



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Investigación en Ingeniería
De Procesos y Sistemas Industriales

MÁSTER EN INVESTIGACIÓN EN INGENIERÍA DE PROCESOS Y SISTEMAS INDUSTRIALES

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

Localización Óptima de Sensores para Detectar Fugas en Sistemas de Distribución de Agua Potable

Autor: D. Ernest Gabriel Romero Tapia
Tutor: Dña. María Jesús de la Fuente Aparicio

Valladolid, julio, 2017



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Investigación en Ingeniería
De Procesos y Sistemas Industriales

MÁSTER EN INVESTIGACIÓN EN INGENIERÍA DE PROCESOS Y SISTEMAS INDUSTRIALES

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

Localización Óptima de Sensores para Detectar Fugas en Sistemas de Distribución de Agua Potable

Autor: D. Ernest Gabriel Romero Tapia
Tutor: Dña. María Jesús de la Fuente Aparicio

Valladolid, julio, 2017

RESUMEN

En este trabajo se desarrolla un método de Detección y Diagnóstico de Fallos basado en el Análisis del Discriminante de Fisher para la detección y localización de fugas en redes de distribución de agua potable, a partir de valores de presiones obtenidos de una simulación hidráulica; para el cálculo de sensibilidades. De la misma manera se usan algoritmos genéticos y un algoritmo evolutivo para realizar una optimización donde se busca determinar el número mínimo de sensores de presión que se deben instalar en la red y donde deben estar ubicados para conseguir la mayor cantidad de fugas localizables. Estos métodos son aplicados en dos redes de agua potable diferentes, realizándose varias pruebas modificando el algoritmo de localización de fugas para hacerlo más robusto. Finalmente se hace una comparación de los resultados obtenidos en los experimentos realizados en ambas redes, con bibliografía existente en el área de este trabajo.

Palabras clave: Detección y Diagnóstico de Fallos, Análisis del Discriminante de Fisher, Redes de agua potable, localización de fugas, algoritmos genéticos, algoritmos evolutivos, ubicación de sensores.

ABSTRACT

This work develops a Fault Detection and Isolation method based on Fisher Discriminant Analysis for leak detection and localization in drinking water distribution networks, using pressure values obtained from hydraulic simulation to compute sensitivities values. Furthermore genetics and evolutive algorithms are used to approach an optimization problem to minimize the number of pressure sensors and reach the optimal placement to obtain the maximum leak isolability. This methods are applied in two different drinking water networks, trying several experiments modifying the leak localization algorithm to obtain a more robust method. Finally a comparison of the results is made with the results obtained in a related paper.

Keywords: Fault Detection and Isolation (FDI), Fisher Discriminant Analysis (FDA), water distribution networks, leak isolation, genetic algorithms, evolutive algorithms, sensor placement.

AGRADECIMIENTOS

A la Ph.D María Jesús de la Fuente Aparicio, por la guía y el tiempo brindado para la elaboración de este trabajo, además de todo el apoyo que recibí durante mi estancia en Valladolid.

Al Ph.D Vicenç Puig Cayuela de la Universidad Politécnica de Cataluña, por la colaboración prestada, a nivel teórico y de software y cuya conferencia sirvió de inspiración para la realización de este trabajo.

A mi familia y amigos en Bolivia, que a pesar de la distancia, me hacían sentir todo su apoyo y cariño.

A todos los amigos que conocí en Valladolid, hicieron que esta estancia sea maravillosa.

ÍNDICE

1. INTRODUCCIÓN Y OBJETIVO	1
1.1 Objetivos	2
1.2 Organización del trabajo	2
2. MARCO TEÓRICO	4
2.1 Detección y Diagnóstico de Fallos	4
2.1.1 Métodos generales de detección y diagnóstico de fallos	6
2.1.1.1 Métodos basados en señal	6
2.1.1.2 Métodos analíticos	7
2.1.1.3 Métodos basados en conocimiento.....	8
2.2 Análisis del Discriminante de Fisher.....	10
2.2.1 Detección y diagnosis de fallos.....	13
2.3 Optimización	15
2.4 Algoritmos Genéticos	16
2.4.1 Bases biológicas.....	17
2.4.2 Codificación de problemas.....	17
2.4.3 Operadores genéticos.....	19
2.4.3.1 Selección	19
2.4.3.2 Cruce	22
2.4.3.3 Algoritmos de Reemplazo	26
2.4.3.4 Copia	27
2.4.3.5 Elitismo	27
2.4.3.6 Mutación	27
2.5 Estrategias Evolutivas	28
2.5.1 Algoritmo CMA-ES.....	30

ÍNDICE

3. METODOLOGÍA.....	31
3.1 Metodología de localización de fugas	31
3.2 Metodología para la optimización del uso de sensores.....	34
3.3 Medidas de robustez.....	37
3.3.1 Distancia.....	37
3.3.2 Magnitudes de fuga	39
3.3.3 Ruido	39
3.3.4 Modificación del enfoque de optimización	40
4. APLICACIÓN Y RESULTADOS	41
4.1 Aplicación a la red de Hanói	41
4.1.1 Selección de datos	41
4.1.1.1 Resultados.....	42
4.1.2 Optimización con GA binario	47
4.1.2.1 Resultados sin robustez	48
4.1.2.2 Resultados con robustez	51
4.1.3 Optimización con GA entero.....	53
4.1.3.1 Resultados sin robustez	53
4.1.3.2 Resultados con robustez	54
4.1.4 Optimización con CMA-ES	55
4.1.4.1 Resultados sin robustez	55
4.1.4.2 Resultados con robustez	57
4.1.5 Comparación de los métodos	58
4.2 Aplicación a la red de Limassol.....	59
4.2.1 Prueba con diferentes magnitudes	60
4.2.2 Optimización con GA binario	62
4.2.3 Optimización con GA entero.....	62

ÍNDICE

4.2.4 Optimización con CMA-ES	63
4.3 Comparación con bibliografía existente	64
4.3.1 Hanói	64
4.3.2 Limassol.....	65
5. CONCLUSIONES	68
5.1 Trabajo futuro.....	69
6. BIBLIOGRAFÍA.....	71

ÍNDICE DE FIGURAS

<i>Figura 2.1 Individuo Genético Binario</i>	18
<i>Figura 2.2 Codificación de una red de neuronas artificiales</i>	19
<i>Figura 2.3 Cruce de un punto</i>	23
<i>Figura 2.4 Cruce de dos puntos</i>	24
<i>Figura 2.5 Cruce uniforme</i>	25
Figura 3.1 Representación de la distancia límite.....	38
Figura 4.1 Errores de localización usando presiones.....	43
Figura 4.2 Errores de localización usando residuos.....	43
Figura 4.3 Errores de localización usando sensibilidades.....	44
Figura 4.4 Errores de localización con Sensibilidad de 50 Ips	45
Figura 4.5 Caracterización de diferentes tipos de fugas	46
Figura 4.6 Valores del Discriminante de Fisher para fuga en el nodo 15.....	47
Figura 4.7 Localización del sensor en el nodo 12	48
Figura 4.8 Localización del sensor en el nodo 25	48
Figura 4.9 Localización de los sensores en los nodos 3 y 11	49
Figura 4.10 Localización de los sensores en los nodos 1 y 6	49
Figura 4.11 Localización de los sensores en los nodos 11 y 18	50
Figura 4.12 Localización de los sensores en los nodos 12 y 18	50
Figura 4.13 Configuración obtenida con 7 sensores	51
Figura 4.14 Localización de los sensores en los nodos 12 y 29	52
Figura 4.15 Configuración obtenida con 11 sensores	52
Figura 4.16 Localización de los sensores en los nodos 4 y 30	53
Figura 4.17 Localización de los sensores en los nodos 3, 12 y 29	54
Figura 4.18 Localización del sensor en el nodo 15	54
Figura 4.19 Localización de los sensores en los nodos 11, 15 y 28	55
Figura 4.20 Localización del sensor en el nodo 22	56
Figura 4.21 Localización de los sensores para los nodos 18 y 31	56
Figura 4.22 Localización de los sensores en los nodos 5, 11 y 12	57
Figura 4.23 Localización del sensor en el nodo 29	57

ÍNDICE

Figura 4.24 Localización de sensores en los nodos 3, 76, 87, 108, 163 y 190	62
Figura 4.25 Localización de sensores en los nodos 9, 158 y 184	63
Figura 4.26 Localización de sensores para los nodos 42, 50 y 197	64

ÍNDICE DE TABLAS

Tabla 4.1 Errores de localización para distintos tipos de datos.....	42
Tabla 4.2 Configuraciones obtenidas sin aplicar robustez	58
Tabla 4.3 Configuraciones obtenidas aplicando robustez	59
Tabla 4.4 Configuraciones para diferentes combinaciones de magnitudes .	60
Tabla 4.5 Índice de error para las configuraciones de la tabla 4.4	60
Tabla 4.6 Índices de error al agregar robustez y ruido	61
Tabla 4.7 Comparación de resultados	65
Tabla 4.8 Configuraciones e índices de error obtenidos en el artículo.....	66
Tabla 4.9 Resultados comparados de optimización.....	66

1. INTRODUCCIÓN Y OBJETIVO

1. INTRODUCCIÓN Y OBJETIVO

La aplicación de técnicas de detección y diagnóstico de fallos se encuentran ampliamente implementadas en procesos de producción, debido a la confiabilidad que ofrece para poder detectar y diagnosticar algún tipo de problema que no puede ser tratado por las herramientas de control existentes en la planta, consiguiéndose de esta manera saber cuándo ocurre un fallo y en qué lugar para que se pueda solucionar este problema en el tiempo más corto posible, de manera que se pueda conseguir mayor eficiencia en la producción.

La importancia de este método lleva a no solo aplicarlo en la industria productiva, sino también a diversos sectores, como en el caso del presente estudio, la distribución de agua potable, donde el objetivo es llevar el agua desde la planta de tratamiento o los tanques de almacenamiento hacia los hogares de los consumidores, manteniendo siempre un nivel de presión y calidad del agua requeridos por los clientes; de la misma manera ahora se hace énfasis en la eficiencia del uso de agua, tratando de minimizar el desperdicio de agua y la cantidad perdida de este recurso natural. Sin embargo existen problemas que pueden ocasionar que no se cumplan estos objetivos, como por ejemplo la aparición de fugas en la red de distribución, que cuando ocurren, generan una caída de presión y pérdidas en el ambiente, estas no pueden ser detectadas hasta que no sean visibles, por lo que la fuga se mantiene durante más tiempo y se pierde gran cantidad de agua.

De esta manera, es posible instalar sensores de presión en la red; que son relativamente baratos y fáciles de mantener; para hacerla análoga a una planta de producción, donde se obtienen y monitorizan datos de presión y a los cuales se puede aplicar un método de detección y diagnóstico de fallos que pueda identificar y localizar cuando y donde se genera una fuga, para que como en una planta se pueda actuar sobre la fuga rápidamente.

1. INTRODUCCIÓN Y OBJETIVO

1.1 Objetivos

El presente trabajo tiene como objetivo desarrollar un algoritmo para la detección y localización de fugas en redes de agua potable, mediante el método del Análisis del Discriminante de Fisher.

El método del análisis del discriminante de Fisher pertenece a las técnicas estadísticas multivariantes de la detección y diagnóstico de fallos que consiste en una reducción de la dimensionalidad del espacio de variables donde cada conjunto de datos pertenecientes a un fallo es una clase y los vectores de reducción tienen como objetivo maximizar la distancia entre clases mientras minimizan la distancia dentro de la clase.

El segundo objetivo es hacer una localización óptima de sensores en la red de distribución de agua, que nos diga dónde poner el mínimo número de sensores en la red de distribución de forma que se sigan detectando e identificando la mayor cantidad posibles de fugas.

En esta localización óptima del mínimo número de sensores se aplican algoritmos de optimización como los algoritmos genéticos que trabajan bastante bien con problemas de muchas variables de naturaleza no lineal y con variables enteras. De la misma manera se trabaja con estrategias evolutivas, que han ido ganando bastante terreno en la resolución de este tipo de problemas de optimización.

1.2 Organización del trabajo

El presente trabajo se organiza de la siguiente manera:

- 1. Introducción y objetivos.** En este capítulo se hace una introducción al trabajo y se contextualiza el método de detección y diagnóstico de fallos a la detección de fugas; se habla un poco sobre las técnicas que se emplean y finalmente se define el objetivo del trabajo.
- 2. Marco teórico.** En este capítulo se da una explicación teórica de lo que es la Detección y Diagnóstico de Fallos, el Análisis del Discriminante

1. INTRODUCCIÓN Y OBJETIVO

de Fisher, la optimización, los algoritmos genéticos y por último el algoritmo evolutivo de Adaptación de la Matriz de Covarianza.

3. **Metodología.** En este capítulo se explica cómo se adapta el Análisis del discriminante de Fisher para poder realizar la localización de fugas y además se explica el planteamiento del problema de optimización.
4. **Aplicación y resultados.** En este capítulo se aplica el método a las dos redes de prueba, Hanói y Limassol, a continuación se presenta los resultados y se explica cada uno de ellos. Finalmente se realiza una comparación de los resultados obtenidos con los resultados obtenidos por otros autores y mediante métodos diferentes.
5. **Conclusiones y trabajo futuro.** En este capítulo se presentan las conclusiones a las que se llegó después de haber obtenido, analizado y comparado los resultados, además de presentar líneas de trabajo con las que se podría mejorar la metodología propuesta.
6. **Bibliografía.** Por último se presentan las referencias bibliográficas a las que se acudió para la realización de este proyecto.

2. MARCO TEÓRICO

2. MARCO TEÓRICO

En el presente capítulo se explica de manera breve los conceptos relacionados con la Detección y Diagnóstico de Fallos (FDI) y con la Optimización, además de los métodos específicos que se utilizaron en este estudio.

2.1 Detección y Diagnóstico de Fallos

Los controladores estándar (PID, control predictivo, etc.) se diseñan para mantener los procesos operando satisfactoriamente, compensando los efectos de las perturbaciones y los cambios que ocurren en el proceso. Pero hay cambios que los controladores no pueden manejar adecuadamente: los fallos. Para los sistemas de seguridad crítica, las consecuencias de estos fallos pueden ser extremadamente serias en términos de vidas humanas, impacto medioambiental o pérdidas económicas. Existe por tanto una gran demanda en sistemas de supervisión y diagnosis de fallos en línea para aumentar la eficiencia, mantenibilidad, disponibilidad y fiabilidad de la planta.

Para la consecución de los objetivos mencionados relativos a la fiabilidad de los sistemas de control, es necesario dotar a dichos sistemas de un esquema de monitorización que detecte fallos incipientes o en el inicio de la degradación de un proceso para evitar el daño o pérdidas que podría suponer el desarrollo completo de los mismos, que debe estar completado con un buen mantenimiento de la planta, con la finalidad de que ciertos componentes sean reemplazados antes de que se produzca su degradación.

El objetivo de la monitorización de un proceso es asegurar el éxito de las operaciones planeadas detectando las anomalías en su comportamiento. Esta información no solo mantiene al operador de la planta y al personal de mantenimiento mejor informados del estado del proceso, sino que les tiene que ayudar a tomar las decisiones adecuadas que eliminen el comportamiento

2. MARCO TEÓRICO

anormal de dicho proceso. La monitorización por tanto consiste en los siguientes pasos:

- Detección de la presencia de un fallo a partir de información extraída de los datos medidos del proceso (módulo de detección).
- Identificación del fallo: identificar las variables más relevantes para diagnosticarlo.
- Diagnóstico del fallo: módulo que someta las características y los fallos producidos en la planta a un procedimiento de clasificación con el fin de determinar el tipo de fallo, la localización del mismo, el tamaño del fallo y la causa que lo produjo.
- Y la última etapa será la recuperación del proceso, es decir, eliminar el efecto del fallo.

Por tanto, las principales funciones a realizar por este esquema de monitorización serán las de detección, predicción, identificación y corrección del fallo en línea durante la operación del sistema. Aunque no hay que implementar todas estas operaciones en un esquema de monitorización, de hecho el objetivo de la monitorización es incorporar a los operarios e ingenieros en el lazo de monitorización de manera eficiente más que automatizar todo el esquema.

Antes de seguir adelante, vamos a ver el significado de la palabra fallo, ('fault' en la literatura anglosajona), fallo es un cambio inesperado del sistema que lleva a una degradación del comportamiento global del mismo, aunque no tiene por qué representar un defecto de ninguno de sus componentes, sino que se refiere a un mal funcionamiento del sistema que puede ser tolerable en su actual estado de desarrollo y que debe ser diagnosticado tan pronto como sea posible para prevenir consecuencias más serias. Mientras que el término avería ('failure') se refiere a un comportamiento catastrófico del sistema, es decir, se refiere a la interrupción permanente de la capacidad para mantener una función requerida del sistema.

Ejemplos de fallos que ocurren en un sistema industrial son cambios en los parámetros del proceso, como la degradación del catalizador o suciedad en

2. MARCO TEÓRICO

los intercambiadores de calor. Otro tipo de fallos se deben a las perturbaciones, como puede ser un cambio extremo en la concentración de entrada en un proceso o en la temperatura ambiente. O también fallos debidos a problemas con los sensores y/o actuadores del proceso.

2.1.1 Métodos generales de detección y diagnóstico de fallos

La idea en que se basa un esquema de monitorización es convertir los datos tomados en línea de un proceso en unas pocas medidas útiles que permitan a los operadores determinar el estado de operación del proceso y si es necesario diagnosticar los fallos. La realización de esta tarea se ha abordado fundamentalmente desde tres perspectivas distintas: métodos basados en señal, métodos analíticos y métodos basados en conocimiento.

Es importante señalar, sin embargo, que estos métodos no plantean soluciones competitivas sino complementarias. La experiencia muestra que la realidad es demasiado compleja para poder ser diagnosticada desde un único prisma óptimo.

2.1.1.1 Métodos basados en señal

Dentro de estos métodos existen diversas técnicas que pueden ser utilizadas:

- **Alarmas:** Las variables medidas son comparadas con valores límites constantes y preestablecidos, que indicarían un fallo. Estos niveles de alarma pueden ser calibrados periódicamente, e incluso variar en función de alguna condición de funcionamiento, pero en general no pueden adaptarse a la dinámica del sistema con la restricción que ello supone, es decir, el sistema puede presentar anomalías sin alcanzar los valores límite de alarmas preestablecidos.
- **Técnicas estadística multivariables:** las variables medidas en un proceso están correlacionadas con otras variables del proceso (correlación espacial) y con otras medidas de la misma variable tomadas en tiempos distintos (correlación temporal), especialmente cuando los intervalos de muestreo son pequeños. La necesidad de

2. MARCO TEÓRICO

manejar correlaciones espaciales ha permitido el desarrollo de técnicas de monitorización basadas en técnicas estadísticas multivariantes: el análisis de componentes principales (PCA), que es una técnica que reduce la dimensionalidad en términos de la varianza de los datos, el análisis del discriminante de Fisher (FDA) que es otra técnica de reducción de la dimensionalidad de un espacio de variables, o Mínimos cuadrados parciales (PLS). Todas estas técnicas se pueden utilizar para la detección y diagnóstico de fallos, como veremos a lo largo de este capítulo.

- **Redundancia física:** Se establecen múltiples componentes físicos (sensores, actuadores y componentes del sistema) distribuidos espacialmente en toda la planta para garantizar una cierta protección en presencia de algún fallo localizado. La comparación de las señales de salida de elementos supuestamente idénticos y un análisis de consistencia de los datos permiten detectar la presencia de malfunciones o fallos y la localización y aislamiento de los mismos. Tiene el problema, de que no es capaz de detectar fallos que afecten a todos los elementos de la misma manera. Además el coste económico y de espacio es elevado al duplicar todos los elementos.
- **Análisis en frecuencia.** Determinadas variables, como las vibraciones en máquina rotativa, son susceptibles de ser estudiadas en el dominio de la frecuencia, aportando gran cantidad de información sobre el estado de operación de la máquina. Cada fallo puede estar representado además por un patrón de frecuencias y amplitudes característico, lo que permite su diagnóstico.

2.1.1.2 Métodos analíticos

Son métodos de detección de fallos que emplean un modelo de la planta. La razón de su utilización es que aunque los sensores sean diferentes, las medidas que producen se derivan de un mismo estado dinámico y por lo tanto están funcionalmente relacionadas, a través de su modelo. Estas técnicas de FDI (Detección e Identificación de fallos) basadas en modelos constan de dos

2. MARCO TEÓRICO

pasos: la generación de los residuos y la toma de decisión basada en esos residuos.

En el primer paso, las entradas y las salidas del sistema se procesan mediante un algoritmo determinado para generar los residuos que nominalmente son cero, y se desvían del cero de forma característica cuando ocurre un fallo particular. Las técnicas para generar los residuos son diferentes de unos métodos a otros, fundamentalmente se diferencian en la forma de calcular el modelo del proceso, encontrándose entre las más conocidas: filtros de Kalman, ecuaciones de paridad, observadores y estimación de parámetros. En el segundo paso (toma de decisión) los residuos se examinan para determinar si hay o no hay fallos. Un proceso de decisión puede consistir en un simple test de umbral en los valores instantáneos, en la media móvil de los residuos, o pueden utilizarse los métodos de la teoría estadística de decisión.

Una característica fundamental en la generación de residuos es su robustez (insensibilidad) respecto de los errores de modelado que son inevitables. Las falsas alarmas o la incorrecta identificación de los fallos debidos al ruido, a las perturbaciones, incertidumbres de los parámetros de la planta y dinámicas no modeladas del sistema han conducido al diseño de generadores de residuos robustos y a la determinación de umbrales apropiados a cada residuo. También es posible introducir la robustez en el segundo paso, es decir, hacer una evaluación robusta de los residuos.

2.1.1.3 Métodos basados en conocimiento

Hay distintas técnicas que se pueden utilizar:

- **Sistemas Expertos.** Esta técnica es complementaria de las anteriores, en el sentido de que puede utilizar toda la información que ellas proporcionan en la detección de los fallos para analizar todos los síntomas y elaborar un diagnóstico completo. El sistema experto representa el conocimiento de los expertos en forma de reglas que relacionan los síntomas con las causas que los han provocado, e incluso pueden ofrecer soluciones o recomendaciones.

2. MARCO TEÓRICO

Las dos formas más usuales de diagnóstico son el diccionario de fallos y el árbol de decisión. El diccionario de fallos contiene una lista de causas y efectos, mientras que el árbol de decisión consiste en un conjunto estructurado de reglas lógicas del tipo “Si-síntoma-Y-síntoma-ENTONCES-conclusión”. Cada conclusión puede a su vez servir de síntoma en la siguiente regla y así sucesivamente hasta que se alcance una conclusión final.

El problema de estas técnicas, es que requieren la existencia de al menos un experto en el sistema estudiado. Dicho experto debe conocer el funcionamiento del sistema nominal y el que presenta para cada uno de los posibles fallos o situaciones extrañas que puedan presentarse. Este hecho limita las posibilidades de desarrollo para sistemas de reciente implantación. En ellos se desconoce su comportamiento anómalo, y las relaciones entre posibles síntomas y sus causas son hipótesis difíciles de validar.

- **Redes neuronales.** La aplicación de redes neuronales en diagnóstico admite varias posibilidades. El modo más directo consiste en la clasificación de los diferentes estados del sistema a partir de las variables medidas. Dichos estados se agruparán en normales y defectuosos, permitiendo clasificar estos últimos para los diferentes tipos de fallos. El aprendizaje de los diferentes modos de operación del sistema, así como los modelos de fallos se realizan mediante la presentación de muestras de ejemplos. Esto realmente entra dentro de las técnicas basadas en reconocimiento de patrones. El inconveniente de este método es que si no existen datos de fallo o son muy escasos, el modelo de red neuronal no puede aprender dicho modo de funcionamiento. Es más, la capacidad de generalización de una red neuronal podría llevar a clasificar modelos de fallo no aprendidos, como modos de comportamiento normal. Esta limitación puede superarse si la simulación de los modos de fallo por computador es suficientemente fiable.

2. MARCO TEÓRICO

Otra forma de utilizar redes neuronales en la detección y diagnóstico de fallos, es calculando un modelo nominal de la planta. Ya que una red neuronal artificial es capaz de encontrar las relaciones funcionales que representan el proceso, filtrando el ruido. La comparación entre la salida de la red neuronal y la salida del proceso, permite la detección de posibles fallos.

- **Sistemas basados en lógica difusa.** Estas técnicas se pueden utilizar tanto para la evaluación de los residuos, incorporando conocimiento del sistema en forma de reglas difusas, como en la generación de residuos, ya que de nuevo un sistema lógico difuso es un aproximador universal de funciones, en particular las funciones que relacionan las distintas variables del proceso, es decir, calculando un modelo difuso del proceso. La comparación entre la salida del sistema lógico difuso y la salida del proceso, permite la detección de posibles fallos.
- **Búsqueda causal:** se basa en el seguimiento de los fallos hasta su causa. La causalidad se representa mediante grafos directos dirigidos, donde los nodos representan las variables de estado y las ramas las relaciones entre ellas, es decir, las restricciones que se tienen que cumplir. Una búsqueda heurística por el grafo nos da la causa final del fallo (Puigjaner et al, 2010).

2.2 Análisis del Discriminante de Fisher

Para la diagnosis de fallos, los datos tomados de la planta en diferentes situaciones (operación normal y/o operación con distintos tipos de fallos) se clasifican en clases, donde cada clase representa un determinado tipo de fallo. La técnica FDA es una técnica de reducción de la dimensionalidad que es óptima en términos de maximizar la separación entre clases. Es decir, determina un conjunto de vectores de transformación lineales, ordenados en términos de maximizar la distancia entre clases y minimizar la distancia dentro de la propia clase; por tanto es una técnica muy útil para diagnosis de fallos.

Se definen n como el número de observaciones (datos) y m como el número de variables medibles, p es el número de clases, y n_j es el número de datos

2. MARCO TEÓRICO

de la clase j . Si todos los datos tomados de la planta se representan en la matriz X , es decir:

$$X = \begin{pmatrix} X_{11} & X_{12} & \dots & X_{1m} \\ X_{21} & X_{22} & \dots & X_{2m} \\ \dots & \dots & \dots & \dots \\ X_{n1} & X_{n2} & \dots & X_{nm} \end{pmatrix} \quad (1)$$

Si el vector x_i representa el conjunto de variables medibles para la observación i , este vector columna x_i es la traspuesta de la fila ' i ' de esta matriz X .

Primero se definirán algunos conceptos, como la matriz de dispersión total:

$$S_t = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (2)$$

Donde \bar{x} es el vector media total, es decir: $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

X_j es el conjunto de vectores x_i que pertenecen a la clase j . Se define la matriz de dispersión de la clase j como:

$$S_j = \sum_{x_i \in X_j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^T \quad (3)$$

Donde \bar{x}_j es el valor medio de la clase j , es decir: $\bar{x}_j = \frac{1}{n_j} \sum_{x_i \in X_j} x_i$

Se define también la matriz de dispersión dentro de la clase como:

$$S_w = \sum_{j=1}^p S_j \quad (4)$$

Y la matriz de dispersión entre clases como:

$$S_b = \sum_{j=1}^p n_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^T \quad (5)$$

2. MARCO TEÓRICO

La matriz de dispersión total, $S_t = S_b + S_w$. El objetivo del primer vector del análisis discriminante de Fisher es maximizar la dispersión entre clases y minimizar la dispersión dentro de la misma clase, es decir:

$$\max_{v \neq 0} \frac{v^T S_b v}{v^T S_w v} \quad (6)$$

Suponiendo S_w sea invertible y $v \in \mathbb{R}^m$. El objetivo del segundo vector FDA es resolver el mismo problema pero considerando los vectores perpendiculares al primer vector FDA, y así sucesivamente. Los vectores FDA son iguales a los vectores propios w_k del problema de valores propios generalizado:

$$S_b w_k = \lambda_k S_w w_k \quad (7)$$

Donde los valores propios λ_k indican el grado de separabilidad entre las clases al proyectar los datos en w_k . Esto se puede resolver siempre que S_w sea invertible, lo cual ocurre siempre que el número de datos (observaciones) n , sea mucho mayor que el número de variables medidas, m . El primer vector FDA es el que está asociado al valor propio más grande, el segundo vector FDA está asociado al segundo valor propio más grande y así sucesivamente. Un valor grande de λ_k indica que cuando los datos de la clase se proyectan en los vectores propios asociados w_k hay una gran separación de la media de la clase relativa a las varianzas de las clases y en consecuencia, una gran separación de las clases a lo largo de la dirección w_k .

Si se define la matriz $W_p \in \mathbb{R}^{m \times (p-1)}$ que contiene los $(p-1)$ vectores FDA como columnas, entonces la transformación lineal de los datos del espacio de dimensión m (con todas las variables) a este espacio de dimensión menor $(p-1)$ se describe como:

$$z_i = W_p^T x_i \quad (8)$$

Donde $z_i \in \mathbb{R}^{(p-1)}$.

El problema ahora es decidir el orden de reducción, es decir el número, p , más adecuado de vectores FDA. Esto se puede realizar de varias formas, cuando

2. MARCO TEÓRICO

se tienen suficientes datos de todas las posibles clases se puede utilizar la correlación cruzada, que consiste en dividir los datos en varios conjuntos, uno llamado de entrenamiento y los otros de test o validación. La técnica de reducción de dimensionalidad FDA se aplica a los datos de entrenamiento, y después el comportamiento del modelo obtenido se valida con los datos de test, y esto se hace para cada orden de reducción, es decir, desde, $a = 1, \dots, p$. Se elige el orden que optimice el comportamiento del modelo para los datos de validación.

Otra forma de calcular el orden cuando tenemos pocos datos, es calcular el orden a , que minimice el siguiente criterio:

$$f_m(a) + \frac{a}{\bar{n}} \quad (9)$$

Donde $f_m(a)$ es la proporción de datos mal clasificados, es decir pertenecientes a una clase y clasificados como de otra clase, para los datos de entrenamiento proyectados sobre los “ a ” vectores FDA, este término tiene en cuenta el error de predicción, mientras que el otro término de la ecuación, a/\bar{n} se añade para penalizar el incremento de dimensionalidad, es decir, tiene en cuenta la complejidad del modelo, siendo \bar{n} el número medio de observaciones por clase.

2.2.1 Detección y diagnosis de fallos

FDA puede utilizarse para detectar fallos definiendo una clase adicional que es la de condiciones normales de operación, y utilizar una función discriminante que nos diga si los datos actuales son más similares a los datos de operación normal o a los datos de alguna las otras clases (posibles fallos). Esto se podrá utilizar siempre y cuando los datos de entrenamiento usados para calcular los FDA de la operación normal sean muy distintos a los datos recogidos de la planta cuando hay fallos. Se asigna una observación a la clase i , si se cumple que $g_i(x) > g_j(x) \forall j \neq i$, y donde a la función $g_j(x)$ se le llama función discriminante para la clase j dado un vector de datos x . La función,

2. MARCO TEÓRICO

$g_i(x)$ que minimiza el error, cuando ocurre el evento w_i (por ejemplo un fallo) es:

$$g_i(x) = P(w_i | x) \quad (10)$$

donde $P(w_i | x)$ es la probabilidad a posterior de que x pertenezca a la clase i . Si se usa la regla de Bayes:

$$P(w_i | x) = \frac{p(x|w_i)P(w_i)}{p(x)} \quad (11)$$

Donde $P(w_i)$ es la probabilidad a priori para la clase w_i , $p(x)$ es la función densidad de probabilidad para x , y $p(x|w_i)$ es la función densidad de probabilidad de x condicionada al evento w_i . Si se supone que los datos están normalmente distribuidos:

$$P(x|w_i) = \frac{1}{(2\pi)^{m/2} [\det(\Sigma_i)]^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right] \quad (12)$$

Con m el número de variables medidas, y μ_i y Σ_i son la media y la matriz de covarianza para la clase i . La función $g_i(x)$ de la ecuación (10) puede ser sustituida por la ecuación (12), y se puede demostrar que la clasificación obtenida con ambas funciones es idéntica:

$$g_i(x) = \ln p(x | w_i) + \ln P(w_i) \quad (13)$$

Si se sustituye la ecuación (12) en la ecuación (13) se obtiene finalmente que la función discriminante es:

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) - \frac{m}{2} \ln 2\pi - \frac{1}{2} \ln[\det(\Sigma_i)] + \ln P(w_i) \quad (14)$$

Si caracterizamos esta función, para este caso, es decir, sea $W_a \in \mathbb{R}^{m \times a}$ la matriz formada por los vectores propios w_1, w_2, \dots, w_a calculados con la ecuación (7), la función discriminante puede calcularse como:

2. MARCO TEÓRICO

$$g_j(x) = -\frac{1}{2}(x - \bar{x}_j)W_a \left(\frac{1}{n_j - 1} W_a^T S_j W_a \right)^{-1} W_a^T (x - \bar{x}_j) + \ln(p_i) - \frac{1}{2} \ln \left[\det \left(\frac{1}{n_j - 1} W_a^T S_j W_a \right) \right] \quad (15)$$

Donde S_j , \bar{x}_j y n_j ya han sido definidas (Puigjaner et al, 2010).

2.3 Optimización

Dentro de la investigación operativa, las técnicas de optimización se enfocan en determinar la política a seguir para maximizar o minimizar la respuesta del sistema. Dicha respuesta, en general, es un indicador del tipo “Costo”, “Producción”, “Ganancia”, etc., la cual es una función de la política seleccionada. Dicha respuesta se denomina objetivo, y la función asociada se llama función objetivo.

Pero, ¿qué entendemos cómo política? Una política es un determinado conjunto de valores que toman los factores que podemos controlar a fin de regular el rendimiento del sistema. Es decir, son las variables independientes de la función que la respuesta del sistema.

Por ejemplo, supongamos que deseamos definir un mix de cantidad de operarios y horas trabajadas para producir lo máximo posible en una planta de trabajos por lotes. En este ejemplo, nuestro objetivo es la producción de la planta, y las variables de decisión son la cantidad de operarios y la cantidad de horas trabajadas. Otros factores afectan a la producción, como la productividad de los operarios, pero los mismos no son controlables por el tomador de decisiones. Este último tipo de factores se denominan parámetros.

En general, el quid de la cuestión en los problemas de optimización radica en que estamos limitados en nuestro poder de decisión. Por ejemplo, podemos tener un tope máximo a la cantidad de horas trabajadas, al igual que un rango de cantidad de operarios entre los que nos podemos manejar. Estas limitaciones, llamadas “restricciones”, reducen la cantidad de alternativas

2. MARCO TEÓRICO

posibles, definiendo un espacio acotado de soluciones factibles (y complicando, de paso, la resolución del problema). Observen que acabamos de hablar de “soluciones factibles”. Y es que, en optimización, cualquier vector con componentes apareadas a las variables independientes que satisfaga las restricciones, es una solución factible, es decir, una política que es posible implementar en el sistema. Dentro de las soluciones factibles, pueden existir una o más soluciones óptimas, es decir, aquellas que, además de cumplir con todas las restricciones, maximizan (o minimizan, según sea el problema a resolver) el valor de la función objetivo.

Resumiendo, un problema de optimización está compuesto de los siguientes elementos:

- Un conjunto de restricciones.
- Un conjunto de soluciones factibles, el cual contiene todas las posibles combinaciones de valores de variables independientes que satisfacen las restricciones anteriores.
- Una función objetivo, que vincula las soluciones factibles con el comportamiento del sistema (Baquela and Redchuk, 2013).

2.4 Algoritmos Genéticos

Los Algoritmos Genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio de supervivencia del más apto.

Más formalmente, y siguiendo la definición dada por Goldberg, “los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas”.

Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de

2. MARCO TEÓRICO

estos individuos representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida.

2.4.1 Bases biológicas

En la naturaleza, los individuos de una población compiten constantemente con otros por recursos tales como comida, agua y refugio. Los individuos que tienen más éxito en la lucha por los recursos tienen mayores probabilidades de sobrevivir y generalmente una descendencia mayor. Al contrario, los individuos peor adaptados tienen un menor número de descendientes, o incluso ninguno. Esto implica que los genes de los individuos mejor adaptados se propagarán a un número cada vez mayor de individuos de las sucesivas generaciones.

La combinación de características buenas de diferentes ancestros puede originar, en ocasiones, que la descendencia esté incluso mejor adaptada al medio que los padres. De esta manera, las especies evolucionan adaptándose más y más al entorno a medida que transcurren las generaciones.

Pero la adaptación de un individuo al medio no sólo está determinada por su composición genética. Influyen otros factores como el aprendizaje, en ocasiones adquirido por el método de prueba y error, en ocasiones adquirido por imitación del comportamiento de los padres, la cultura, la religión, etc. aspectos no contemplados en el Algoritmo Genético clásico, pero sí en otras variantes.

2.4.2 Codificación de problemas

Cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros. El conjunto de todos los parámetros (genes en la terminología de Algoritmos Genéticos) se codifica en una cadena de valores denominada cromosoma (ver Figura 2.1).

2. MARCO TEÓRICO

El conjunto de los parámetros representado por un cromosoma particular recibe el nombre de genotipo. El genotipo contiene la información necesaria para la construcción del organismo, es decir, la solución real al problema, denominada fenotipo. Por ejemplo, en términos biológicos, la información genética contenida en el ADN de un individuo sería el genotipo, mientras que la expresión de ese ADN (el propio individuo) sería el fenotipo.

Desde los primeros trabajos de John Holland la codificación suele hacerse mediante valores binarios. Se asigna un determinado número de bits a cada parámetro y se realiza una discretización de la variable representada por cada gen. El número de bits asignados dependerá del grado de ajuste que se desee alcanzar. Evidentemente no todos los parámetros tienen por qué estar codificados con el mismo número de bits. Cada uno de los bits pertenecientes a un gen suele recibir el nombre de alelo.

En la Figura 2.1 se muestra un ejemplo de un individuo binario que codifica 3 parámetros.

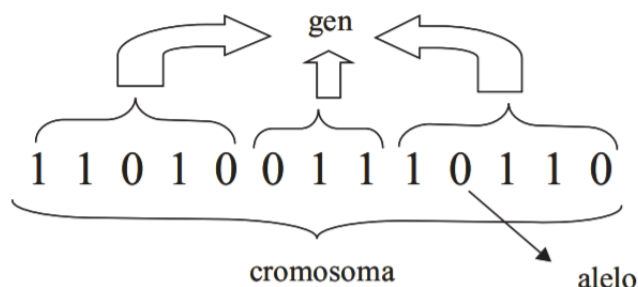


Figura 2.1 Individuo Genético Binario

Sin embargo, también existen representaciones que codifican directamente cada parámetro con un valor entero, real o en punto flotante. A pesar de que se acusa a estas representaciones de degradar el paralelismo implícito de las representaciones binarias, permiten el desarrollo de operadores genéticos más específicos al campo de aplicación del Algoritmo Genético.

En la Figura 2.2, se muestra un ejemplo típico de la aplicación de los Algoritmos Genéticos: cómo representar una red de neuronas artificiales para posteriormente realizar el proceso de optimización de los pesos sinápticos.

2. MARCO TEÓRICO

Codificar una red de neuronas artificiales en forma de cromosoma es tan sencillo como asignar un gen del cromosoma a cada uno de los pesos de la red. También se podrían añadir genes que indiquen el número de capas y el número de elementos de procesado en cada una.

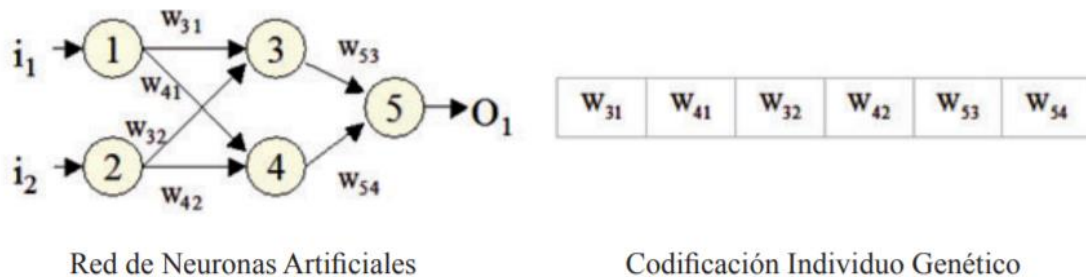


Figura 2.2 Codificación de una red de neuronas artificiales

2.4.3 Operadores genéticos

Para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce, copia y mutación. En el caso de no trabajar con una población intermedia temporal también cobran relevancia los algoritmos de reemplazo. A continuación se verán en mayor detalle.

2.4.3.1 Selección

Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto, la selección de un individuo estará relacionada con su valor de ajuste. No se debe, sin embargo, eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea.

En cuanto a algoritmos de selección se refiere, estos pueden ser divididos en dos grandes grupos: probabilísticos y determinísticos. Ambos tipos de algoritmos basan su funcionamiento en el principio indicado anteriormente

2. MARCO TEÓRICO

(permitir escoger una mayor cantidad de veces a los más aptos). Sin embargo, como su nombre indica, el primer tipo adjudica estas posibilidades con un importante componente basado en el azar. Es en este grupo donde se encuentran los algoritmos de selección por ruleta o por torneo que, dado su importancia por ser los más frecuentemente utilizados, se describen con detalle en esta sección. El segundo grupo engloba una serie de algoritmos que, dado el ajuste conocido de cada individuo, permite asignar a cada uno el número de veces que será escogido para reproducirse. Esto puede evitar problemas de predominancia de ciertos individuos y cada uno de estos algoritmos presentan variaciones respecto al número de veces que se tomarán los mejores y peores y, de esta forma, se impondrá una presión en la búsqueda en el espacio de estados en la zona donde se encuentra el mejor individuo (en el caso de que se seleccionen más veces los mejores), o bien que se tienda a repartir la búsqueda por el espacio de estados, pero sin dejar de tender a buscar en la mejor zona (caso de repartir más la selección). Algunos de estos algoritmos son sobranste estocástico, universal estocástica o muestreo determinístico.

Una opción bastante común consiste en seleccionar el primero de los individuos participantes en el cruce mediante alguno de los métodos expuestos en esta sección y el segundo de manera aleatoria.

- **Selección por ruleta.** Propuesto por De Jong, es posiblemente el método más utilizado desde los orígenes de los Algoritmos Genéticos. A cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente, la población está ordenada en base al ajuste, por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo $[0..1]$ y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la

2. MARCO TEÓRICO

población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

Es un método muy sencillo pero ineficiente a medida que aumenta el tamaño de la población (su complejidad es $O(n^2)$). Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez.

- **Selección por torneo.** La idea principal de este método de selección consiste en escoger a los individuos genéticos en base a comparaciones directas entre sus genotipos.

Existen dos versiones de selección mediante torneo, el torneo determinístico y el torneo probabilístico, que a continuación pasan a detallarse. En la versión determinística se selecciona al azar un número p de individuos (generalmente se escoge $p=2$). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

La versión probabilística únicamente se diferencia en el paso de selección del ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo $[0..1]$, si es mayor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo más alto y en caso contrario el menos apto. Generalmente p toma valores en el rango $0.5 < p \leq 1$.

Variando el número de individuos que participan en cada torneo se puede modificar la presión de selección. Cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. Un caso particular es el elitismo global. Se trata de un torneo en el que participan todos los individuos de la población, con lo cual la selección se vuelve totalmente determinística. Cuando el tamaño del torneo es reducido, la presión de selección disminuye y los peores individuos tienen más oportunidades de ser seleccionados.

Elegir uno u otro método de selección determinará la estrategia de búsqueda del Algoritmo Genético. Si se opta por un método con una alta

2. MARCO TEÓRICO

presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Existen muchos otros algoritmos de selección. Unos buscan mejorar la eficiencia computacional, otros el número de veces que los mejores o peores individuos pueden ser seleccionados. Algunos de estos algoritmos son muestreo determinístico, escalamiento sigma, selección por jerarquías, estado uniforme, sobrante estocástico, brecha generacional, etc.

2.4.3.2 Cruce

Una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación. Tal y como se ha indicado anteriormente, el cruce es una estrategia de reproducción sexual.

Su importancia para la transición entre generaciones es elevada puesto que las tasas de cruce con las que se suele trabajar rondan el 90%. Los diferentes métodos de cruce podrán operar de dos formas diferentes. Si se opta por una estrategia destructiva los descendientes se insertarán en la población temporal aunque sus padres tengan mejor ajuste (trabajando con una única población esta comparación se realizará con los individuos a reemplazar).

Por el contrario, utilizando una estrategia no destructiva la descendencia pasará a la siguiente generación únicamente si supera la bondad del ajuste de los padres (o de los individuos a reemplazar). La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de los padres. Al compartir las características buenas de dos individuos, la descendencia, o al menos parte de ella, debería tener una bondad mayor que cada uno de los padres por separado. Si el cruce no agrupa las mejores características en uno de los hijos y la descendencia tiene un peor

2. MARCO TEÓRICO

ajuste que los padres no significa que se esté dando un paso atrás. Optando por una estrategia de cruce no destructiva garantizamos que pasen a la siguiente generación los mejores individuos. Si, aún con un ajuste peor, se opta por insertar a la descendencia, y puesto que los genes de los padres continuarán en la población –aunque dispersos y posiblemente levemente modificados por la mutación–, en posteriores cruces se podrán volver a obtener estos padres, recuperando así la bondad previamente pérdida.

Existen multitud de algoritmos de cruce. Sin embargo los más empleados son los que se detallarán a continuación:

- **Cruce de 1 punto.** Es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres.

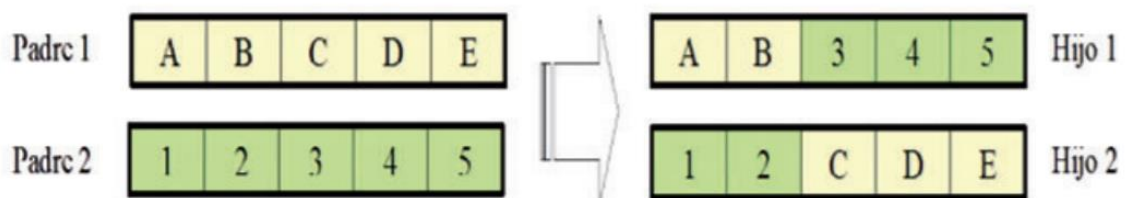


Figura 2.3 Cruce de un punto

En la Figura 2.3 se puede ver con claridad el proceso descrito anteriormente. En la bibliografía suele referirse a este tipo de cruce con el nombre de SPX (Single Point Exchange)

- **Cruce de 2 puntos.** Se trata de una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres, como en el caso anterior, se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres

2. MARCO TEÓRICO

segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre.

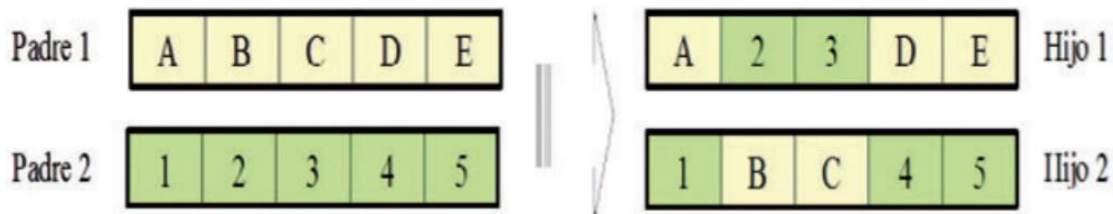


Figura 2.4 Cruce de dos puntos

Generalmente, es habitual referirse a este tipo de cruce con las siglas DPX (Double Point Crossover). En la Figura 2.4 se muestra un ejemplo de cruce por dos puntos. Generalizando, se pueden añadir más puntos de cruce dando lugar a algoritmos de cruce multipunto. Sin embargo existen estudios que desaprueban esta técnica. Aunque se admite que el cruce de 2 puntos aporta una sustancial mejora con respecto al cruce de un solo punto, el hecho de añadir un mayor número de puntos de cruce reduce el rendimiento del Algoritmo Genético. El problema principal de añadir nuevos puntos de cruce radica en que es más fácil que los segmentos originados sean corrompibles, es decir, que por separado quizás pierdan las características de bondad que poseían conjuntamente. Hay que evitar, por lo tanto, romper dichos segmentos, denominados bloques constructivos. Sin embargo no todo son desventajas y añadiendo más puntos de cruce se consigue que el espacio de búsqueda del problema sea explorado con más intensidad.

- **Cruce uniforme.** El cruce uniforme es una técnica completamente diferente de las vistas hasta el momento. Cada gen de la descendencia tiene las mismas probabilidades de pertenecer a uno u otro padre.

2. MARCO TEÓRICO

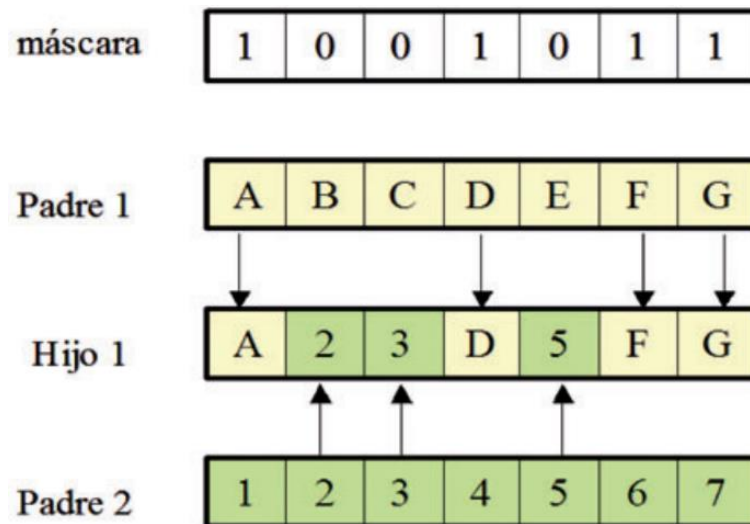


Figura 2.5 Cruce uniforme

Aunque se puede implementar de muy diversas formas, la técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen situado en esa posición en uno de los descendientes se copia del primer padre. Si por el contrario hay un 0 el gen se copia del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

Tal y como se puede apreciar en la Figura 2.5, la descendencia contiene una mezcla de genes de cada uno de los padres. El número efectivo de puntos de cruce es fijo pero será por término medio $L/2$, siendo L la longitud del cromosoma (número de alelos en representaciones binarias o de genes en otro tipo de representaciones).

La máscara de cruce puede no permanecer fija durante todo el proceso evolutivo. Se genera de manera aleatoria para cada cruce. Se suele referir a este tipo de cruce con las siglas UPX (Uniform Point Crossover).

- **Cruces específicos de codificaciones no binarias.** Los tres tipos de cruce vistos hasta el momento son válidos para cualquier tipo de representación del genotipo. Si se emplean genotipos compuestos por

2. MARCO TEÓRICO

valores enteros o reales pueden definirse otro tipo de operadores de cruce:

- Media: el gen de la descendencia toma el valor medio de los genes de los padres. Tiene la desventaja de que únicamente se genera un descendiente en el cruce de dos padres.
- Media geométrica: cada gen de la descendencia toma como valor la raíz cuadrada del producto de los genes de los padres. Presenta el problema añadido de qué signo dar al resultado si los padres tienen signos diferentes.
- Extensión: se toma la diferencia existente entre los genes situados en las mismas posiciones de los padres y se suma al valor más alto o se resta del valor más bajo. Solventa el problema de generar un único descendiente. Una variante basada en este cruce son los Algoritmos genéticos de evolución diferencial.

2.4.3.3 Algoritmos de Reemplazo

Cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, deberá tenerse en cuenta que para insertar un nuevo individuo deberá de eliminarse previamente otro de la población. Existen diferentes métodos de reemplazo:

- Aleatorio: el nuevo individuo se inserta en un lugar escogido de manera aleatoria en la población.
- Reemplazo de padres: se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.
- Reemplazo de similares: una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.
- Reemplazo de los peores: de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

2. MARCO TEÓRICO

2.4.3.4 Copia

La copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación.

El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de ese individuo.

Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce y, si éste finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados.

2.4.3.5 Elitismo

El elitismo es un caso particular del operador de copia consistente en copiar siempre al mejor, o en su caso, mejores individuos de una generación en la generación siguiente. De esta manera se garantiza que el proceso de búsqueda nunca dará un paso atrás en cuanto a la calidad de la mejor solución obtenida, sino que un cambio en ésta siempre implicará una mejora.

Una variación de este proceso consiste en copiar al mejor o mejores individuos de una generación en la siguiente, únicamente cuando tras el paso de una generación no se haya mejorado con los operadores de cruce o mutación la mejor solución de la generación actual.

2.4.3.6 Mutación

La mutación de un individuo provoca que alguno de sus genes, generalmente uno sólo, varíe su valor de forma aleatoria.

Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la mutación se

2. MARCO TEÓRICO

suele utilizar de manera conjunta con el operador de cruce. Primeramente se seleccionan dos individuos de la población para realizar el cruce. Si el cruce tiene éxito entonces uno de los descendientes, o ambos, se muta con cierta probabilidad P_m . Se imita de esta manera el comportamiento que se da en la naturaleza, pues cuando se genera la descendencia siempre se produce algún tipo de error, por lo general sin mayor trascendencia, en el paso de la carga genética de padres a hijos.

La probabilidad de mutación es muy baja, generalmente menor al 1%. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de mutados. Sin embargo se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga una probabilidad nula de ser examinado.

Tal y como se ha comentado, la mutación más usual es el reemplazo aleatorio. Este consiste en variar aleatoriamente un gen de un cromosoma. Si se trabaja con codificaciones binarias, consistirá simplemente en negar un bit. También es posible realizar la mutación intercambiando los valores de dos alelos del cromosoma. Con otro tipo de codificaciones no binarias existen otras opciones:

- Incrementar o decrementar a un gen una pequeña cantidad generada aleatoriamente.
- Multiplicar un gen por un valor aleatorio próximo a 1.

Aunque no es lo más común, existen implementaciones de Algoritmos Genéticos en las que no todos los individuos tienen los cromosomas de la misma longitud. Esto implica que no todos ellos codifican el mismo conjunto de variables. En este caso existen mutaciones adicionales como puede ser añadir un nuevo gen o eliminar uno ya existente (Gestal, 2010).

2.5 Estrategias Evolutivas

Las estrategias evolutivas son técnicas heurísticas que imitan los procesos evolutivos de las especies en la naturaleza, y que han sido utilizadas en la solución de problemas de optimización. Están basadas en una población de

2. MARCO TEÓRICO

individuos, y utilizan los operadores de recombinación, mutación y selección en la búsqueda de la mejor solución en variables reales.

Una de las estrategias evolutivas más conocidas es $EE-(\mu+\lambda)$, donde un conjunto de μ padres se utiliza para generar λ hijos en cada generación. En dicha estrategia, los μ padres junto con todos sus λ hijos son reunidos en un mismo conjunto del cual se seleccionan los individuos mejor evaluados, es decir, con mejor aptitud. Para la generación de los λ hijos, se utilizan los operadores de recombinación y mutación aplicados sobre los μ padres.

La recombinación consiste en el intercambio de información entre las variables objetivo de los padres. La mutación consiste en realizar pequeños cambios aleatorios en las variables objetivo. Las mutaciones en las estrategias evolutivas usan un control sobre el tamaño del cambio aleatorio (desviación estándar). Dichas desviaciones estándar (llamadas variables de control) forman parte de la información de cada uno de los individuos de la población, y por lo tanto también se les aplican los operadores de recombinación y mutación para modificarlas, teniéndose entonces un proceso auto-adaptable. En cada individuo, se tiene una variable de control por cada variable objetivo. El algoritmo general de una estrategia evolutiva es el siguiente:

Iniciar con una población de μ padres y evaluar su aptitud, y repetir desde $t=0$ hasta $t= max_gen$.

- Recombinación: producir λ hijos a partir de los μ padres, recombinando variables de control y variables objetivo.
- Mutación: realizar una mutación a cada uno de los λ hijos. Primero deben mutarse las variables de control, y luego se hace la mutación de las variables objetivo.
- Evaluación: calcular la aptitud de cada hijo.
- Selección: Seleccionar los μ individuos que pasan a la siguiente generación.

2. MARCO TEÓRICO

Puede añadirse al algoritmo criterios de paro adicionales al número máximo de generaciones, basados en estancamiento y/o convergencia de la solución (Botello et al, 2007)).

2.5.1 Algoritmo CMA-ES

Las estrategias evolutivas usadas para optimizaciones de valores reales, usualmente recaen en variaciones aleatorias Gaussianas, por lo que adaptar de manera apropiada las matrices de covarianza de estas mutaciones durante la optimización, permite el aprendizaje y el empleo de una medida variable para la distribución de búsqueda. De esta manera realiza la adaptación automática de la distribución mutada mejora drásticamente el rendimiento de la búsqueda sobre las funciones objetivo no separables y mal escaladas. La Adaptación de la Matriz de Covarianza (CMA por sus siglas en inglés) emplea este concepto de adaptación automática y ha probado ser un mecanismo de adaptación efectivo para las estrategias evolutivas (Igel et al, 2006).

El (μ, λ) CMA-ES es un algoritmo que emplea una función de distribución de probabilidad para generar μ soluciones, y, posteriormente, emplea las λ mejores para refinar la propia función de distribución de probabilidad para producir mejores soluciones, en un proceso iterativo. Finalmente, el algoritmo devuelve la mejor solución encontrada.

Este método requiere únicamente dos parámetros de entrada: el centro inicial de la distribución, y el valor σ que es la varianza inicial de la distribución. Para el resto de parámetros de la búsqueda los autores proponen valores por defecto (Molina et al, 2007).

3. METODOLOGÍA

En este capítulo se explica la metodología seguida para el uso del método del Análisis del Discriminante de Fisher y su aplicación a la localización de fugas y la posterior aplicación de los algoritmos de optimización para llegar a formular el problema de minimización que nos de la localización de los sensores de forma que usando la menor cantidad posible de sensores se sigan detectando y localizando las fugas existentes en la red.

3.1 Metodología de localización de fugas

La metodología consiste en la aplicación del método del Análisis Discriminante de Fisher sobre un conjunto de datos, los cuales pueden ser presiones, residuos o sensibilidades. Los valores de presión p_i son las medidas de presión existentes en la red, que en este caso son obtenidos directamente de la simulación, el residuo R_i es la diferencia entre el valor de presión medida y su correspondiente estimación, \hat{p}_i , usando un modelo hidráulico de la red sin fugas. En esta aplicación, el residuo se calculará usando la ecuación (16), como la diferencia entre la presión estimada por el modelo en situación de fuga y el valor de presión estimada por el mismo modelo en situación normal, finalmente la sensibilidad $Sens_i$ se obtiene de restar el valor de la presión en situación de fuga menos el valor de la presión en situación normal y se la divide entre el valor de la magnitud de la fuga.

$$R_i = \hat{p}_i^{f_j} - \hat{p}_i \quad (16)$$

$$Sens_i = \frac{\hat{p}_i^{f_j} - \hat{p}_i}{f_j} \quad (17)$$

Donde:

f_j representa a las diferentes posibles magnitudes que se pueden dar en las fugas en el nodo j , donde:

3. METODOLOGÍA

$i=1,\dots,n$, siendo n el número de medidas de presión obtenidas por los sensores existentes en la red, en este caso es igual al número de nodos.

$j = 1,\dots, J$ donde J es el número de fugas que se pueden dar, y que sin pérdida de generalidad es este caso suponemos que es igual al número de nodos de la red, n .

Por otro lado cualquiera de estas medidas, valor de la presión, residuo o sensibilidad se pueden recolectar en distintos instantes de tiempo, es decir tenemos valores desde $k=1,\dots, m$, donde m es el número de observaciones que se recogen de la red.

Para la elección del conjunto de datos a usar, se aplica la metodología que sigue a continuación, primero se obtienen $j=n$ matrices de datos, una por cada posible fuga a considerar, de tamaño $m \times n$, (m observaciones \times n variables), es decir una matriz de datos para cada uno de los nodos donde se puede dar una fuga.

$$D_j = \begin{bmatrix} D_{1,1} & \dots & D_{1,n} \\ \vdots & \ddots & \vdots \\ D_{m,1} & \dots & D_{m,n} \end{bmatrix} \quad (18)$$

A partir de la obtención de esta matriz, se empieza a calcular los parámetros necesarios para la aplicación del Análisis del Discriminante de Fisher, para este fin se define n tipos de clases, una por cada fallo que se pueda dar en un nodo y se procede a calcular la media y la dispersión de cada clase:

$$\bar{x}_j = \frac{1}{m} \sum_i^m x_i \quad (19)$$

$$S_j = \sum_i^m (x_i - \bar{x}_i) (x_i - \bar{x}_i)^T \quad (20)$$

A partir de la matriz de dispersión de cada clase, calculamos la matriz de dispersión entre cada clase, que consiste en sumar cada una de las n matrices de dispersión.

3. METODOLOGÍA

$$S_w = \sum_j^n S_j \quad (21)$$

Entonces se procede a calcular la matriz total de datos que se denominará X, esto se logra al concatenar cada una de las matrices de datos D_j , donde se obtiene una matriz de (mxn) filas y n columnas.

$$X = \begin{bmatrix} D \\ \vdots \\ D_j \end{bmatrix} \quad (22)$$

Entonces de la matriz X se calculan la media de la matriz y la dispersión total S_t , de manera similar a la que se realiza en cada clase. De la misma manera se calcula la matriz de dispersión entre clases S_b

$$\bar{x} = \frac{1}{m * n} \sum_i^{m*n} x_i \quad (23)$$

$$S_t = \sum_i^{m*n} (x_i - \bar{x})(x_i - \bar{x})^T \quad (24)$$

$$S_b = S_t - S_w \quad (25)$$

El siguiente paso es calcular los autovalores y autovectores que definen la reducción de dimensionalidad del problema, esto se obtiene al resolver la siguiente ecuación:

$$W_{ka}, \lambda, W_k = eig(S_b, S_w) \quad (26)$$

Donde λ y W_k son las matrices que contienen a los autovalores y a los autovectores respectivamente, que definen la reducción de dimensionalidad, para decidir que autovectores se utilizan para la reducción de la dimensionalidad, se toma aquellos autovectores, cuyos autovalores sumados dan un valor del 95% del total de la suma de los mismos, esto equivale a ordenar los valores de la matriz λ de mayor a menor y empezar a sumarlos hasta que el resultado sea el 95% de la suma de todos los autovalores. Una vez conseguido esto, se toma las columnas de los autovectores que corresponden al número de autovalores usados, obteniéndose la matriz W_a de dimensión (nxa), tal que:

3. METODOLOGÍA

$$\text{tal que } \sum_i^a \lambda_i \geq 0.95 \sum_i^n \lambda_i \quad (27)$$

A continuación lo que se hace es probar el método calculado para la localización de las fugas, es decir, si el método FDA calculado es capaz de distinguir entre las diferentes clases, recordando que cada clase es un fallo. Para este fin primero se debe obtener una matriz de datos de fuga de una magnitud diferente a la que se usó para el cálculo de los parámetros anteriores, a partir de aquí se calcula la función discriminante para cada una de las clases posibles.

$$\begin{aligned} g_{j,i} &= -\frac{1}{2}(x - \bar{x}_j)W_a \left(\frac{1}{n_j - 1} W_a^T S_j W_a \right)^{-1} W_a^T (x - \bar{x}_j) - \\ &= -\frac{1}{2} \ln \left[\det \left(\frac{1}{n_j - 1} W_a^T S_j W_a \right)^{-1} \right] \end{aligned} \quad (28)$$

La función discriminante $g_{j,i}$ obtenida con la ecuación anterior nos da un conjunto de valores g_j para cada instante de tiempo i , que nos indican que tan parecido es el conjunto de datos que se está probando a cada una de las clases, por lo que el valor máximo de la sumatoria de los $g_{j,i}$ indica donde se encuentra ubicada la fuga, por lo tanto, cuando el valor máximo del discriminante se dé en la clase j y esta coincida con el número de nodo en el que se generó la fuga, entonces la localización es correcta.

$$g_k = \max \left(\sum_i^m g_{1,i}, \sum_i^m g_{2,i}, \dots, \sum_i^m g_{j,i} \right) \quad (29)$$

Siendo k el nodo donde se generó la fuga.

Es decir, el máximo de la ecuación (29) nos indica donde está la fuga, si este valor coincide con el nodo donde produjimos el fallo, la identificación es correcta, y el método es capaz de detectar e identificar los fallos en el sistema.

3.2 Metodología para la optimización del uso de sensores

Una vez conseguida la localización de las fugas, ahora se debe resolver el problema de minimizar la cantidad de sensores que se emplazaran en el

3. METODOLOGÍA

terreno, y su localización, es decir, donde ponerlos en la red de distribución de agua, de forma que se maximice la capacidad de localización de los mismos, a través del método descrito en la sección anterior, es decir el método FDA.

Partiendo de la matriz de datos D (ecuación 18) con n igual al número de nodos de la red y por lo tanto, un sensor de presión existente en cada nodo, se debe buscar una configuración, tal que, al quitar sensores de presión de los nodos, el método aun sea capaz de localizar las fugas, para esto se genera un vector binario q que representará la configuración que será puesta a prueba.

$$q = [q_1 \quad \dots \quad q_n] \quad (30)$$

Donde $q_i = 1$ si existe un sensor en ese nodo o $q_i = 0$ en el caso contrario, de esta manera habrá tantos q_i iguales a uno que sensores instalados en la red, a partir de este vector se define otro vector z que contiene el número del nodo (la localización del mismo) donde se encuentra instalado el sensor.

$$z = [i_{q1} \quad \dots \quad i_{qn}] \quad (31)$$

Así, los valores del vector z nos indican que columnas de la matriz D_j (ecuación 18) se utilizarán para realizar el cálculo de los parámetros necesarios para FDA, quedando como resultado D_q que contendrá los valores de las medidas de los nodos donde existe un sensor instalado para todos los instantes de tiempo, para cada fuga, después se concatenan las correspondientes matrices D_q una para cada fallo, para obtener a matriz total X_q (ecuación 22) para aplicar el método FDA. De esta manera se calcula:

- La media de cada clase, x_j
- La dispersión de la clase j , S_j
- La dispersión dentro las clases, S_w
- La dispersión total, S_t
- La dispersión entre cada clases, S_b
- Los autovalores y autovectores.

3. METODOLOGÍA

Con estos valores se aplica el método a una matriz de datos nueva donde hay una fuga de tamaño diferente al del entrenamiento en alguno de los nodos, y que al igual que en el caso de entrenamiento, solamente contiene valores para los nodos donde se tiene sensores instalados, el resultado de esto es la obtención del discriminante de Fisher en forma de un vector g_j , donde si el máximo de dicha función (ecuación 29) coincide con el nodo donde se produjo la fuga, entonces el método localiza la fuga y el error de diagnóstico es cero, porque es la situación correcta. En el caso contrario, si el máximo de g_{ij} no coincide con el nodo donde se produjo la fuga, el método no identifica bien el fallo, y se produce un error. Para evaluar la capacidad de la configuración de sensores para localizar fugas, se introduce el siguiente índice de error:

$$e_j = \begin{cases} 0 & \text{si } g_k = \max \left(\sum_{i=1}^m g_{1,i}, \sum_{i=1}^m g_{2,i}, \dots, \sum_{i=1}^m g_{j,i} \right) \\ 1 & \text{en el caso contrario} \end{cases} \quad (32)$$

Esto significa que el índice de error e_j será 0 cuando la fuga sea localizada correctamente en el índice donde se introdujo la fuga, o tomará un valor de 1 cuando suceda lo contrario.

Como el objetivo es maximizar la capacidad de localización de las fugas, cuando estas se den en cada uno de los nodos de la red, se introduce un índice de error que considera a las fugas de cada nodo para su cálculo.

$$\bar{e} = \sum_{j=1}^n \frac{e_j}{n} \quad (33)$$

Este índice de error \bar{e} tomara un valor de 0 cuando todas las fugas sean correctamente localizadas con la configuración probada, o tomará una valor de $100 \cdot \bar{e}$ que será el porcentaje de fugas incorrectamente localizadas, de esta manera se puede plantear un problema de optimización:

3. METODOLOGÍA

$$\min \bar{e}$$

$$s. a \quad (34)$$

$$\sum_{i=1}^m q_i = \text{sensores}$$

3.3 Medidas de robustez

Para conseguir que el método sea más robusto, se realizan tres modificaciones en el algoritmo de localización, la primera tiene que ver con hacer la asignación del error una función lineal de la distancia desde donde se encuentra la fuga hasta donde se la localiza, la segunda consiste en considerar todas las magnitudes de los fallos cuando se realiza la búsqueda de la configuración óptima y finalmente se adiciona un ruido blanco a la medición obtenida por los sensores.

3.3.1 Distancia

Se propone que el error asignado a la localización sea una función lineal de la distancia, ya que cuando el método localiza una fuga en un nodo contiguo a la ubicación real de la fuga, es interesante no asignar un error de 1 a este resultado, por lo que se propone la creación de un área de precisión, que sería un área circular alrededor del nodo, donde se considera cierto grado de eficacia del método si la localización de la fuga cae dentro de esta superficie; el tamaño de esta circunferencia dependerá del tamaño de la red y está dado por una distancia límite y por una distancia máxima.

3. METODOLOGÍA

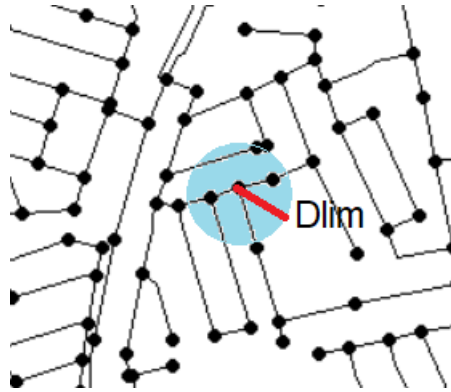


Figura 3.1 Representación de la distancia límite

La distancia límite representa el radio de la circunferencia y delimita hasta donde se aplicará una puntuación entre 0 y 1 cuando la fuga sea localizada dentro de su área y 1 en el caso contrario.

La distancia máxima depende de la cantidad de nodos que tenga la red y representa la cantidad de nodos hasta donde se considera importante asignar un error menor a 1; estaría definida por $\frac{1}{2}\sqrt{n}$. Entonces la distancia límite llegaría a ser definida como d_{max} veces la media de la distancia del nodo i a su nodo más cercano; expresado matemáticamente sería:

$$d_{max} = \frac{1}{2}\sqrt{n} \quad (35)$$

$$d_{lim} = d_{max} \frac{1}{n} \sum_{i=1}^n d_i \quad (36)$$

Donde:

n es la cantidad de nodos que tiene la red

d_i es la distancia entre el nodo i al nodo más cercano.

A partir de esto, la ecuación la ecuación 32 se modifica como sigue:

$$e_j = \begin{cases} \frac{d_{ji}}{d_{lim}} & \text{si } d_{ji} < d_{lim} \\ 1 & \text{en el caso contrario} \end{cases} \quad (37)$$

3. METODOLOGÍA

Donde d_{ji} es la distancia del nodo j donde se encuentra la fuga, al nodo i que presenta el valor máximo del Discriminante de Fisher: $\max(\sum_{i=1}^m g_{1,i}, \sum_{i=1}^m g_{2,i}, \dots, \sum_{i=1}^m g_{j,i})$.

3.3.2 Magnitudes de fuga

Debido a que la localización de fugas puede cambiar, dependiendo de la magnitud de la fuga que se ha generado, se decide aumentar la robustez de algoritmo al considerar todas las magnitudes de fugas posibles que se puedan generar al realizar la búsqueda de los nodos donde se localizaran los sensores.

Para esto, una vez obtenidos las medias, las dispersiones y los autovectores, en vez de aplicar el método ante un conjunto de datos de fuga de magnitud f_j para el nodo i , se lo aplica para todos los conjuntos de datos de fuga que se puedan dar en el nodo i para las diferentes magnitudes f_j con $j=1, \dots, l$, siendo l las diferentes magnitudes con las que se realiza la simulación en EPANET.

Para este caso, en vez de obtener sólo un vector con valores del discriminante de Fisher, se obtienen l vectores, para evaluar la localización se calcula la media de todos los valores del discriminante obtenidos para cada uno de los nodos y se aplica la ecuación (ecuación de error), que modificándola queda de la siguiente forma:

$$e_j = \begin{cases} 0 & \text{si } g_k = \max\left(\frac{1}{l} \sum_{k=1}^l \left(\sum_{i=1}^m g_{k,1,i}\right), \frac{1}{l} \sum_{k=1}^l \left(\sum_{i=1}^m g_{k,2,i}\right), \dots, \frac{1}{l} \sum_{k=1}^l \left(\sum_{i=1}^m g_{k,j,i}\right)\right) \\ 1 & \text{en el caso contrario} \end{cases} \quad (38)$$

3.3.3 Ruido

Para aumentar la robustez del método, considerando la precisión de los sensores de presión, se agrega un ruido blanco gaussiano (de media 0 y varianza v) a la medida esperada del sensor, siendo v la precisión que se considera que tienen los sensores.

3. METODOLOGÍA

3.3.4 Modificación del enfoque de optimización

Los algoritmos de optimización de población binaria presentan limitaciones al momento de imponer restricciones y que el número de variables con las que se trabaja en este tipo de problemas son directamente el número de nodos presentes en la red, por estas razones se realiza un cambio en el enfoque que se aplica para la optimización.

Los algoritmos genéticos de población binaria buscan un vector binario de longitud n (número de nodos) que minimice el error de localización, donde los valores 1 indican en que nodo se encuentra ubicado el sensor y los valores de 0 indican lo contrario, por lo que el número de variable del método es n .

Para disminuir la cantidad de variables, se busca un conjunto de números enteros, de cantidad igual al número de sensores que se desean instalar; estos números estarán dentro de un vector y tomarán valores enteros desde 1 hasta s , representando al nodo en el que se encuentra ubicado el sensor. Es decir que se cambia de un vector de búsqueda con n elementos a otro de menor tamaño de s elementos, de esta manera se consigue disminuir el número de variables que utiliza el algoritmo de optimización de n (número de nodos) a s (número de sensores) y de la misma manera se permite cambiar el tipo de población con la que trabaja el algoritmo, por lo que ahora el trabajar con restricciones es posible, así siguiendo el sentido común, se tienen un problema de menor cantidad de variables y por lo tanto, más fácil y rápido de resolver.

4. APLICACIÓN Y RESULTADOS

4. APLICACIÓN Y RESULTADOS

La metodología explicada en la sección anterior se aplica a dos benchmarks de redes de agua potable, que son usadas para este tipo de estudios, la primera es la red de Hanoi en Vietnam y la segunda es la red de transporte de agua de Limassol en Chipre, se realizan diferentes pruebas del método para cada uno de los tres algoritmos de optimización, el algoritmo genético de población binaria, el algoritmo genético de población entera y el algoritmo CMA-ES, en cada uno de estos casos, se realiza la optimización sin aplicar ningún tipo de robustez al algoritmo en primera instancia y luego se aplican los diferentes tipos de robustez de manera paulatina.

4.1 Aplicación a la red de Hanói

La primera red con la que se trabaja es la de Hanói, una red que cuenta con 31 nodos y donde se simulan 31 tipos de fuga, una para cada nodo con 7 diferentes magnitudes, de 20, 30, 40, 50, 60, 70 y 80 litros por segundo. Se toma un tiempo de simulación de 24 horas, con un tiempo de muestro de 15 minutos, dando como resultado un total de 97 medidas de presión para cada uno de los nodos. En cuanto a la robustez del método, al tener la red 31 nodos, aplicando la ecuación 35, da como resultado 3 nodos, por lo que al calcular la distancia límite con la ecuación 36, se tiene como resultado 3150 metros, siendo este valor el 31% de la distancia media de un nodo a su nodo más alejado.

De la misma manera se aplica la robustez de las magnitudes y posteriormente se agrega un ruido blanco gaussiano del 0,5% a la medida esperada del sensor de presión.

4.1.1 Selección de datos

El primer paso en la aplicación del método, es decidir qué tipo de datos se usarán para el mismo, teniendo como opciones los valores de presión, los

4. APLICACIÓN Y RESULTADOS

residuos y las sensibilidades, para tomar una decisión, se aplicó la metodología de localización para cada una de las 31 posibles fugas que se pueden dar para las 7 magnitudes, además considerándose que se tienen medidas en todos los nodos, todo esto para poder determinar que valores son con los que el método ofrece una menor cantidad de errores de localización.

4.1.1.1 Resultados

Aplicándose el Análisis del Discriminante de Fisher para los tres tipos de datos y probando con las diferentes magnitudes para el entrenamiento del modelo, se obtienen los siguientes resultados:

Tabla 4.1 Errores de localización para distintos tipos de datos

Magnitudes de entrenamiento	Presiones	Residuos	Sensibilidades
20	210	6	5
30	14	6	3
40	23	4	1
60	54	14	5
70	60	21	9
80	103	11	8

En la tabla 4.1 se puede observar como los errores de localización son mucho mayores cuando se aplica el método del Discriminante de Fisher a la presiones, disminuye al aplicarse a los residuos y aún menores cuando se aplica a las sensibilidades, en las figuras 4.1, 4.2 y 4.3 se puede apreciar cómo es que varían la cantidad de errores al cambiarse la magnitud de entrenamiento para cada uno de los tipos de datos usados.

4. APLICACIÓN Y RESULTADOS

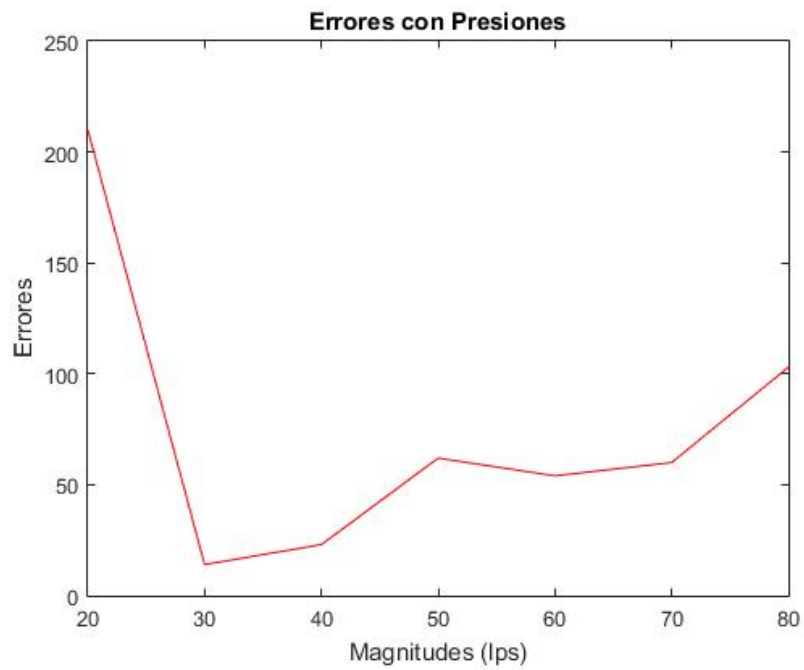


Figura 4.1 Errores de localización usando presiones

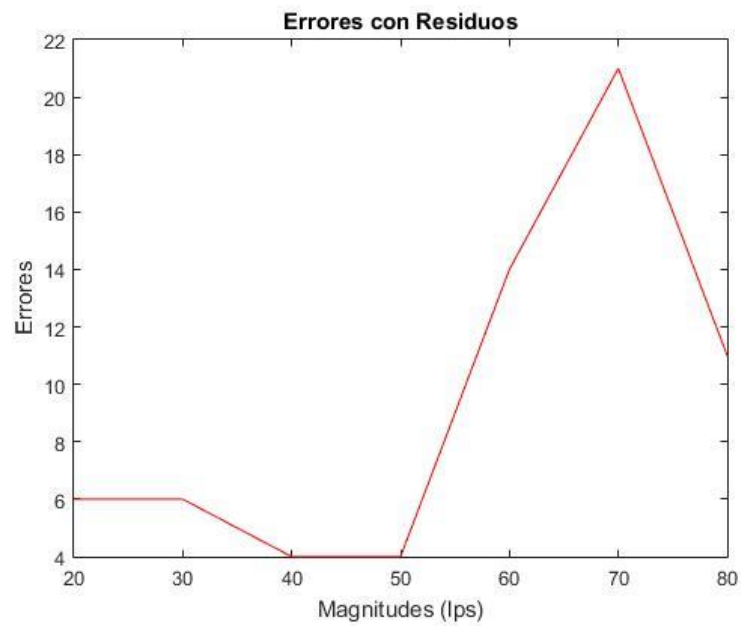


Figura 4.2 Errores de localización usando residuos

4. APLICACIÓN Y RESULTADOS

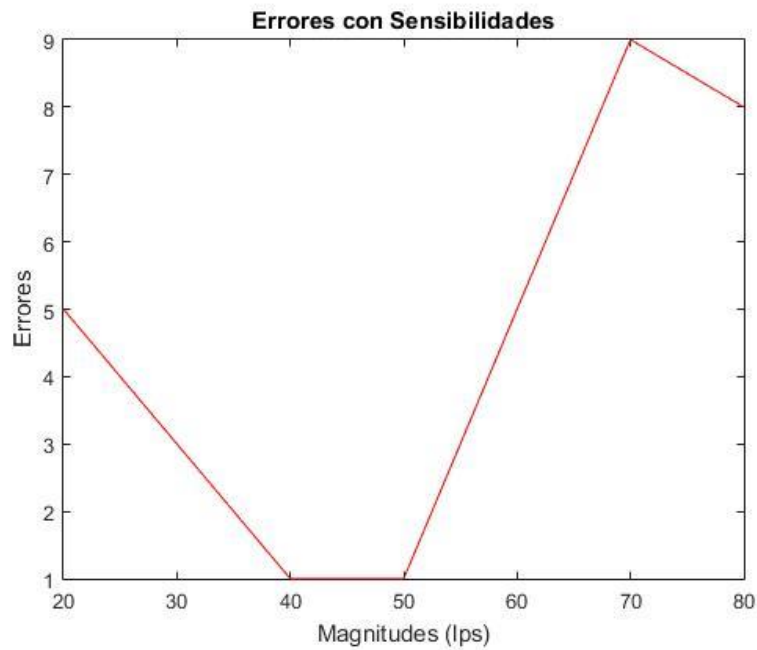


Figura 4.3 Errores de localización usando sensibilidades

Se puede observar claramente en las gráficas que el entrenar el método con sensibilidades proporciona la menor cantidad de errores al realizar la localización, por lo que se decide trabajar con este tipo de datos, ahora queda decidir que magnitud de las sensibilidades se usará, en la figura 4.3 se puede ver que al trabajar con 40 o 50 litros por segundo, se tiene solamente un error de localización.

Se decide trabajar con 50 litros por segundo porque es la magnitud media de simulación. Ahora se grafica el comportamiento del método al entrenar con esta magnitud, para ver dónde es que ocurre el error de localización.

4. APLICACIÓN Y RESULTADOS

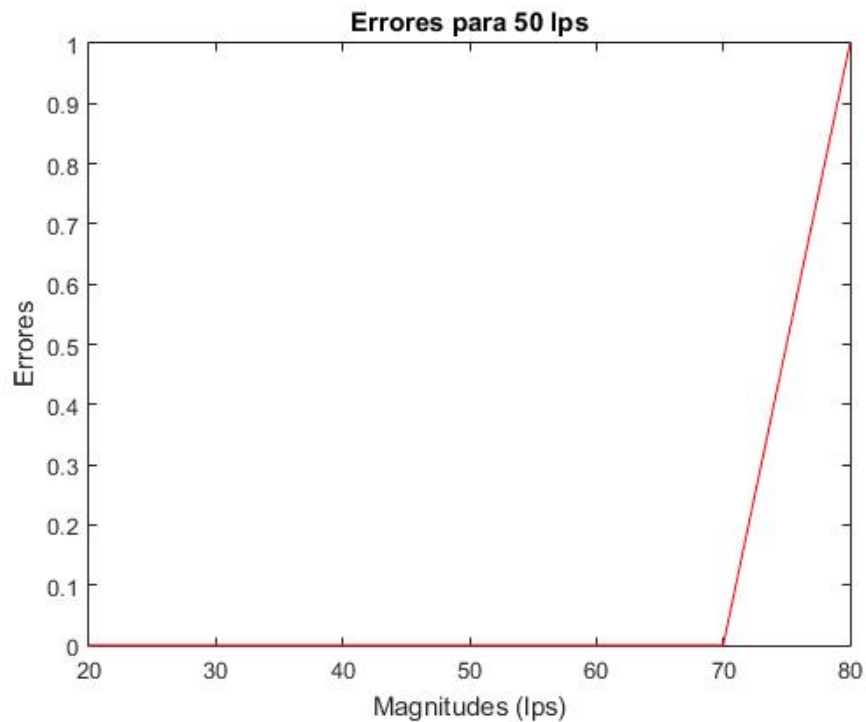


Figura 4.4 Errores de localización con Sensibilidad de 50 lps

En la figura 4.4 se puede ver que el método entrenado con 50 litros por segundo, realiza una localización perfecta para las magnitudes desde 20 a 70 litros por segundo, sin embargo se da un error al realizar la localización de una fuga de 80 litros por segundo.

Por lo tanto para la red de Hanói, se decide realizar el entrenamiento del método con una magnitud de 50 litros por segundo, siendo esta la que presentó menor cantidad de errores. En la figura 4.5 se puede ver cómo quedan caracterizadas algunas de las diferentes 31 fugas que pueden existir, luego de aplicarse la transformación con los autovectores.

4. APLICACIÓN Y RESULTADOS

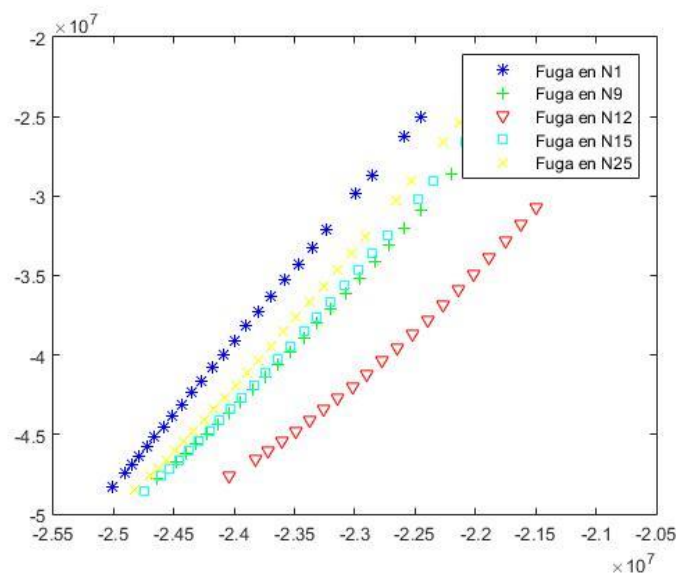


Figura 4.5 Caracterización de diferentes tipos de fugas

Se realiza una prueba para ver cómo es que trabaja el Discriminante de Fisher para localizar fugas, para esto entrenamos el método con 50 litros por segundo de magnitud y generamos una fuga en el nodo 15 de 70 litros por segundo de magnitud.

4. APLICACIÓN Y RESULTADOS

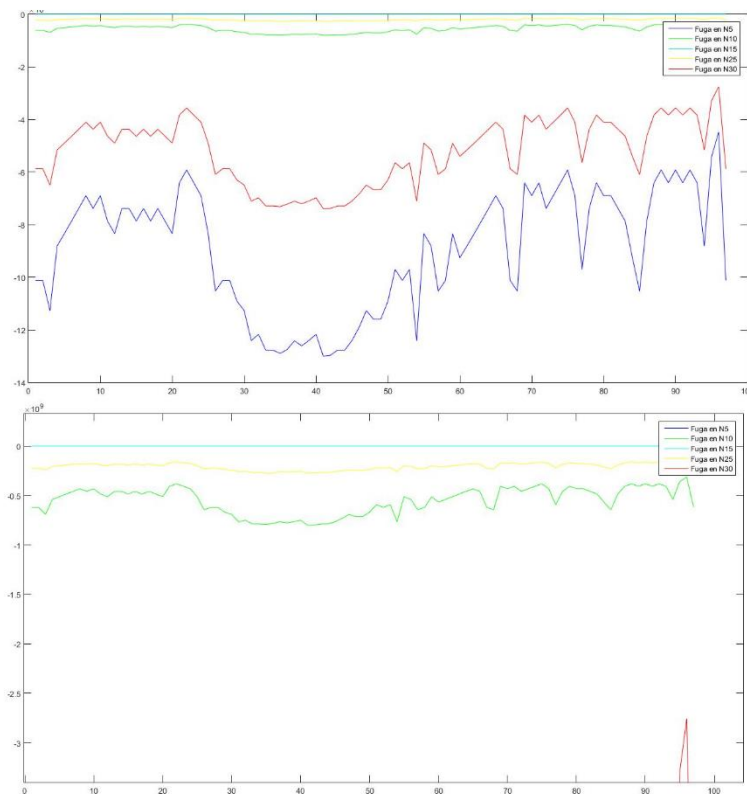


Figura 4.6 Valores del Discriminante de Fisher para fuga en el nodo 15

Aplicando el discriminante de Fisher, se obtiene la gráfica 4.6, donde se puede ver que los valores más altos del discriminante corresponden a la fuga en el nodo 15, coincidiendo esto con el nodo donde se generó la fuga, por lo que se la localiza correctamente.

4.1.2 Optimización con GA binario

Se aplica el algoritmo de optimización en la red de Hanói con el algoritmo genético de población binaria, usando 5 iteraciones del método, con una población inicial de 30 vectores y 10 generaciones para cada iteración, se hace esto para 1, 2 y 3 sensores y para 2 situaciones:

- La primera se aplica el algoritmo sin ningún tipo de robustez.
- La segunda situación con todas las modificaciones de robustez, es decir agregando las distancias, magnitudes y ruido.

4. APLICACIÓN Y RESULTADOS

4.1.2.1 Resultados sin robustez

Se empieza probando el algoritmo genético para un sólo sensor, obteniéndose como resultado los siguientes nodos donde se debe agregar un sensor:

- [12] con un error del 0% y un tiempo de cómputo de 600 minutos.
- [25] con un error del 0% y un tiempo de cómputo de 383 minutos.

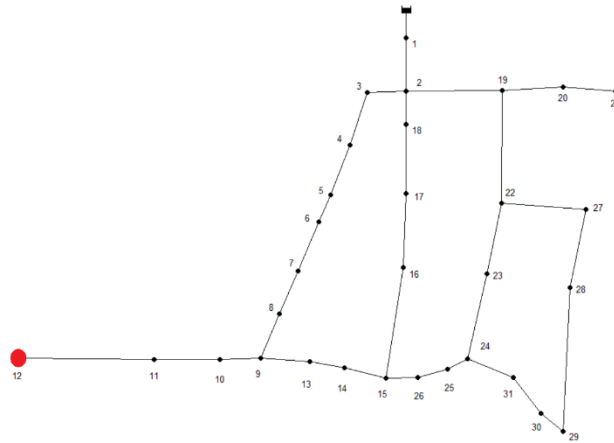


Figura 4.7 Localización del sensor en el nodo 12

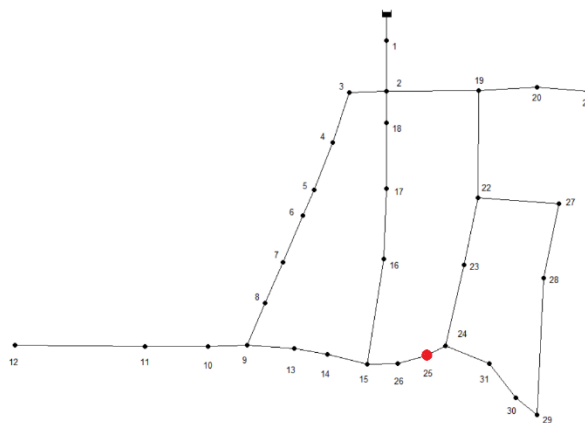


Figura 4.8 Localización del sensor en el nodo 25

Al usar el método con 2 sensores, se obtienen las siguientes configuraciones que también son óptimas:

4. APLICACIÓN Y RESULTADOS

- [3 11] con un error del 0% y un tiempo de cómputo de 155 minutos.
- [1 6] con un error del 0% y un tiempo de cómputo de 440 minutos.

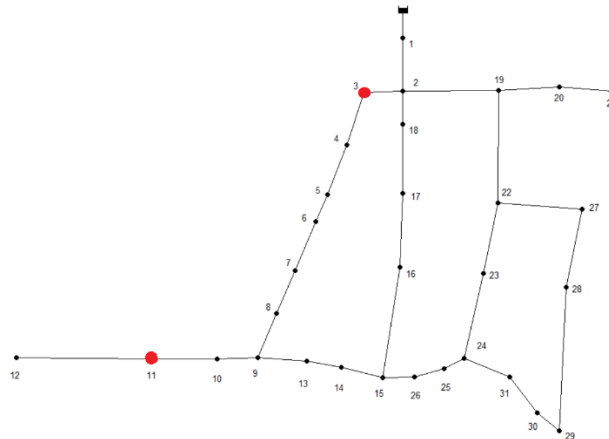


Figura 4.9 Localización de los sensores en los nodos 3 y 11

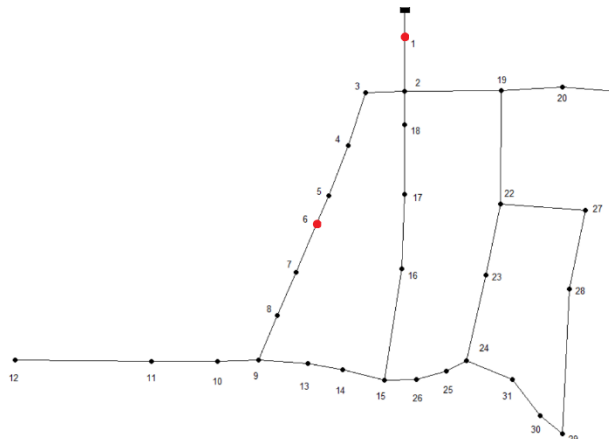


Figura 4.10 Localización de los sensores en los nodos 1 y 6

En el caso de dos sensores, el método encuentra muchas otras configuraciones más, que también son óptimas.

Finalmente al trabajar con 3 sensores, el algoritmo genético, que se encuentra implementado en Matlab, no respeta el número de sensores asignados, ya que estos se encuentran especificados por una restricción, sin embargo al tratarse de una población binaria, el algoritmo considera la restricción, pero no

4. APLICACIÓN Y RESULTADOS

fuerza a que se cumpla, por lo que el resultado de la optimización genera en algunos casos una configuración con dos nodos [11 18] de error 0% y con un tiempo de cómputo de 330 minutos y en otros casos una configuración de 3 nodos [12 18 30] con un índice de error de 0.0842 y un tiempo de cálculo de 300 minutos.

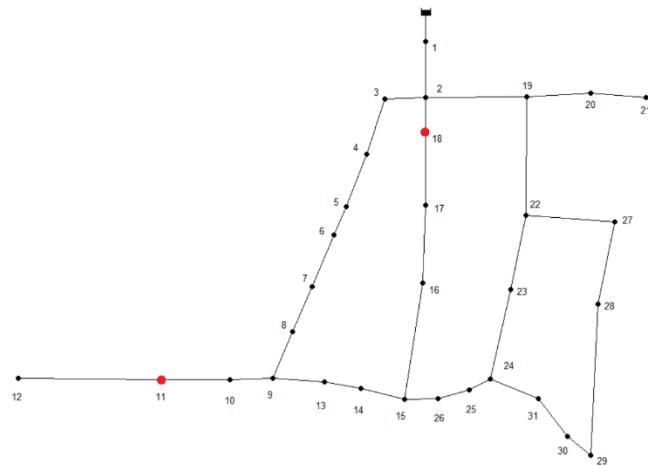


Figura 4.11 Localización de los sensores en los nodos 11 y 18

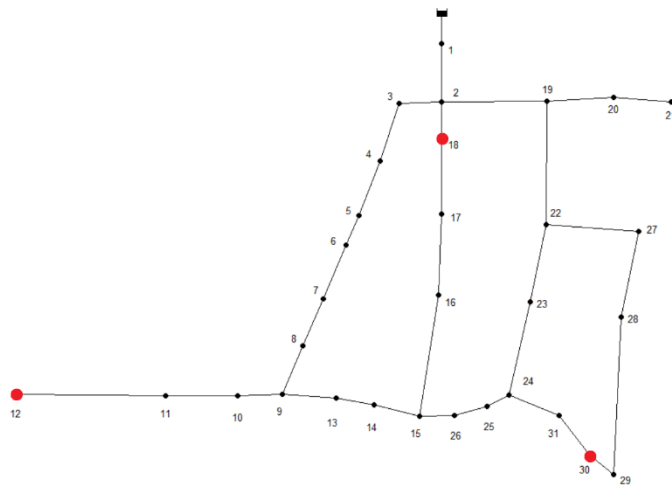


Figura 4.12 Localización de los sensores en los nodos 12 y 18

4. APLICACIÓN Y RESULTADOS

4.1.2.2 Resultados con robustez

El siguiente paso es aplicar los métodos de robustez al algoritmo, por lo que empezando a probar con un sensor, el algoritmo no respeta la restricción del número de sensores y encuentra una configuración de 7 sensores en los nodos [9 10 11 15 18 21 29] con un índice de error de 0.0588, como se puede ver en la figura 4.13.

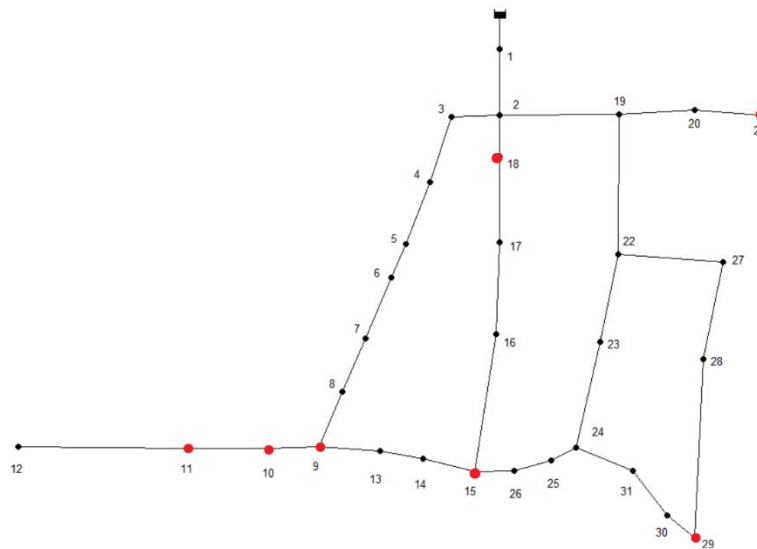


Figura 4.13 Configuración obtenida con 7 sensores

Al realizar el experimento, ahora con 2 sensores se obtiene la configuración [12 29] con un índice de error de 0.2459 y un tiempo de cómputo de 232 minutos.

4. APLICACIÓN Y RESULTADOS

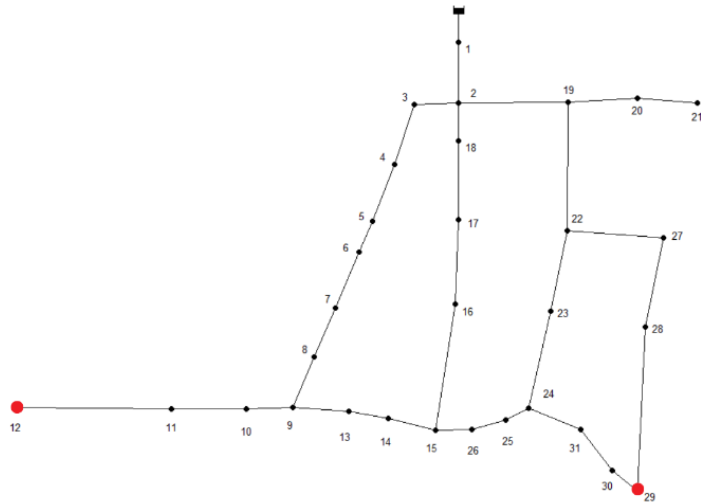


Figura 4.14 Localización de los sensores en los nodos 12 y 29

Finalmente se aplica el algoritmo genético para tres sensores, pero al igual que en el caso sin robustez, el algoritmo no respeta el número de sensores y encuentra una configuración de 11 sensores en total, [6 8 9 11 12 14 18 19 21 28 29], con un índice de error de 0,0126; el algoritmo de optimización encuentra un valor muy cercano al óptimo, sin embargo para conseguir esto va agregando más sensores al sistema.

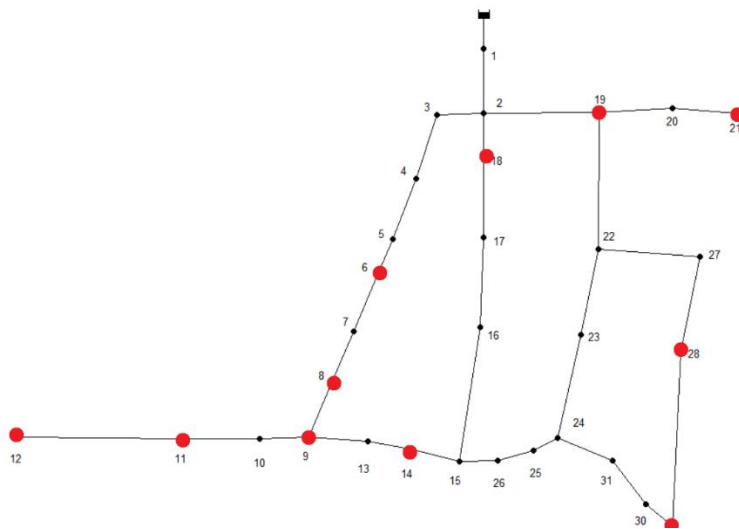


Figura 4.15 Configuración obtenida con 11 sensores

4. APLICACIÓN Y RESULTADOS

4.1.3 Optimización con GA entero

Debido al problema de que el algoritmo genético no cumple con las restricciones al tratarse de una población binaria, se modifica el método para que la población sea entera, como se explicó en la sección 3.3.4. Al igual que en el punto anterior, se realizan las mismas pruebas para 1, 2 y 3 sensores.

4.1.3.1 Resultados sin robustez

En primera instancia se prueba el algoritmo para un sensor, lo que da como resultado la configuración [12] con un índice de error de 0 y un tiempo de cómputo de 44 minutos. Este resultado coincide con el algoritmo genético binario.

A continuación se prueba el algoritmo para 2 sensores, lo que da como resultado la configuración [4 30] con un índice de error de 0 y un tiempo de cómputo de 58 minutos.

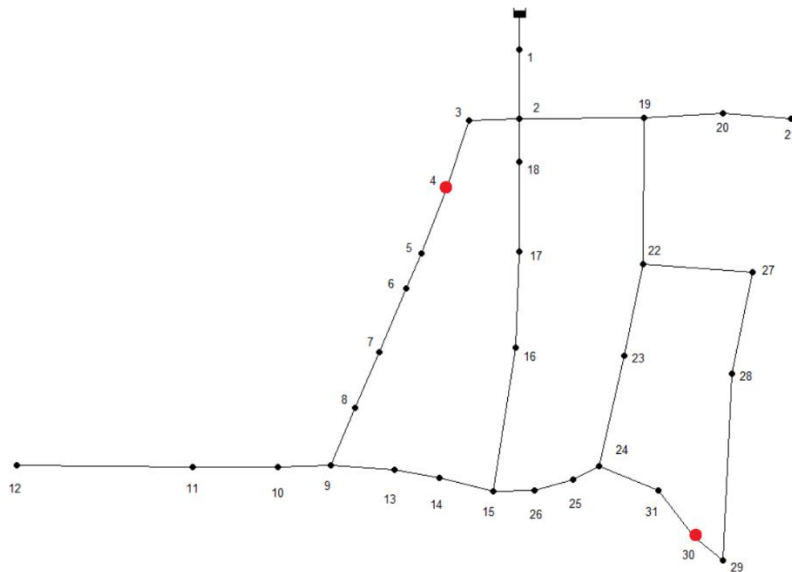


Figura 4.16 Localización de los sensores en los nodos 4 y 30

Por último se prueba el método para 3 sensores, el algoritmo encuentra la configuración [3 12 29] con un índice de error de 0.0359 y un tiempo de cómputo de 135 minutos.

4. APLICACIÓN Y RESULTADOS

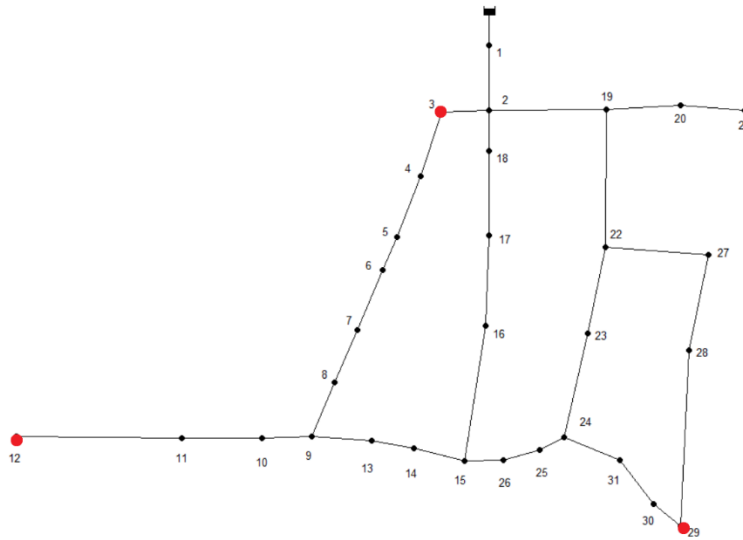


Figura 4.17 Localización de los sensores en los nodos 3, 12 y 29

4.1.3.2 Resultados con robustez

Aplicando los métodos de robustez para el algoritmo de optimización con 1 sensor, se obtiene como mejor solución usar un sensor en el nodo [15] con un índice de error de 0.4695 y un tiempo de cómputo de 62 minutos.

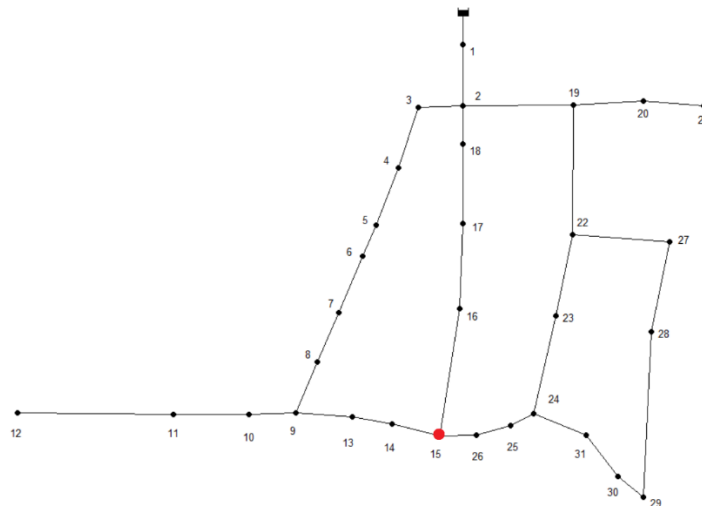


Figura 4.18 Localización del sensor en el nodo 15

Luego al aplicar el algoritmo genético entero para dos sensores, se obtiene la configuración [12 29] con un índice de error de 0.2459 y un tiempo de cómputo

4. APLICACIÓN Y RESULTADOS

de 310 minutos, esta configuración coincide con la obtenida por el algoritmo genético binario.

Finalmente se prueba el método con 3 sensores, donde después de 350 minutos de tiempo de cómputo se obtiene la configuración de [11 15 28], que da un índice de error de 0.0971.

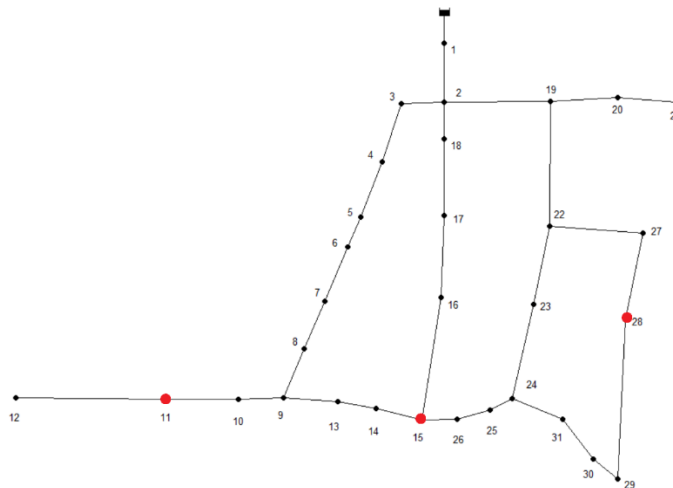


Figura 4.19 Localización de los sensores en los nodos 11, 15 y 28

4.1.4 Optimización con CMA-ES

El último algoritmo de optimización aplicado es el de la Adaptación de la Matriz de Covarianza, para aplicar este método se definen un máximo de 100 iteraciones, un tamaño de población de 50 y además se define un valor de parada para la función objetivo de 0, por lo que cuando el algoritmo encuentre una función objetivo de valor óptimo, se detiene.

4.1.4.1 Resultados sin robustez

Al aplicarlo para un solo sensor, se obtiene que se debe agregar un sensor en el nodo [22], un índice de error de 0 y un tiempo de cómputo de 16 segundos.

4. APLICACIÓN Y RESULTADOS

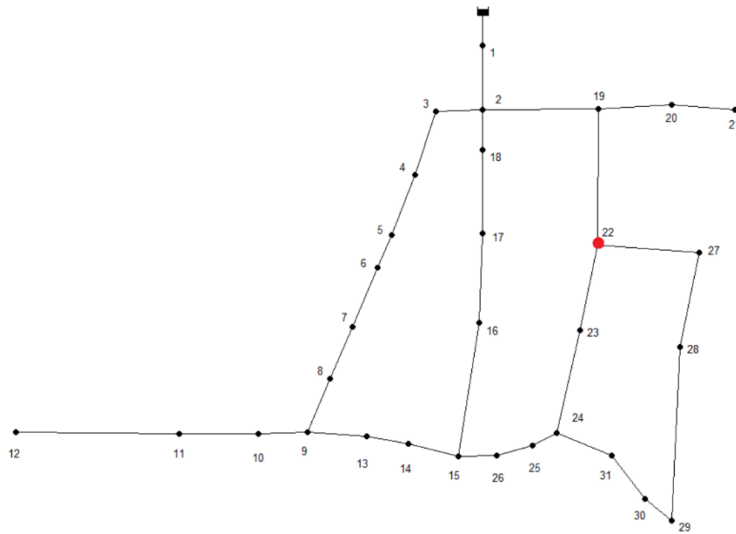


Figura 4.20 Localización del sensor en el nodo 22

Al aplicar el método para dos sensores, se obtiene la configuración [18 31], que también es óptima y en un tiempo de 12 minutos.

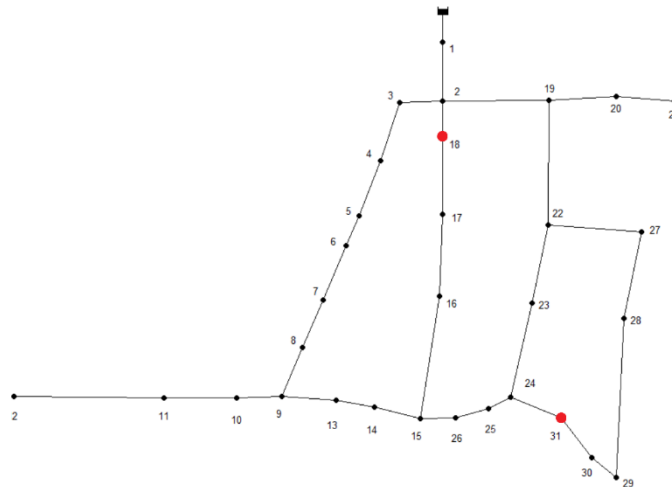


Figura 4.21 Localización de los sensores para los nodos 18 y 31

Por último se aplica el método para 3 sensores, obteniéndose como resultado la configuración [5 11 12] con un índice de error de 0.0543 con un tiempo de cómputo de 12 minutos.

4. APLICACIÓN Y RESULTADOS

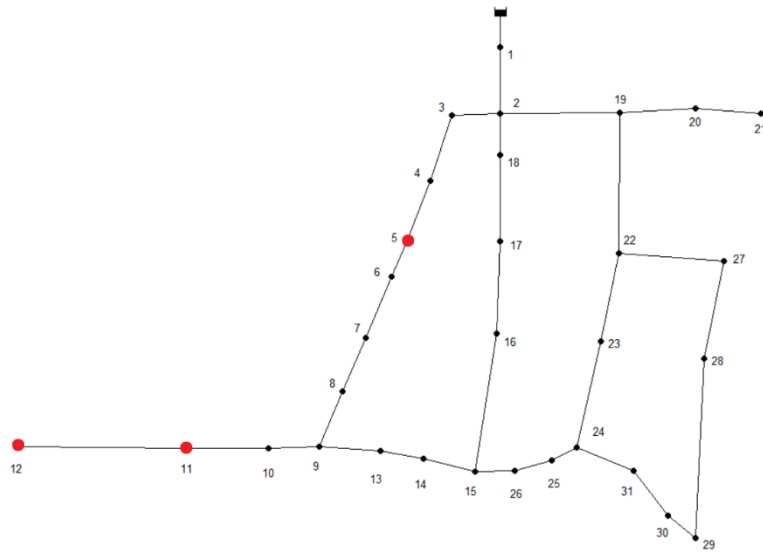


Figura 4.22 Localización de los sensores en los nodos 5, 11 y 12

4.1.4.2 Resultados con robustez

Siguiendo la metodología ahora se realiza la optimización con el algoritmo CMA-ES aplicando la robustez, obteniendo como configuración para un sensor el nodo [29] con un índice de error de 0.4982 y un tiempo de cómputo de 109 minutos.

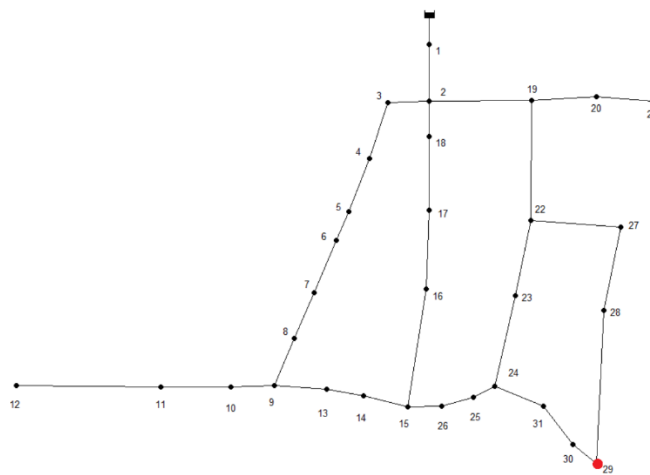


Figura 4.23 Localización del sensor en el nodo 29

4. APLICACIÓN Y RESULTADOS

Después se aplica el método para 2 sensores, donde se obtiene la configuración [12 29] con un índice de error de 0.2459 y un tiempo de cómputo de 54 minutos.

Finalmente se realiza la optimización para 3 sensores, obteniéndose la configuración [11 15 28] que al igual que con el algoritmo genético entero, se obtiene un índice de error de 0.0903 y un tiempo de cómputo de 965 minutos.

4.1.5 Comparación de los métodos

Una vez aplicados los 3 métodos de optimización para 1, 2 y 3 sensores, en la tabla 4.2 se tienen resumidos los resultados para el algoritmo sin robustez.

Tabla 4.2 Configuraciones obtenidas sin aplicar robustez

Sin Robustez						
Sensores	GA Binario		GA Entero		CMA-ES	
	Configuración	Índice error	Configuración	Índice error	Configuración	Índice error
1	[12]	0	[12]	0	[22]	0
2	[3 11]	0	[4 30]	0	[18 31]	0
3	[12 18 30]	0,0842	[3 12 29]	0,0359	[5 11 12]	0,0543

Se puede ver que para 1 y 2 sensores existen múltiples configuraciones óptimas y que dependiendo del algoritmo de optimización, la configuración encontrada cambia y se mantiene óptima. Sin embargo se puede ver que al aplicar el método con 3 sensores, se puede ver que el método disminuye su capacidad de localización, aumentando el índice de error.

Por otro lado, en la tabla 4.3 se tienen los resultados obtenidos de la optimización con la robustez aplicada.

4. APLICACIÓN Y RESULTADOS

Tabla 4.3 Configuraciones obtenidas aplicando robustez

Con Robustez						
Sensores	GA Binario		GA Entero		CMA-ES	
	Configuración	Índice error	Configuración	Índice error	Configuración	Índice error
1	7 Sensores	0,0588	[15]	0,4695	[29]	0,4982
2	[12 29]	0,2459	[12 29]	0,2459	[12 29]	0,2459
3	11 Sensores	0,01265	[11 15 28]	0,0903	[11 15 28]	0,0903

En la tabla anterior se puede apreciar que para 1 sensor, el algoritmo genético binario no cumple con las restricciones del número de sensores, pero los otros algoritmos si consiguen encontrar diferentes configuraciones con índices de error bastante cercanos. Al comparar los resultados de 2 sensores, se puede ver que los 3 algoritmos encuentran la misma solución, finalmente al aumentar el número de sensores a 3, nuevamente el algoritmo genético binario no respeta la restricción del número de sensores; los demás algoritmos encuentran la misma configuración con un índice de error de 0.0903.

4.2 Aplicación a la red de Limassol

La segunda red de agua potable con la que se trabaja es la de Limassol en Chipre, esta red cuenta con 197 nodos y se simulan 197 tipos de fuga, una para cada nodo y de 5 magnitudes diferentes; de 1, 1.33, 1.66, 2 y 2.33 litros por segundo. El tiempo de simulación es de 24 horas, tomándose medidas durante cada hora, dando como resultado un total de 25 medidas para cada nodo.

Para obtener la distancia límite, primero usamos la ecuación 35 para obtener como resultado 7 nodos, luego calculamos la distancia límite con la ecuación 36 obteniéndose como resultado 1443 metros, siendo este valor el 25% de las distancia media de un nodo, a su nodo más alejado.

Como en la red de Hanói, se aplica la robustez de las magnitudes y se agrega un ruido blanco Gaussiano del 0,5% a la medida esperada por el sensor.

4. APLICACIÓN Y RESULTADOS

4.2.1 Prueba con diferentes magnitudes

Para la red de Limassol, el primer paso es realizar pruebas de optimización modificando las magnitudes con las que se entrena el Análisis del discriminante de Fisher y las magnitudes de las fugas que se detectan, en la tabla 4.4 se puede ver la configuración para 3 sensores que se obtiene y en la tabla 4.5 sus índices de error.

Tabla 4.4 Configuraciones para diferentes combinaciones de magnitudes

		Magnitud de entrenamiento				
		1	1,33	1,66	2	2,33
Magnitud de prueba	1		[1 29 156]	[7 118 135]	[40 76 117]	[5 34 197]
	1,33	[29 94 179]		[6 45 197]	[45 91 146]	[9 47 197]
	1,66	[1 72 155]	[1 49 197]		[7 38 197]	[1 36 197]
	2	[2 80 197]	[15 38 197]	[1 107 197]		[1 75 149]
	2,33	[35 80 197]	[1 71 197]	[2 80 134]	[2 82 197]	

Tabla 4.5 Índice de error para las configuraciones de la tabla 4.4

		Magnitud de entrenamiento				
		1	1,33	1,66	2	2,33
Magnitud de prueba	1		0,3249	0,3858	0,3401	0,3706
	1,33	0,2843		0,3452	0,3959	0,3858
	1,66	0,2858	0,4213		0,401	0,4112
	2	0,4264	0,401	0,4569		0,4721
	2,33	0,4569	0,4721	0,4467	0,4721	

Con las configuraciones obtenidas se realiza un experimento en el que para cada configuración, se calcula el índice de error que tendría la configuración si se agrega la robustez de las distancias y las magnitudes; de la misma manera se calcula el índice de error de la configuración al agregarse un ruido blanco gaussiano del 0.5% a la medida esperada del sensor.

4. APLICACIÓN Y RESULTADOS

Tabla 4.6 Índices de error al agregar robustez y ruido

Configuración	Errores		
	Sin Robustez	Distancia y Magnitudes	Ruido
[29 94 179]	0,2843	0,0888	0,9275
[1 72 155]	0,2858	0,1186	0,9246
[2 80 197]	0,4264	0,1058	0,9064
[35 80 197]	0,4569	0,086	0,9169
[1 29 156]	0,3249	0,1173	0,9468
[1 49 197]	0,4213	0,12	0,9052
[15 38 197]	0,401	0,0993	0,9408
[1 71 197]	0,4721	0,1025	0,9337
[7 118 135]	0,3858	0,1447	0,9179
[6 45 197]	0,3452	0,1066	0,9318
[1 107 197]	0,4569	0,1201	0,9174
[2 80 134]	0,4467	0,127	0,9243
[40 76 117]	0,3401	0,106	0,8951
[45 91 146]	0,3959	0,0942	0,9101
[7 38 197]	0,401	0,1107	0,9095
[2 82 197]	0,4721	0,0925	0,9103
[5 34 197]	0,3706	0,1059	0,9299
[9 47 197]	0,3858	0,1159	0,9095
[1 36 197]	0,4112	0,1196	0,9189
[1 75 149]	0,4721	0,0915	0,892

En la tabla 4.6 se pueden ver los resultados de los experimentos, se puede apreciar que al calcular el índice de error con las distancias y las magnitudes, la eficacia de la localización aumenta considerablemente, hasta valores del orden de 0.1, lo que significa que el método es capaz de localizar las fugas con una precisión promedio de 144 metros.

Sin embargo, al agregar también el ruido, el índice de error aumenta demasiado, elevándose al orden de 0.9, esto da a pensar que pueden existir otras configuraciones que generen un índice de error más bajo a los encontrados en este experimento, por lo que se decide aplicar los algoritmos de optimización a la red con las medidas de robustez y ruido, para ver si existe una configuración que al aplicarse ruido, genere un índice de error menor a los obtenidos en la tabla.

4. APLICACIÓN Y RESULTADOS

4.2.2 Optimización con GA binario

Para la aplicación del algoritmo genético de población binaria en la red de Limassol, se realizan 5 iteraciones del método, cada uno con una población inicial de 30 vectores y 5 generaciones para cada iteración, para este caso se realizan pruebas con tres sensores como restricción. Se prueba el algoritmo con todas las medidas de robustez y un ruido del 0,5% sobre la medida esperada, el resultado obtenido es una configuración de 6 sensores [3 76 87 108 163 190] con un índice de error de 0.8458 y un tiempo de cómputo de 637 minutos.

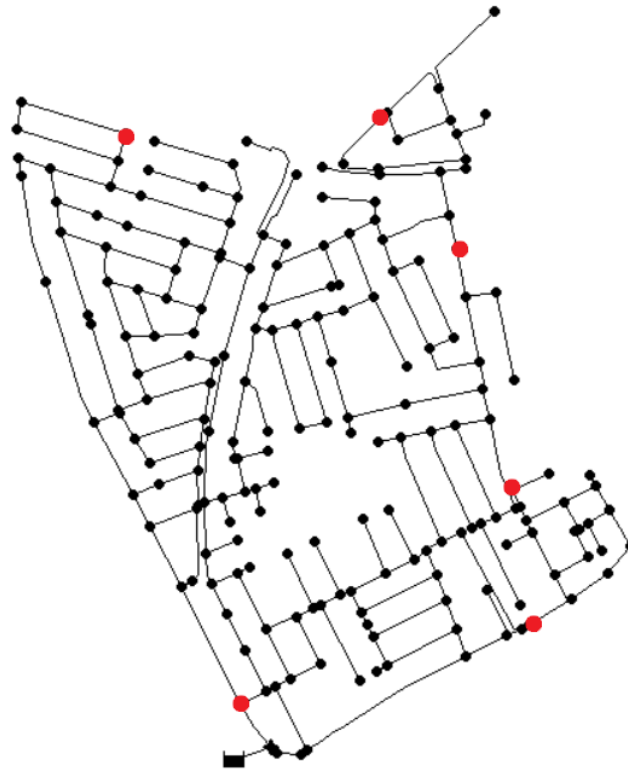


Figura 4.24 Localización de sensores en los nodos 3, 76, 87, 108, 163 y 190

4.2.3 Optimización con GA entero

El siguiente algoritmo que se pone a prueba para la optimización es el algoritmo genético, pero en este caso con población entera, consiguiendo de esta manera que se cumplan las restricciones del número de sensores, al igual

4. APLICACIÓN Y RESULTADOS

que con el algoritmo genético binario, se trabaja con 3 sensores, se realizan 5 iteraciones del método, cada uno con una población inicial de 30 vectores y 5 generaciones para cada iteración. El resultado obtenido es la configuración [9 158 184] con un índice de error de 0.8762 y un tiempo de cómputo de 4380 minutos.

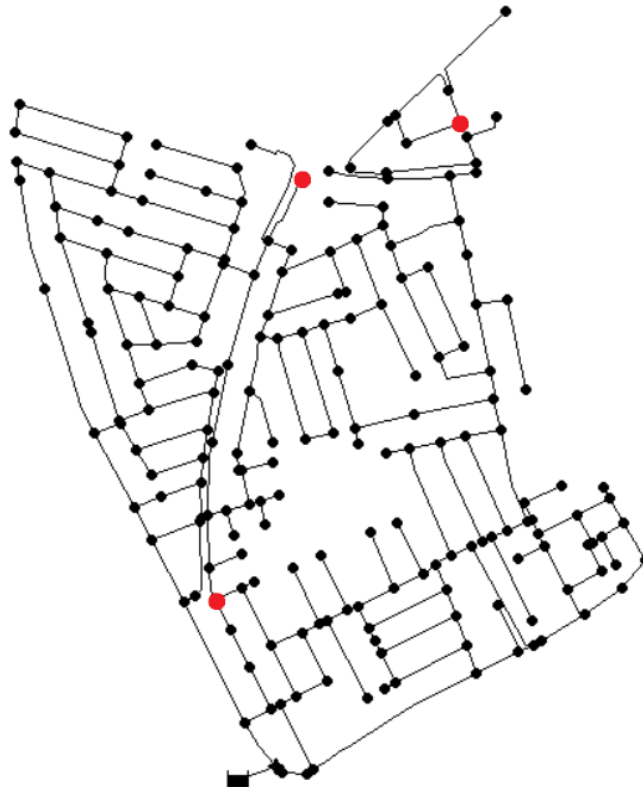


Figura 4.25 Localización de sensores en los nodos 9, 158 y 184

4.2.4 Optimización con CMA-ES

Finalmente se aplica el algoritmo CMA-ES, al igual que con la red de Hanói, se definen un máximo de 100 iteraciones, un tamaño de población de 50 y un valor de parada de la función objetivo de 0,07. La configuración obtenida es la [42 50 197] con un índice de error de 0.8838 y un tiempo de cómputo de 1655 minutos.

4. APLICACIÓN Y RESULTADOS

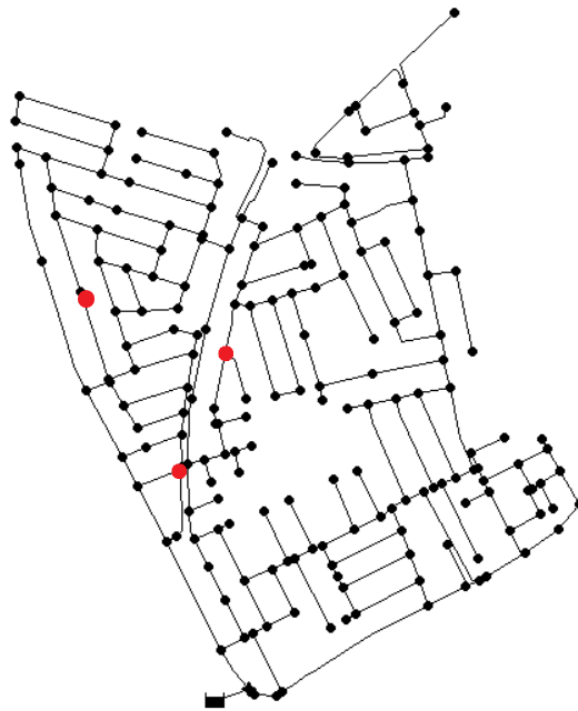


Figura 4.26 Localización de sensores para los nodos 42, 50 y 197

4.3 Comparación con bibliografía existente

Una vez probadas las dos redes con el método propuesto, se realiza la comparación de los resultados con los del artículo *“Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Genetic Algorithms”*, donde se realiza la localización óptima de los sensores a partir de algoritmos genéticos binarios y como método de localización de fugas el cálculo de la proyección de los vectores de residuos con los vectores de sensibilidad.

4.3.1 Hanói

Para la red de Hanói, en el artículo se realizan pruebas de optimización con el algoritmo genético binario, aplicándose la combinación de todas las magnitudes y la asignación de errores en base a la distancia. Los resultados obtenidos son la configuración [12 21] para dos sensores con un índice de error de 0.061 y para 3 sensores la configuración [12 14 21] con un índice de error de 0.011.

4. APLICACIÓN Y RESULTADOS

Tabla 4.7 Comparación de resultados

Sensores	Proyección ángulo		FDA	
	Configuración	Índice error	Configuración	Índice error
1	-	-	[12]	0
			[22]	0
2	[12 21]	0,061	[3 11]	0
			[4 30]	0
			[18 31]	0
3	[12 14 21]	0,011	[3 12 19]	0,0359

Comparando estos resultados con los obtenidos en este trabajo, para dos sensores se tiene un mejor rendimiento del Análisis del discriminante de Fisher, que encuentra varias configuraciones que dan un resultado óptimo, más aun, al aplicar un solo sensor, también se encuentran resultados óptimos.

Al comprar el rendimiento con 3 sensores, el discriminante de Fisher disminuye su rendimiento en comparación al método de la proyección del ángulo, ya que este método encuentra la configuración [3 12 19] con índice de error de 0.0359, que es un poco mayor al 0.011 encontrado por el método del artículo.

Por otra parte, con la configuración obtenida en el artículo [12 14 21], se prueba el método del Discriminante de Fisher, obteniéndose un índice de error de 0.1541 cuando se trabaja sin ruido, valor que es más alto que cuando se aplica el método de la proyección del ángulo; también se prueba esta configuración con los valores de presión con ruido, obteniéndose un índice de 0.3177, que es mucho mayor al índice de error de la configuración obtenida en este trabajo.

4.3.2 Limassol

Comparando los resultados del primer experimento en la red de Limassol, donde se hace variar las magnitudes de entrenamiento con las de la fuga, análogamente al artículo, donde se hacen variar las magnitudes con las que se calculan las sensibilidades y las magnitudes con las que se calculan los residuos. En la tabla 4.8 se pueden ver los resultados obtenidos en el artículo,

4. APLICACIÓN Y RESULTADOS

se puede apreciar que los índices de error tienen valores algo menores a los valores conseguidos en este trabajo (ver tabla 4.5) además de configuraciones diferentes.

Tabla 4.8 Configuraciones e índices de error obtenidos en el artículo

CONFIGURACIONES						
		Magnitud de residuos				
		1	1,33	1,66	2	2,33
Magnitud de sensibilidades	1		[40 77 172]	[25 77 133]	[76 133 185]	[76 133 152]
	1,33	[76 133 152]		[76 86 152]	[77 124 152]	[76 110 173]
	1,66	[85 156 196]	[8 76 150]		[75 116 157]	[72 115 150]
	2	[72 118 163]	[76 133 141]	[77 111 150]		[23 75 152]
	2,33	[76 128 140]	[75 120 150]	[77 115 137]	[29 112 152]	
ÍNDICES DE ERROR						
		Magnitud de residuos				
		1	1,33	1,66	2	2,33
Magnitud de sensibilidades	1		0,324	0,294	0,299	0,314
	1,33	0,299		0,284	0,279	0,294
	1,66	0,279	0,274		0,243	0,243
	2	0,309	0,279	0,263		0,258
	2,33	0,324	0,279	0,263	0,258	

Finalmente se comparan los resultados de la optimización obtenida por el algoritmo genético del artículo, con los 3 algoritmos de optimización usados en este trabajo, obteniéndose como resultado un mejor rendimiento del método de la proyección ante la presencia de ruido, como se ve en la tabla 4.9.

Tabla 4.9 Resultados comparados de optimización

Algoritmo	Proyección ángulo		FDA	
	Configuración	Índice error	Configuración	Índice error
GA binario	[2 75 158]	0,302	6 sensores	0,8458
GA entero	-	-	[9 58 184]	0,8762
CMA-ES	-	-	[42 50 197]	0,8838

4. APLICACIÓN Y RESULTADOS

Se puede ver en la tabla que el método de la proyección del ángulo obtiene un índice de error de 0.302, comparándolo con los resultados obtenidos en este trabajo, que son del orden de 0.87 para 3 sensores, de la misma manera se obtiene un resultado de 6 sensores con el algoritmo genético binario, que tiene un índice de error de 0.8458.

Se puede apreciar claramente que al agregar ruido, los resultados obtenidos con el presente trabajo, tienden a bajar su eficacia muy drásticamente si lo comparamos con los resultados obtenidos sin ruido, donde se obtienen configuraciones de error muy bajo, del orden de 0,1.

5. CONCLUSIONES

5. CONCLUSIONES

Tras analizar los resultados obtenidos en este trabajo, se puede afirmar que el método del Análisis del Discriminante de Fisher ofrece una muy buena capacidad de localización de fugas, cuando es aplicado sin ruido a un conjunto de valores de sensibilidad calculados a partir de las presiones obtenidas de la simulación.

Para la red de Hanói se pudo observar un comportamiento excelente del método ya que al aplicarlo sin ruido, pudo encontrar varias configuraciones de uno o dos sensores que ofrecen una localización perfecta de fugas, sin embargo al aumentar el número de sensores a tres, la efectividad disminuyó un poco, teniendo un índice de error de 0.0359. Al comparar estos resultados con la bibliografía existente, el método presentado en este trabajo tiene una capacidad de localización mucho mejor que con el método comparado, ya que el método de la proyección del ángulo ofrece configuraciones de 2 sensores con un error de 0.061 y el discriminante de Fisher encuentra varias configuraciones de 2 sensores con índice de error de 0. En cuanto a configuraciones de 3 sensores, si bien el rendimiento de este método disminuye, continua siendo una posibilidad bastante atractiva por el índice de error bajo.

Cuando se aplica ruido a las mediciones de la red de Hanói, el rendimiento de la red disminuye, aunque no existen resultados para comparar en el artículo usado, se consiguen configuraciones 2 sensores con índice de error de 0.2456 y de 3 sensores de 0.0903, que son valores de índice de error bastante buenos, considerando que con 3 sensores se puede localizar perfectamente el 91% de las fugas.

Al trabajar con la red de Limassol, se obtienen resultados del orden 0.3 a 0.4 de índice de error, al aplicarse el algoritmo sin robustez y variándose las magnitudes de prueba, pero al aplicar la robustez de la distancia estos valores disminuyen hasta el orden de 0,1, lo que indica un buen funcionamiento del

5. CONCLUSIONES

método al conseguir localizarse las fugas en un área de 144 metros en una red donde la distancia media de un nodo, a su nodo más alejado es de 5760 metros.

Al aplicar ruido a las medidas y aplicar el método, el rendimiento del método baja mucho, ya que los índices de error de las configuraciones encontradas son en promedio de 0.9, lo que significa que el 90% de las fugas no son localizadas, al comparar este resultado con el obtenido en el artículo, se ve que el método de la proyección del ángulo presenta un rendimiento mucho mejor, ya que obtiene índice de error de 0.302.

En cuanto al método del discriminante de Fisher, tiene un muy buen rendimiento al aplicarlo al problema de localización de fugas, el problema se da cuando se agrega ruido a las mediciones, el descenso drástico de su efectividad hace pensar que se debe realizar un pretratamiento de datos para aplicar el método.

El comparar los tres algoritmos de optimización, se puede ver que la mejoras más importantes al usar el algoritmo genético entero y CMA-ES, es que con estos métodos se pueden imponer restricciones, lo que permite realizar la búsqueda de las configuraciones solamente con el número de sensores que se desea, de la misma manera la disminución del número de variables de búsqueda, que disminuye de ser el número de sensores con los que cuenta la red, al número de sensores de búsqueda. También se pudo observar que no cuanto al tiempo de cómputo, el algoritmo CMA-ES es el que encuentra las soluciones más rápidamente que los algoritmos genéticos; mientras que algoritmo genético de población entera demostró ser el segundo más rápido.

5.1 Trabajo futuro

Las líneas de trabajo futuro que se abren a partir de la realización de este proyecto son:

- Desarrollar o aplicar un método de pretratamiento de datos para la aplicación del Análisis del Discriminante de Fisher, para poder disminuir

5. CONCLUSIONES

los efectos ocasionados por el ruido en el método, ya que sin existir el mismo, el método demostró un muy buen rendimiento en la localización de fugas.

- Realizar más pruebas para determinar los parámetros más adecuados con los que se debe trabajar en los algoritmos de optimización, entre ellos el tamaño de la población, el número de generaciones y la cantidad de iteraciones que debe realizar el método, considerando que mientras mayores sean estos valores, mayor será el tiempo de cómputo pero los resultados pueden ser mejores.
- Trabajar en el algoritmo de cálculo del discriminante de Fisher, para disminuir el tiempo de cómputo del mismo debido al cálculo de matrices inversas y determinantes, ya que al aumentar el tamaño de las matrices con las que se trabaja, el tiempo de cómputo de estas operaciones aumenta considerablemente.

6. BIBLIOGRAFÍA

- Baquela, E., & Redchuk, A. (2013). *Optimización Matemática con R. Volumen I: Introducción al modelado y resolución de problemas*. Ed. Bubok Publishing S.L.
- Botello, S., Esqueda, H., Hernández, A., Zárate Araiza, J. F., & Valdés, J. G. (2007). *Estrategias evolutivas aplicadas en la optimización de problemas con interacción fluido-sólido*. *Revista internacional de métodos numéricos para cálculo y diseño en ingeniería*, 23(4), 415-428.
- Casillas, M. V., Puig, V., Garza-Castanón, L. E., & Rosich, A. (2013). *Optimal sensor placement for leak location in water distribution networks using genetic algorithms*. *Sensors*, 13(11), 14984-15005.
- Gestal, M., Rivero, D., Rabuñal, J. R., Dorado, J., & Pazos, A. (2010). *Introducción a los Algoritmos genéticos y la Programación genética*. A Coruña, 2010, 30-68.
- Igel, C., Suttrop, T., & Hansen, N. (2006, July). *A computational efficient covariance matrix update and a (1+ 1)-CMA for evolution strategies*. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 453-460). ACM.
- Molina, D., Herrera, F., & Lozano, M. (2007) *Algoritmo Memético con Intensidad de Búsqueda Local Adaptativa*.
- Puigjaner, L., Ollero, P., Prada, C., Jiménez, L., 2006. *Estrategias de modelado, simulación y optimización de procesos químicos*. Editorial Síntesis