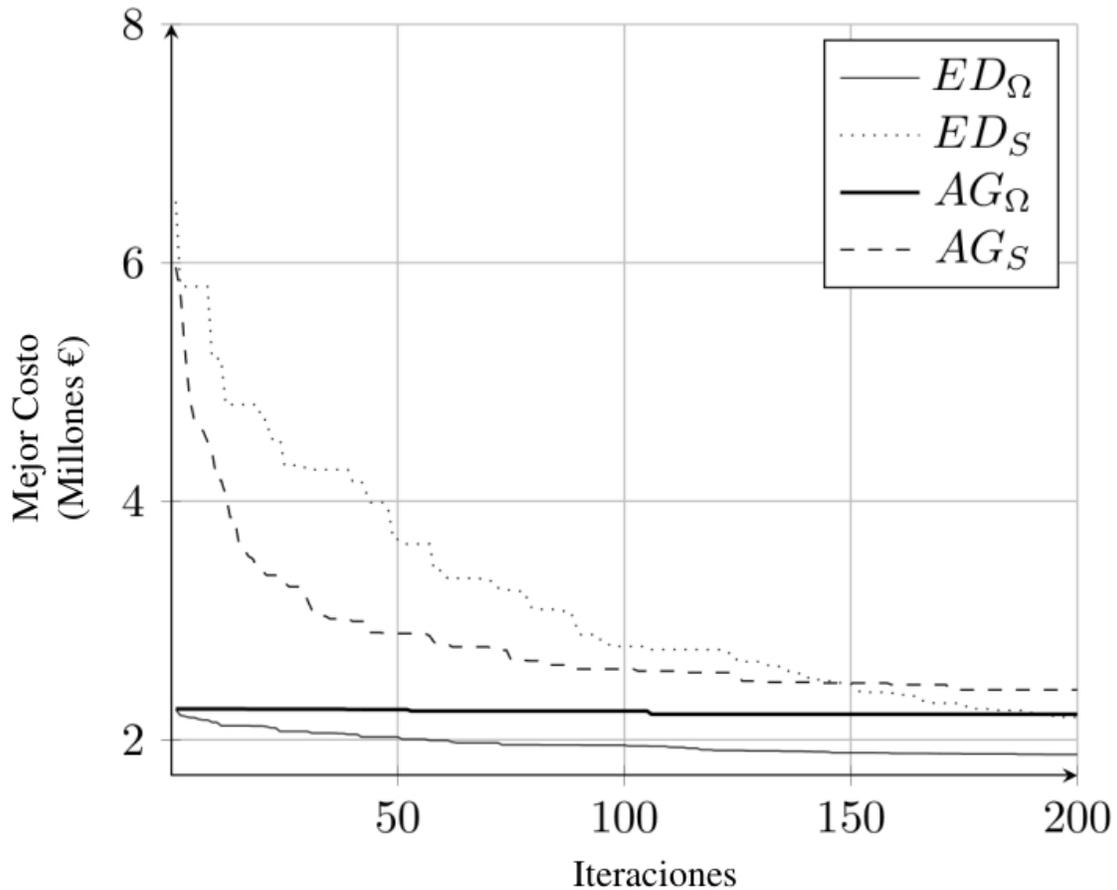


Figura 21. Costo vs. número de iteraciones, tercer caso.

Elaboración propia.

El comportamiento del algoritmo evolutivo diferencial estudiado es similar al del algoritmo genético simple. A medida que aumenta la aleatoriedad del método, el tiempo de cálculo aumenta debido a los cambios excesivos en las soluciones propuestas. Estos cambios pueden causar desviaciones que alejan la solución del óptimo. Esto explica el aumento considerable en el tiempo de cálculo a medida que aumentan los factores de variación y combinación, así como los resultados altamente costosos en comparación con el ajuste elegido.

Debido a que el algoritmo genético simple tiene un comportamiento repetitivo, se utilizó la misma configuración que en los dos casos anteriores para el tercer caso. Teniendo

en cuenta esto, así como el mejor ajuste obtenido, se obtuvo finalmente el siguiente cuadro, que muestra los resultados de cada uno de los métodos con la red de Pescara.

La complejidad de una red de distribución de agua afecta a los resultados obtenidos. El tercer caso de estudio es el más complejo, ya que tiene un gran tamaño y un alto número de mínimos locales. Esto dificulta la búsqueda de la solución óptima, incluso con los mejores parámetros. El primer caso de estudio es el más simple, ya que tiene un número menor de tuberías y un diseño más sencillo. Esto permite que la solución converja más fácilmente.

A partir de los óptimos obtenidos entre el costo, eficiencia y tiempo, se puede ver que a medida que disminuye el valor de eficiencia, el tiempo aumenta. Al mismo tiempo, aquellas evaluaciones donde el algoritmo de evolución diferencial consigue una mejor eficiencia, el costo de la RDA aumenta y el tiempo disminuye. Este comportamiento se observa claramente en el segundo y tercer caso de estudio. La eficiencia disminuye porque ahora el algoritmo tiene una mayor variación en los vectores del conjunto de soluciones, aumentando el espacio de búsqueda y generando un gran número de propuestas con mayor aleatoriedad, por lo que la cantidad de vectores factibles es aun menor en comparación al número de vectores generados.

El algoritmo ED obtuvo una reducción del 20,22% en relación al algoritmo genético.

4.4 Cuarto caso de estudio

Durante la aplicación del método de partición de redes, se encontró un fallo en el método de subdivisión propuesto. El fallo se produjo cuando un nodo (276) quedó aislado de la red. Lo que trae como consecuencia la imposibilidad de realizar el análisis hidráulico de la red.

Tabla 22. Pendiente hidráulica en nodos 275 y 276. Elaboración propia.

Nodos	Ruta óptima	Distancia	Diferencia de carga (m) $H_k - H_i^{\min}$	Pendiente hidráulica (adimensional)
275	Desde el Reservorio 38	2995	117-86	0,010350584
	Desde el Reservorio 44	3498	122-86	0,010291595
276	Desde el Reservorio 38	3132	117-91	0,008301405
	Desde el Reservorio 44	3635	122-91	0,008528198

El cuadro muestra las pendientes hidráulicas de dos nodos pertenecientes a la red de Balerna, se observa que solo uno de los tanques puede surtir de la presión requerida al nodo (276) aunque exista un reservorio más cercano, este conflicto aísla el nodo de la red ya que el algoritmo detecta que no pertenecen a una misma subdivisión.

Por lo tanto se sugiere una metodología que evite la posibilidad de aislar nodos debido a la remoción de tuberías, como la empleada por H. M. Lee y otros (2016).

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Como consecuencia de lo expuesto, se puede concluir que todos los métodos de búsqueda heurística estudiados son muy buenos para el análisis de redes de acueductos, destacando el Algoritmo Evolutivo Diferencial por los bajos costos obtenidos en cada uno de los casos de estudio, dicho algoritmo fue desarrollado con éxito en lenguaje FORTRAN e implementado en el código del tutor sin utilizar ningún tipo de paquetes comerciales.

De los resultados obtenidos se desprende que la complejidad de una red de acueductos afecta en gran medida los tiempos de ejecución. El tercer caso de estudio es el más complejo de todos, ya que posee un alto número de tuberías y mínimos locales. Esto dificulta la búsqueda de la solución óptima, incluso con los mejores parámetros, sin embargo a pesar de la no linealidad del problema los algoritmos desarrollados fueron capaces de encontrar soluciones en tiempos aceptables. El primer caso de estudio es el más simple, lo que permite que la solución converja con mayor facilidad.

A partir de los óptimos obtenidos en el costo, eficiencia y tiempo, se observa que a medida que disminuye el valor de eficiencia, se incrementa el tiempo. En cambio, arroja altos costos de la red de distribución de agua en donde las eficiencias son mayores, y en tiempos menores. Este comportamiento se observa en el segundo y tercer caso de estudio. La eficiencia disminuye debido al aumento en la variación de los vectores, provocando un incremento del espacio de búsqueda generando propuestas a solución con una mayor dispersión, por lo que la cantidad de vectores factibles es mucho menor en contraste con el número de vectores generados.

5.2 Recomendaciones

Desarrollar una rutina que identifique el mejor ajuste mediante un módulo de autoadaptación. Esto con el fin de seleccionar los operadores de variación y combinación que mejor se adaptan a un problema concreto en una menor cantidad de iteraciones. La rutina de autoadaptación también puede adaptarse a los cambios y seleccionar operadores que sean más robustos ante cambios en los parámetros del problema. Esto puede ayudar al algoritmo ED a encontrar soluciones que sean válidas para un rango más amplio de condiciones.

El método propuesto de descomposición y optimización de múltiples etapas es una herramienta prometedora para optimizar redes de distribución de agua. Sin embargo, el método actual solo ha sido verificado mediante RDA con redes de tuberías simples. Para abordar la optimización de redes más complejas con bombas, tanques y/o válvulas, será necesario modificar el método.

Se toma como recomendación mejorar o variar el tipo de subdivisión de redes para evitar el aislamiento de nodos, de esta manera se garantiza un análisis completo del sistema de red de acueductos.

Durante el análisis mediante evolución diferencial se encontraron diferencias en el costo al cambiar la estrategia de variación, la ecuación 2.26 “azar a mejor” fue la que obtuvo mejores resultados. Se sugiere llevar a cabo un estudio para seleccionar la estrategia que mejor se adapte para optimizar redes de acueductos

REFERENCIAS BIBLIOGRÁFICAS

- Amat, J. (2019) Optimización con algoritmo genético en Python. Obtenido de https://www.cienciadedatos.net/documentos/py01_optimizacion_ga
- Angira, R. y Babu, B. (2005) Non-dominated Sorting Differential Evolution (NSDE): An Extension of Differential Evolution for Multiobjective Optimization. Obtenida el 17 de abril de 2022, de <https://www.researchgate.net/publication/220888373>
- Bilal Pant, M. (2022). Differential Evolution for Water Management Problems. Obtenido de: Kumar, B.V., Oliva, D., Suganthan, P.N. Differential Evolution: From Theory to Practice. Studies in Computational Intelligence, Vol. 1009 (1): 197-214. Springer, Singapore.
- Chakraborty, U. (2008) Advances in Differential Evolution. Springer, Verlag Berlin. 340p.
- Corvera A., Medellín P., Capetillo C. y Badillo H. (2021) Herramienta informática para diseño de redes hidráulicas presurizadas. Obtenido de <https://www.scielo.org.mx/pdf/au/v31/2007-9621-au-31-e3093.pdf>
- Črepinšek, M., Liu, SH., Mernik, M., Ravber, M. (2022). The Trap of Sisyphean Work in Differential Evolution and How to Avoid It. Obtenido de: Kumar, B.V., Oliva, D., Suganthan, P.N. Differential Evolution: From Theory to Practice. Studies in Computational Intelligence, Vol. 1009 (1): 137-174. Springer, Singapore.
- Deo, N. (2017) Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, Englewood Cliffs, N. J. 598p.
- Kadu, M., Gupta, R., Bhave, P. (2008) Optimal Design of Water Networks Using a Modified Genetic Algorithm with Reduction in Search Space. Journal of Water Resources Planning and Management, Vol. 134 (2): 147–160.

Kumar, D. y Reddy, M. (2007) Multipurpose Reservoir Operation Using Particle Swarm Optimization. *Journal of Water Resources Planning and Management*, Vol. 133 (3): 192–201.

Kumar, V., Oliva, D., Suganthan, P. (2022) *Differential Evolution From Theory to Practice*. Springer, Nature Singapore. Vol. 1009 (1) 389p.

Feoktistov, V. (2006) *Differential Evolution, In Search of Solutions*. Springer, Estados Unidos de America. 200p.

Fister, I., Fister, D., Fister, I. (2022) Reinforcement Learning-Based Differential Evolution for Global Optimization. Obtenido de: Kumar, B.V., Oliva, D., Suganthan, P.N. *Differential Evolution: From Theory to Practice*. *Studies in Computational Intelligence*, Vol. 1009 (1): 43-75. Springer, Singapore.

Gaceta Oficial de la República de Venezuela, 4044 (Extraordinario), septiembre 8, 1988.

García, M. (Año Desconocido) Ubicando fuentes de agua. *Revista El Maquinista*, vol. 3 Obtenido de <https://www.yumpu.com/es/document/read/12721320/el-maquinista-3pdf-bomberos-del-estado-miranda/2>

García, R. (1993) Dimensionamiento óptimo de redes de distribución de agua ramificadas considerando los elementos de regulación. Tesis doctoral. Universidad Politecnica de Valencia, Valencia.

Gomez, M. (2013) Diseño de la red de suministro de agua a una población de 20000 habitantes y alta variación estacional. Trabajo Especial de Grado. Universidad Politécnica de Valencia, Valencia.

González, M. (2013) Optimización de Redes Hidráulicas Mediante Aplicación de

Algoritmos Genéticos y de Recocido Simulado Utilizados en Programas Comerciales. Trabajo Especial de Grado. Escuela Colombiana de Ingeniería Julio Garavito, Bogotá.

Ho Min Lee, Do Yoo, Sadollah A. y Joong Kim (2016) Optimal cost design of water distribution networks using a decomposition approach. *Engineering Optimization*. Vol. 48. 2141-2156.

Mansouri, R., Torabi, H., Hoseini, M., Morshedzadeh, H. (2015) Optimization of the Water Distribution Networks with Differential Evolution (DE) and Mixed Integer Linear Programming (MILP). *Journal of Water Resource and Protection*, 7, 715-729.

Mansouri, R. y Mohamadizadeh, M. (2017) Optimal Design of Water Distribution System Using Central Force Optimization and Differential Evolution. *International Journal of Optimization in Civil Engineering*, Vol. 7 (3): 469-491

Montesinos, M., Guzmán, A., Ayuso, J. (1996) Optimización de Redes de Distribución de Agua Utilizando un Algoritmo Genético. *Ingeniería del Agua*, Vol. 4 (1): 71-77

Mora, D. (2012) Diseño de redes de distribución de agua mediante algoritmos evolutivos y análisis de eficiencia. Tesis doctoral. Universidad Politécnica de Valencia, Valencia.

Pino, E., Valle, A., Condori, F., Mejia, J., Chavarri, E., Alfaro, L. (2017) Diseño Óptimo de Redes de Distribución de Agua Usando Un Software Basado En Microalgoritmos Genéticos Multiobjetivos. *Ribagua*, Vol. 4 (1): 6-23

Plagianakos, V., Tasoulis D., Vrahatis, M. (2008) A Review of Major Application Areas of Differential Evolution

Potter, M. y Wiggert, D. (2012) *Mechanics of Fluids*, 4ta ed. Cengage Learning 818p.

Price, K., Storn, R., Lampinen, J. (2005) Differential Evolution A Practical Approach to Global Optimization. Springer, Verlag Berlin. 543p.

Onwubolu, G. y Davendra, D. (2009) Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization. Springer, Verlag Berlin. 226p.

Saldarriaga J., Páez D., Salcedo C., Cuero P., López L., León N., Celeita D. A (2020) Direct Approach for the Near-Optimal Design of Water Distribution Networks Based on Power Use. Journal of Water 2020, Vol. 12 (4). 1037.

Sirsant, S., Janga Reddy, M. (2022). Sobol Sequence-based MOSaDE Algorithm for Multi-objective Design of Water Distribution Networks. Obtenido de: Kumar, B.V., Oliva, D., Suganthan, P.N. (eds) Differential Evolution: From Theory to Practice. Studies in Computational Intelligence, Vol. 1009 (1): 215-240. Springer, Singapore.

Storn, R. y Price K. (1995) Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI March

Suribabu, C. (2010) Differential evolution algorithm for optimal design of water distribution networks. Journal of Hydroinformatics, Vol. 12 (1): 66-88.

Vasan, A. y Simonovic, P. (2010) Optimization of Water Distribution Network Design Using Differential Evolution. Journal of Water Resources Planning and Management, Vol. 136 (2): 279–287.

Velásquez, J. (2022) Solución del Flujo Estacionario Incompresible en redes de Acueductos mediante el Método del Elemento Finito. Trabajo Especial de Grado, Universidad Central de Venezuela, Caracas.

Vermetten, D., Van Stein, B., Kononova, A.V., Caraffini, F. (2022). Analysis of Structural Bias in Differential Evolution Configurations. Obtenido de: Kumar, B.V., Oliva, D., Suganthan, P.N. (eds) Differential Evolution: From Theory to Practice. Studies in Computational Intelligence, Vol. 1009 (1): 1-22. Springer, Singapore.

Zecchin, A., Maier, H., Simpson, A., Leonard, M., Nixon, J. (2007) Ant Colony Optimization Applied to Water Distribution System Design: Comparative Study of Five Algorithms. Journal of Water Resources Planning and Management, Vol. 133 (1): 87-92

Zheng, F., Simpson, A., Zecchin A. (2013a) A decomposition and multistage optimization approach applied to the optimization of water distribution system with multiple supply sources. Water Resources Research, Vol. 49: 380-399

Zheng, F., Simpson, A., Zecchin, A. (2013b) Self-Adaptive Differential Evolution Algorithm Applied to Water Distribution System Optimization. Journal of Computing in Civil Engineering, Vol. 27 (2): 148-158

GLOSARIO

AE: Algoritmos evolutivos

Cr: Factor de “cruce” o combinación.

DFC: Dinámica de fluidos computacional.

ED: Evolución Diferencial.

EPANET: Es un software de código abierto desarrollado por la Agencia de Protección Ambiental de Estados Unidos (EPA) o *Environmental Protection Agency Network Evaluation Tool* por sus siglas en inglés.

F: Factor de “mutación” o de variación.

FORTRAN: es un lenguaje de programación de alto nivel, procedimental e imperativo, que está especialmente adaptado al cálculo numérico y a la computación científica. Sus siglas significan vienen del inglés *FORmula TRANslating system*, que en español significa "Sistema de Traducción de Fórmulas".

ISO: Las normas ISO son un conjunto de requisitos técnicos que se aplican a productos, servicios, procesos y sistemas. Las siglas se refieren a la Organización Internacional de Normalización o *International Standart Organization* por su nombre en inglés.

L: Largo de la tubería.

MCA: Un metro de columna de agua (mca) es una unidad de presión que equivale a la presión ejercida por una columna de agua pura de un metro de altura bajo la gravedad terrestre.

NG: Número máximo de iteraciones o “generaciones”.

NP: Número de “Población” o número de vectores solución que se usan en el proceso de optimización.

NT: Número de tuberías.

P: “Población” o número de vectores que se usa en el proceso de optimización.

PL: Programación lineal

PNL: Programación no Lineal

PR: Carga de presión requerida.

RDA: Redes de distribución de agua.

TSP: Problema del Viajante Vendedor o *Travel Salesman Problem* por sus siglas en inglés.

ANEXOS

Anexo 1. Esquema del algoritmo de Dijkstra

Se ejecuta el algoritmo de Dijkstra para un grafo representado mediante una matriz de adyacencia. De esta manera se encuentra la ruta de distancia mas corta para cada nodo a diferentes fuentes de suministro. Siendo:

k : Indica cual nodo actúa como reservorio de agua

N : Numero de nodos de la red

$G_{N,j}$: Grafo o Matriz de adyacencia

Rec : Vector con el camino más corto desde el nodo k hasta el resto de nodos, la distancia del nodo fuente a si mismo siempre es cero.

Prc : Vector que lleva el conteo de nodos procesados

Nmn : Nodo mas cercano al nodo k

Crt : Vector que almacena la ruta mas corta desde el nodo fuente hasta el resto de nodos de la red.

Np : Dimensión del conjunto de soluciones

La secuencia a seguir es la siguiente:

Hacer $Rec_k = 0$

Para $i = 1, \dots, N$

$Nmn = \text{Evaluar}(Rec, N, Prc)$

$Prc(Nmn) = \text{Verdadero}$

Para $j = 1, \dots, N$

Sí ($G_{Nmn,j}$ es distinto de 0) y (Prc_j es Falso) y ($Rec_{Nmn} + G_{Nmn,j}$ es menor que Rec_j) Entonces

$Rec_j = Rec_{Nmn} + G_{Nmn,j}$

$Crt_j = Nmn$

Como se observa, el algoritmo de Dijkstra hace uso de la función Evaluar, la cual también se conoce como algoritmo voraz. Esta se encarga de seleccionar el nodo adyacente más cercano al nodo fuente, con la condición de que el recorrido sea menor

y que no haya sido evaluado anteriormente. Este funciona siguiendo el siguiente proceso:

```

subroutine optith_disjkt
  use global_mod
  use varthe_mod
  use optpar_mod
  implicit none
  integer :: inods, jnodes, itank, nodmn, greedy
  integer, dimension(NPOIN) :: recor, parnt
  logical, dimension(NPOIN) :: procs

  !algoritmo de Disjktra
  allocate(lamda(NPOIN,NTANK))
  call mtxini(lamda,NPOIN,NTANK)
  do itank = 1, NTANK
    parnt(tnkvc(itank)%knode) = 0
    do inods = 1, NPOIN
      recor(inods) = infin
      procs(inods) = .FALSE.
    enddo
    recor(tnkvc(itank)%knode) = 0
    do inods = 1, NPOIN
      nodmn = greedy(recor,procs)
      procs(nodmn) = .TRUE.
      do jnodes = 1, NPOIN
        if ((grafo(nodmn,jnodes) .ne. 0) .and. &
            (.not. procs(jnodes)) .and. (recor(nodmn)+grafo(nodmn,jnodes) &
            .lt. recor(jnodes))) then
          recor(jnodes) = recor(nodmn) + grafo(nodmn,jnodes)
          parnt(jnodes) = nodmn
        endif
      enddo
    enddo
    lamda(:,itank) = recor
  enddo
end subroutine optith_disjkt

```

Finalmente, el algoritmo Djisktra se ejecuta K veces para encontrar el camino más corto desde cada fuente de suministro al resto de los nodos.

Anexo 2. Esquema del programa de partición de RDA

Paso 1:

Para $k=1, \dots, K$

Desarrollar el algoritmo de Dijkstra para la fuente de suministro k para identificar la ruta más corta hasta el resto de nodos de la red.

Computar $P_{fric_{k,N}}$ para cada nodo n usando la ecuación 2.32.

Paso 2:

Para $i=1, \dots, n$

Comparar los valores de $P_{h_{k,N}}$ para cada nodo y el que tenga el mayor valor para la pendiente hidráulica en el nodo i , es asignado a el conjunto G_k .

Paso 3:

Para $i=1, \dots, NT$ analizar los extremos de cada tubería

Si son iguales ambos nodos, se asigna la tubería a dicha subred.

Si ambos nodos pertenecen a distintas subredes, es asignada al conjunto de partición óptimo Ω .

Anexo 3. Estrategia de exploración para encontrar el mejor parámetro

Paso 1: Descomposición de la RDA

Paso 2: Establecer los parámetros de control para la optimización

Para $i = 1, \dots, 16$

$F = DC(i)(1)$

$Cr = DC(i)(2)$

Paso 3: Realizar el algoritmo ED para optimizar la red

Fin de Para

```
subroutine optimi_proced
  call optimi_access
  call optimi_introd
  call optimi_subdvs

  F = (/0.2,0.2,0.2,0.2,0.3,0.3,0.3,0.3,0.5,0.5,0.5,0.5,0.8,0.8,0.8,0.8/)
  C = (/0.3,0.4,0.5,0.6,0.3,0.4,0.5,0.6,0.3,0.4,0.5,0.6,0.3,0.4,0.5,0.6/)

  timpr = 0
  N = 16
  iteri = 50
  band = .TRUE.

  do corcd = 1,N                                !busqueda de la mejor combinacion
    ahorr = 0
    fac_F = F(corcd)
    fac_C = C(corcd)
    call cpu_time(tempo1)
    call optimi_DEproc                            !Algoritmo de Optimizacion
    call cpu_time(tempo2)
    cstof(corcd) = bests(NTANK+1)%price          !se guarda el resultado

    deallocate (bests, effgb, subnt, DEvct%headp, DEvct%piezm, DEvct%presn)
  enddo

  .
  .
  .

  band = .FALSE.
  do corcd = 1, iteri                            !se ejecuta el algoritmo ED 50 veces
    !con el mejor ajuste del ciclo anterior
    call cpu_time(tempo1)
    call optimi_DEproc                            !Algoritmo de Optimizacion
    call cpu_time(tempo2)
    opcio(corcd) = bests(NTANK+1)%price

    deallocate (bests, effgb, subnt, DEvct%headp, DEvct%piezm, DEvct%presn)
  enddo
end subroutine optimi_proced
```

Anexo 4. Base de Datos

La base de datos se genera añadiendo los datos de cada vector evaluado en una variable estructurada que almacena información de los vectores generados durante el proceso de optimización.

Declaración de la base de datos en FORTRAN.

```
module varthe_mod
  use global_mod
  implicit none
  type population
    real(kind=dp), allocatable, dimension(:) :: piped, presn
    real(kind=dp) :: price, pnltly
  end type population

  type(population), allocatable, dimension(:) :: bdata
end module varthe_mod
```

Uso de la base de datos en conjunto con la función objetivo

```
subroutine objtiv_funcn(persn,itank)
  !funcion objetivo
  if (.not. its_bd(persn)) then
    .
    . !se calcula la funcion objetivo empleando el ANÁLISIS HIDRÁULICO
    . !de ELEMENTOS FINITOS mediante el trabajo de Velazques (2022) y la tutoría
    . !del prof. Eugenio Sansone. De no encontrarse el elemento en la base de datos, se anexa.
    .
    dbdat = dbdat + 1
    allocate(bdata(dbdat)%piped(NELEM),bdata(dbdat)%presn(NELEM))
    bdata(dbdat) = persn
  else
    persn = bdata(posiv)
  endif
end subroutine objtiv_funcn
```

Anexo 5. Algoritmo de Evolución Diferencial

- 1) Se emplean distintos operadores de variación para comprender cual se ajusta mejor para problemas de RDA.
- 2) Por cada ciclo en la evaluación de la función objetivo, se ejecuta la penalización por pared de ladrillos para descartar los vectores que incumplen con las presiones requeridas.
- 3) Se implementa una base de datos que guarda la configuración de diámetros de cada vector solución, luego en una nueva operación donde sea requerido efectuar un análisis hidráulico de la RDA, se compara con la base de datos para verificar si ya ha sido evaluado antes dicho vector.

Notación:

NP: cantidad de vectores del conjunto de soluciones.

NT: número de tuberías.

$P_G[X_{i,G}][X_{j,G}^D]$: P_G es la variable que almacena el numero de vectores del conjunto de soluciones, $[X_{i,G}]$ es el subíndice que indica el numero del vector, $[X_{j,G}^D]$ indica el diámetro para cada tubería.

$FO[K][P_G]$: es el comando que ejecuta la función objetivo, para ejecutarse recibe el subíndice $[K]$ que indica el reservorio de agua y $[P_G]$, guardar los diámetros de tuberías de un vector a la vez.

$f(X_{1,G}), \dots, f(X_{NP,G})$: notación que simboliza la aplicación de la función objetivo, se ejecuta una vez por cada vector hasta cubrir toda el conjunto de soluciones

Paso 1: Generar el conjunto de soluciones

Para $i=1, \dots, NP$

Para $j=1, \dots, NT$

$P_G[X_{i,G}][X_{j,G}^D] = \text{Diámetro aleatorio de la lista de diámetros comerciales}$

Paso 2: Computar función objetivo para NP vectores iniciales

$$\{FO[k][P_G] = f(X_{1,G}), \dots, f(X_{NP,G})\}$$

Contador = 1

Repetir hasta. Contador > NG

Paso 3: Realizar la variación ED para generar N soluciones variantes V

Para $i=1, \dots, NP$

$$V_{i,G} = X_{r1} + F(X_{r2} - X_{r3})$$

Paso 4: Realizar el operador de combinación ED para generar soluciones de prueba $U_{i,G}$

Para $i=1, \dots, NP$

Para $j=1, \dots, NT$

Si Aleatorio(0,1) \leq CR,

$$u_{i,G}^j = v_{i,G}^j$$

Si No

$$u_{i,G}^j = x_{i,G}^j$$

Paso 5: Discretizar el diámetro continuo de la tubería a el diámetro más cercano y evaluar función objetivo.

Calcular la función objetivo del vector de prueba $u_{i,G}$ si este cumple con las restricciones.

Paso 6: Tomar para la siguiente iteración el vector objetivo si este tiene un menor costo, en caso contrario se conserva el vector de prueba.

$$X_{i,G+1} = \begin{cases} U_{i,G} & \text{si } f(u_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{de otra manera.} \end{cases}$$

Contador = Contador + 1

Algoritmo de evolución diferencial desarrollado en FORTRAN 95:

```
subroutine thesis_DEalgr(itank)
  use global_mod
  use varthe_mod
  implicit none
  integer                :: itank, ipers, gener, ielem
  integer, dimension (6) :: nrand
  real(kind=dp)         :: randm

  !ALGORITMO GENETICO DIFERENCIAL EVOLUTIVO
  mxgen = 200
  popsz = 10*NCOST
  dbdat = 0
  fac_F = 0.3
  fac_C = 0.3

  allocate(popln(popsz), xpopl(popsz), cross(popsz))

  !PASO 1: creacion de la poblacion inicial
  do ipers = 1, popsz
    allocate(popln(ipers)%piped(NELEM), xpopl(ipers)%piped(NELEM), cross(ipers)%piped(NELEM))
    call create_person(popln(ipers), itank)

    !PASO 2: evaluamos el individuo creado
    call objtiv_functn(popln(ipers), itank)
  enddo

  call bests_person(itank)
```

```

do gener = 1, mxgen

    !PASO 3: mutaciones
    do ipers = 1, popsz
        call random(6,popsz,rand)
        xpopl(ipers)%piped = popln(nrand(1))%piped+ fac_F*(bests(itank)%piped - popln(nrand(2))%piped)
        xpopl(ipers)%piped = xpopl(ipers)%piped+ fac_F*(popln(nrand(3))%piped - popln(nrand(4))%piped)
        xpopl(ipers)%piped = xpopl(ipers)%piped+ fac_F*(popln(nrand(5))%piped - popln(nrand(6))%piped)
    enddo

    !PASO 4: cruces
    do ipers = 1, popsz
        do ielem = 1, NELEM
            call random_number(randm)
            if ((randm .lt. fac_C) .and. (popln(ipers)%piped(ielem) .ne. 0)) then
                cross(ipers)%piped(ielem) = xpopl(ipers)%piped(ielem)
            else
                cross(ipers)%piped(ielem) = popln(ipers)%piped(ielem)
            endif
        enddo
    enddo

    !PASO 5: Discretizacion de los resultados
    do ipers = 1, popsz
        call dsqrtz_result(cross(ipers),itank)
        call objtiv_functn(cross(ipers),itank)
    enddo

    !PASO 6: Seleccion
    do ipers =1, popsz
        if (cross(ipers)%price*cross(ipers)%pnlty .le. popln(ipers)%price*popln(ipers)%pnlty) then
            popln(ipers) = cross(ipers)
        endif
    enddo

    enddo
    deallocate(popln,xpopl,cross,bdata)
end subroutine thesis_DEalgr

```

Anexo 6. Procedimiento para optimizar RDA con Algoritmo Genético

Notación:

S : Vector seleccionado por torneo.

C : Vector combinación.

NT : Número de tuberías

NP: Número del conjunto de soluciones

La metodología de optimización empleando Algoritmo Genético sugerida consiste de los siguientes pasos:

Paso 1: Se codifica un conjunto de soluciones inicial de P vectores elegidos aleatoriamente de la lista de diámetros comerciales.

Para $i=1, \dots, NP$

Para $j=1, \dots, NT$

$P_G[X_{i,G}][X_{j,G}^D] =$ Diámetro aleatorio de la lista de diámetros comerciales

Paso 2: Calcular la aptitud de cada vector del conjunto de soluciones. Se calcula el valor de la función objetivo para cada vector.

$$\{FO[P_G] = f(X_{1,G}), \dots, f(X_{NP,G})\}$$

Reproducir un nuevo conjunto de soluciones manipulando los vectores del conjunto de soluciones, hasta generar una descendencia nueva de P_G vectores de acuerdo a los siguientes pasos:

Para $i=1, \dots, P$

Paso 3: Selección ponderada: se eligen aleatoriamente dos parejas de vectores, son comparadas y se selecciona la que tenga un menor costo.

$$S_{j,G}^D = X_{r1,G} \text{ o } X_{r2,G} \text{ (El que posea menor costo, con } X_{r1,G} \text{ distinto a } X_{r2,G}\text{)}$$

Paso 4: Combinación de un solo punto: se selecciona aleatoriamente una posición que actúa como punto de corte entre el vector objetivo y el vector seleccionado por

torneo, cada vector se divide en dos partes y se intercambian las mitades para generar dos vectores de los cuales se toma cualquiera de ellos para generar un nuevo descendiente C.

$$C_{j,G}^D = \%A \text{ aleatorio de } P_G(X_{i,G}) + \%B \text{ aleatorio de } S_{j,G}^D$$

Paso 5: Aplicar un proceso de variación aleatorio sobre el nuevo vector.

Para $j=1, \dots, NT$

R = Número aleatorio entre 0 y 1

Sí $R < \text{Factor de variación}$

$$C_{j,G}^D = \text{Diámetro aleatorio de la lista de diámetros comerciales}$$

Paso 6: Añadir el nuevo vector al conjunto de soluciones si este tiene un menor costo, en caso contrario se conserva el vector anterior.

$$X_{i,G+1} = \begin{cases} C_{j,G}^D & \text{sí } f(X_{i,G}) \leq f(C_{j,G}^D) \\ X_{i,G} & \text{de otra manera.} \end{cases}$$

Si no se cumple el criterio de parada, volver al paso 2 hasta completar el número determinado de iteraciones.

```

subroutine thesis_AGalgr(itank)
  use global_mod
  use varthe_mod
  implicit none
  integer                :: itank, ipers, jpers, kpers, ichld, gener, ielem
  integer, dimension (4) :: nrand
  real(kind=dp)         :: randm, diamtr_random
  type(population)     :: indivA, indivB, winer, child_1, child_2, AUX

  !ALGORITMO GENETICO
  mxgen = 200
  popsz = 10*NCOST*2
  fac_F = 0.1

  allocate(popln(popsz), xpopl(popsz), cross(popsz))

  !PASO 1: creacion de la poblacion inicial
  do ipers = 1, popsz/2
    allocate(popln(ipers)%piped(NELEM), xpopl(ipers)%piped(NELEM), cross(ipers)%piped(NELEM))
    call create_person(popln(ipers), itank)

    !PASO 2: evaluamos el individuo creado
    call objtiv_funcfn(popln(ipers), itank)
  enddo

  do gener = 1, mxgen
    do ichld = 1, popsz/2

      !PASO 3: Seleccion por Torneo
      call random(4, popsz, nrand)
      if (popln(nrand(1))%price*popln(nrand(1))%pnlty .lt. &
         popln(nrand(2))%price*popln(nrand(2))%pnlty) then
        indivA = popln(nrand(1))
      else
        indivA = popln(nrand(2))
      endif
      if (popln(nrand(3))%price*popln(nrand(3))%pnlty .lt. &
         popln(nrand(4))%price*popln(nrand(4))%pnlty) then
        indivB = popln(nrand(3))
      else
        indivB = popln(nrand(4))
      endif
      if (indivA%price*indivA%pnlty .lt. indivB%price*indivB%pnlty) then
        winer = indivA
      else
        winer = indivB
      endif
    enddo
  enddo

```

```

!PASO 4:Cruce de un solo punto
call random(1,NELEM,rand)
child_1 = popln(ichld)
child_2 = popln(ichld)
child_1%piped(1:rand(1)) = winer%piped(1:rand(1))
child_2%piped(rand(1):NELEM) = winer%piped(NRand(1):NELEM)
call random_number(randm)
if (randm .lt. 0.5) then
    cross(ichld) = child_1
else
    cross(ichld) = child_2
endif

!PASO 5: mutacion aleatoria
if (itank .le. NTANK) then
    !subredes
    do ielem = 1, sbntw(itank)%npipe
        call random_number(randm)
        if (randm .lt. fac_F)then
            cross(ichld)%piped(sbntw(itank)%pipes(ielelem)) = diamtr_random(1,NCOST)
        endif
    enddo
else
    !red completa
    do ielem = 1, NELEM
        call random_number(randm)
        if (randm .lt. fac_F)then
            cross(ichld)%piped(ielelem) = diamtr_random(1,NCOST)
        endif
    enddo
endif
call objtiv_funcfn(cross(ichld),itank)
popln(popsz/2+ichld) = cross(ichld)
enddo

!PASO 6: Seleccion
do jpers = 1, popsz-1
    do kpers = jpers + 1,popsz
        if (popln(jpers)%price*popln(jpers)%pnlty .gt. popln(kpers)%price*popln(kpers)%pnlty) then
            AUX = popln(jpers)
            popln(jpers) = popln(kpers)
            popln(kpers) = AUX
        endif
    enddo
enddo
bests(itank) = popln(1)
enddo
deallocate(popln,xpopl,cross)
end subroutine thesis_AGalgr

```

Anexo 7. Rutina de conexión con código de análisis DFC

A continuación se presenta la conexión de las variables empleadas por el código de análisis mediante elemento finitos por Velásquez (2022) con las variables empleadas en el algoritmo de optimización por Guzmán y Ramírez (2023).

```
subroutine subnet_opfill
use DEgpar_mod
use varthe_mod
use iterat_mod
use lstapr_mod
!en este procedimiento se añade la cantidad de tuberías y las tuberías por cada subred en la variable subnt
implicit none
integer      :: ipipe, itank
real(kind=dp) :: flow0
allocate(subnt(NTANK))

flow0 = 0.
do ipipe=1,NELEM
write(ITERA_FILE,rec=ipipe) flow0
enddo

DEFLG = 0
call hntwrk_setntw(0)
DEFLG = 1

do itank = 1, NTANK
subnt(itank)%npipe = sbntw(itank)%npipe
subnt(itank)%nnods = sbntw(itank)%nnods

allocate(subnt(itank)%pipes((subnt(itank)%npipe)), subnt(itank)%nodes(subnt(itank)%nnods))

subnt(itank)%pipes = sbntw(itank)%pipes(1:sbntw(itank)%npipe)
subnt(itank)%nodes = sbntw(itank)%nodes(1:sbntw(itank)%nnods)
```

```

subnt(itank)%pipes = sbntw(itank)%pipes(1:sbntw(itank)%npipe)
subnt(itank)%nodes = sbntw(itank)%nodes(1:sbntw(itank)%nnods)

subnt(itank)%niter = 50 !total iterations by subnetwork that are applied in iterat procedure
subnt(itank)%toler = 1e-5 !minimal error tolerance for convergence of flow variables (nodal head, pipe's flow) used in iterat procedure

allocate(subnt(itank)%flowr(NELEM))
allocate(subnt(itank)%headp(NPOIN))
allocate(subnt(itank)%piezm(NPOIN))
allocate(subnt(itank)%presn(NPOIN))

subnt(itank)%nbloc = 1

allocate(subnt(itank)%block(subnt(itank)%nbloc))
subnt(itank)%block(1)%nelmd(1) = subnt(itank)%npipe
subnt(itank)%block(1)%nelmd(2) = 1
subnt(itank)%block(1)%nelmd(3) = 1
subnt(itank)%block(1)%nsbel = 1
subnt(itank)%block(1)%itype = 2
subnt(itank)%block(1)%nelem = subnt(itank)%npipe
subnt(itank)%block(1)%mxmfr = (subnt(itank)%npipe+1)*NDGRE
subnt(itank)%block(1)%npoin = subnt(itank)%npipe+1

call hntwrk_setntw(itank) !set initial subnetwork's flow variables
call lstapr(itank) !set initial finite element matrix parameters of each subnetwork
!
! enddo
end subroutine

```