



Universidad de Guanajuato
División de Ingenierías
Maestría en ciencias del Agua

“DETECCIÓN Y ESTIMACIÓN DE FUGAS EN REDES DE
ABASTECIMIENTO DE AGUA POTABLE MEDIANTE LA
METODOLOGÍA PARTICLE SWARM OPTIMIZATION (PSO) Y
EPANET”

Tesis de Maestría

Presenta:

Ing. Juan Pablo Ramírez Gutiérrez.

Director:

Dr. José de Jesús Mora Rodríguez.

Co-Directores:

Dr. Idel Montalvo.

Dr. Martín Jiménez.

Guanajuato, Gto., 8 de Mayo del 2026

A Mónica, Sophie y Paula; ejes de mi vida.

Agradecimientos

A mi familia, por su apoyo emocional y económico; no habría podido alcanzar esta meta más en mi vida sin ustedes.

A mi padre, el Ing. Sergio Pablo Ramírez Salmerón a quien le hubiera encantado estar presente en mi obtención de grado. Papá, donde quiera que estés, te dedico este logro.

A Mónica, por su ayuda en todo momento; desde el vaciado de datos en las simulaciones, hasta los innumerables debates (a veces incomprensibles) en los que le explicaba las modificaciones a la PSO.

Al Dr. José de Jesús Mora Rodríguez, por su valiosa ayuda en la obtención de recursos, por su orientación, por haber sido el motor principal en la elaboración de esta tesis y en la culminación de la misma; así como por haber tenido la visión inicial de este tema de investigación. Gracias Dr., por no haber quitado el dedo del renglón.

Al Dr. Idel Montalvo por su indispensable orientación en la comprensión de la PSO. No siempre se tiene la asesoría de un experto de clase mundial en el desarrollo de una tesis como lo fue en este caso.

Finalmente al M. Martín Jiménez por todas las atenciones y explicaciones que me brindó, por su apoyo en innumerables ocasiones. Fue un honor el haber conocido a un investigador tan brillante y tan comprometido con la docencia. Usted ha sido el mejor catedrático que he tenido durante mi vida académica.

>>> ÍNDICE

“Si no sabes a dónde vas, cualquier camino sirve...”

-Lewis Carrol

CAPÍTULO I. >>> Introducción	1
I.1) Objetivo de la Tesis.-	1
I.2) Metas.-	1
I.3) Hipótesis.-	2
I.4) Justificación.-.....	2
I.5) Antecedentes	3
1.6) Estructura de la tesis.-.....	6
CAPÍTULO II. >>> Generalidades del tema	9
II.1) Sistemas de abastecimiento de agua potable.-	9
II.1.1) Componentes de un sistema de abastecimiento	10
II.1.2) Tipos de Redes de Distribución.....	14
II.1.3) Componentes de una red de distribución	17
II.2) Problemática actual de los sistemas de agua potable en México.	19
II.3) Fugas en las redes de Agua Potable.....	20
CAPÍTULO III. >>> Particle Swarm Optimization (PSO)	24
III.1) Generalidades	24
III.2) Descripción de la PSO.....	26

III.3) Espacio de búsqueda, dimensiones, tipos de variables y partículas	31
III.4) Función Objetivo	34
III.5) Procedimiento de la PSO	36
III.6) Factores que impactan en el desempeño de la PSO	44
CAPÍTULO IV. >>> EPANET	54
IV.1) Generalidades	54
IV.2) Antecedentes	56
IV.3 Método del Gradiente (Ejemplo Práctico)	59
CAPÍTULO V. >>> Procedimiento de detección propuesto.....	80
V.1) Fundamentación.	80
V.2) Descripción.	85
V.3) Ejemplo Práctico.	90
V.4) Uso de la rutina propuesta.	107
CAPÍTULO VI. >>> Análisis y Modelado.....	129
VI.1) Consideraciones.-.....	129
VI.2) Experimentación con diversas metodologías.	133
VI.2.1.) Mejor partícula en la iteración (MPI).	136
VI.2.2.) Mejor partícula histórica (MPH).	144
VI.2.3.) Formación Anillo (PSOFA).....	150
VI.2.4.) Inicialización Hammersley (IH).	156
VI.2.5.) Inicialización velocidades por Half-Diff (IHD).	162
VI.2.6.) Variación de Velocidad Máxima (VVM).	166
VI.2.7.) Clamping PSO (CMP).	173
VI.2.8.) Inercia Dinámica (ID).	180

VI.2.9.) Reinicialización y Mutación con división de trabajo (RMDL).	186
VI.3) Modelado en otras redes.	197
VI.3.1) Resultados Obtenidos.	202
CAPÍTULO VII. >>> Conclusiones, Recomendaciones y Limitaciones ..	209
>>> Referencias	213
>>> ANEXOS	223

CAPÍTULO I

>>> INTRODUCCIÓN

“Un viaje de mil millas comienza con el primer paso....”

-Lao Tsé

I.1) Objetivo de la Tesis.-

El objetivo principal de la tesis consiste en: “Proponer un método eficiente para detectar y cuantificar el fenómeno de fuga en una red de abastecimiento de agua potable”.

I.2) Metas.-

Las metas de la tesis son las siguientes:

- Definir cómo se podrá detectar el fenómeno de fuga en una red de agua potable, qué parámetros se tomarán en cuenta y cómo determinar su magnitud.
- Elegir o diseñar la función objetivo para la detección de fugas. Dicha función deberá ser efectiva y de fácil evaluación para no impedir el desarrollo óptimo de la rutina.
- Analizar la metodología PSO, observar sus ventajas y desventajas, aplicaciones en el tema y modificaciones existentes.
- Detección e identificación de los factores que afectan el desempeño de la metodología PSO en la detección de fugas en redes de agua potable.
- Diseñar una rutina computacional sencilla y amigable que se pueda aplicar a múltiples problemas y escenarios.

- Realizar múltiples simulaciones hidráulicas con la rutina diseñada en diversos escenarios, con la finalidad de determinar un patrón de comportamiento.
- Evaluación de la metodología PSO en la detección y estimación del fenómeno de fuga. Proponer nuevas líneas de investigación, limitaciones y problemas encontrados.
- Proveer una investigación explícita, que le facilite al lector la comprensión de esta, para que pueda ser consultada o ampliada en investigaciones posteriores.

I.3) Hipótesis.-

“Mediante la técnica Particle Swarm Optimization (PSO) y EPANET, se podrá estimar la ubicación y magnitud de las fugas existentes en una red de abastecimiento de agua potable”.

I.4) Justificación.-

El problema de la escasez del agua es cada vez más preocupante, el agua dulce es un recurso limitado y a menudo escaso. La falta de agua cada vez más notable, la migración de la población hacia las ciudades y la urbanización, están causando estrés sobre los recursos hídricos.

México ocupa el lugar 53 de 180 países evaluados a nivel mundial en el grado de estrés sobre los recursos hídricos, con un 17.36% (FAO, 2011). Actualmente, a cada mexicano le corresponden 4,416 m³ de agua potable al año. Se contempla que para el 2030, solamente se tendrán 3,841 m³ por habitante (Gama et al., 2010).

En los países en vías de desarrollo, se tiene un problema puntual que es el agua no facturada. Esta se compone por las pérdidas físicas, el consumo no autorizado y los volúmenes suministrados sin ser cobrados.

En 2006, el Banco Mundial estimó que entre el 40% y 50% del agua producida en los países en vías de desarrollo, es agua no facturada. En cuanto a fugas físicas, se estima que el agua que se pierde diariamente en los sistemas de agua potable en países en vías de desarrollo, sería suficiente para abastecer a 200 millones de personas (Kingdom et al., 2006).

En los últimos años se han manifestado incrementos substanciales en la demanda del agua. La gente migra de las poblaciones rurales a los centros urbanos y genera un estrés en los servicios públicos al demandar mayores gastos del vital líquido. En México, la población en zonas urbanas mayores a 2,500 habitantes, aumentó del 42.6% en 1950 al 76.8% en el 2010 (INEGI, 2010). Para el año 2030, se estima que el 81% de la población en la República Mexicana se encuentre en localidades urbanas (CONAPO, 2007).

Se estima que el 85% del agua no contabilizada en México se debe a fugas en la red de distribución, alcanzando un 36.70% del agua total en la red (IMTA, 2010). La ciudad de México recibe 35.1 m³/s del sistema Cutzamala, Lerma y de pozos profundos. La SEMARNAT y CONAGUA han reportado que la ciudad de México pierde el 38% del agua total que recibe (13.3 m³/s). Este volumen de pérdidas es mayor a cualquiera de los volúmenes suministrados por cualquier fuente a la Ciudad de México (Sosa, 2010).

I.5) Antecedentes

La detección de fugas ha sido un tema de estudio recurrente en la comunidad científica a nivel mundial (Esquivel et al., 2010). Se tienen diferentes técnicas para la detección de fugas como los métodos acústicos, inyección de sustancias trazadoras en el agua, termografía, medios electromagnéticos, registradores de flujo y presión (Burn et al., 1999).

Actualmente, gracias a la era computacional, se ha desarrollado la detección de fugas mediante software. Los métodos utilizados por el software pueden ser divididos en cuatro categorías generales: Cambio de presión o flujo, análisis de presión en un punto, balance de masa o volumen y sistemas basados en modelos dinámicos (Zhang, 1996).

El software basado en el cambio de presión o flujo indica que una alta tasa de cambio en el flujo o en la presión indica una posibilidad de fuga. Esta tasa se compara con datos predefinidos y, de ser mayor, se genera una alarma (Zhang, 1996). Si, en esta modalidad se utiliza solamente el criterio de la presión, la metodología se denomina análisis de presión en un punto (Geiger, 2008).

El método por balance de masa o volumen se basa en el principio de conservación de la masa (Geiger, 2008). En este, se establece la diferencia de cualquiera de estos dos factores entre un punto aguas arriba y otro aguas abajo. Si la diferencia obtenida rebasa una tolerancia pre-establecida (por cambios de presión o temperatura), se emite una alarma de fuga. Este método permite la detección de volúmenes de fuga que no generen una gran tasa de cambio en el flujo o presión (Zhang, 1996).

Los sistemas basados en modelos dinámicos detectan fugas mediante discrepancias entre la modelación matemática y los valores medidos físicamente (Zhang, 1996).

En la última década, varios investigadores en el campo de la hidráulica han dejado de lado las técnicas de optimización tradicional y han recurrido a los algoritmos evolutivos como algoritmos genéticos, búsqueda de la armonía, colonia de hormigas, PSO, etc. (Montalvo et al., 2010).

Se tienen varias investigaciones referentes a la detección de fugas en redes presurizadas por algoritmos evolutivos. Diversas metodologías han sido empleadas en la resolución de esta problemática.

De Silva et al. (2011), usaron EPANET y máquinas de vectores de soporte para la detección de fugas en una red en Melbourne, Australia. El principio de detección se basó en la presión monitoreada en varios nodos de la red. A pesar de haber detectado fugas del orden de 90 l/hr, se concluyó que la efectividad del método en la práctica dependería de la precisión de EPANET para registrar pequeñas variaciones en el flujo.

Fuentes-Mariles et al. (2011), estimaron el gasto y la localización de fugas en una red de tuberías del laboratorio de hidráulica del Instituto de Ingeniería de la Universidad Nacional Autónoma de México (UNAM). El método heurístico aplicado fue algoritmos genéticos, el análisis hidráulico se realizó con EPANET. El factor tomado en cuenta para la detección de fuga fue la diferencia entre los gastos medidos físicamente y los gastos estimados por el algoritmo. En este trabajo se concluye que el método presentado es robusto, pues presenta un porcentaje de error bajo, aunque se recomienda su análisis posterior en una red no experimental.

Belsito et al. (1998), determinaron el tamaño y la ubicación de las fugas por medio de redes neuronales artificiales. En dicha investigación se detectaron fugas con un volumen equivalente al 1% del flujo de entrada y se localizaron fugas con una precisión del 50% en fugas pequeñas.

Izquierdo et al. (2010), realizaron la detección de fugas mediante la PSO con diversidad enriquecida en la Red de Hanoi. El método de detección de fugas se basó en la diferencia de presiones físicas y las obtenidas gracias a la PSO. Se concluyó que la PSO fue de gran utilidad para la detección de fugas, pues se tuvieron diferencias de presiones menores a 0.15 mca.

Como se puede observar, para la detección de fugas por métodos heurísticos se comparan las condiciones obtenidas por el algoritmo contra las condiciones físicas imperantes. Por requerir únicamente de datos de la red que, el organismo operador debe de tener a la mano por cuestiones de operación, servicio y mantenimiento (gasto y presión), resultan en un método económico para el pronóstico de fugas.

1.6) Estructura de la tesis.-

Para su presentación, esta tesis se dividió en siete capítulos. La intención del autor es el proporcionar toda la información necesaria para el análisis, crítica y desarrollo posterior de este tema de investigación.

El primer capítulo consta de los objetivos de la tesis, la hipótesis, los alcances, justificación del tema y los antecedentes.

El capítulo dos habla de los sistemas de abastecimiento de agua potable y sus características generales. El capítulo concluye mencionando el fenómeno de fuga, sus causas, las repercusiones que conlleva y los métodos de detección disponibles. La finalidad de este capítulo es el dar un panorama global de la problemática de estudio para ir entrando en materia. Nunca se pretendió entrar en detalles que, a juicio del autor, ya deben ser conocidos por el lector.

El tercer capítulo habla de la metodología Particle Swarm Optimization (PSO). En este capítulo se da una breve información acerca de los orígenes de esta metodología. Se intentó explicar de la manera más sencilla posible todos los detalles en el funcionamiento de la PSO, los factores a considerar y a que hacen referencia. También se mencionan los factores importantes que modifican el desempeño de la PSO y se citan diversos autores para que el lector pueda consultarlos si lo desea. Finalmente se cierra el capítulo con el pseudo-código de la PSO. Es importante mencionar que esta tesis no se adentra mucho en los

fenómenos sociales y teoría evolutiva que inspiraron a la PSO, si se desea un mayor conocimiento del tema se recomienda la lectura del libro “Swarm Intelligence” de Kennedy & Eberhart.

El capítulo cuatro indica las características principales del programa de revisión hidráulica EPANET. Se indican breves rasgos del programa, su interfaz y algunas características de este. Se desglosa a detalle el método del gradiente utilizado por EPANET mediante la solución de un ejemplo práctico. La intención del autor al incluir el método del gradiente es que, el lector pueda entender de manera analítica el proceso que el software sigue. Para mayor detalle del funcionamiento de EPANET se recomienda la lectura del manual del usuario que se puede obtener de manera gratuita, al igual que el software, en el sitio de la EPA. <https://www.epa.gov/water-research/epanet>

El quinto capítulo explica el procedimiento a seguir para la detección de fugas en una red de agua potable. Se explica la unión entre la PSO y el EPANET, se explica la función objetivo y se incluye la resolución de un ejemplo detallado para su mayor comprensión.

El capítulo seis inicia explicando el uso de la rutina elaborada por el autor, sus consideraciones y limitaciones. Posteriormente se explica el código inicial (PSO clásica), las modificaciones que se fueron realizando al código, se explica en qué consisten dichas modificaciones y los autores que las proponen. Se presentan los resultados obtenidos con cada una de las modificaciones y se discute si se tuvo alguna mejora y porqué.

El séptimo capítulo analiza el desempeño de la última rutina obtenida, se explican las limitaciones de la investigación y se indican trabajos a futuro.

En el apartado de anexos se incluye el código completo y explicado que se usó en la última rutina programada. Lamentablemente por cuestiones de espacio, no se anexan todas las rutinas usadas en esta investigación.

Si se desea obtener dicha información se puede solicitar de manera gratuita al e-mail del autor: jprg1981@gmail.com

CAPÍTULO II

>>> GENERALIDADES DEL TEMA

“La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general, pueden ser expresadas en un lenguaje comprensible para todos...”
-Albert Einstein

II.1) Sistemas de abastecimiento de agua potable.-

Una necesidad primordial humana es la disponibilidad de agua, ya sea para su consumo, por cuestiones de higiene, confort y hasta entretenimiento; el agua es fundamental para la subsistencia humana y el desarrollo del hombre. La localización y el desarrollo de los asentamientos humanos se han visto influenciados por la disponibilidad y cercanía de este vital líquido.

Desde tiempos ancestrales, el hombre ha inventado tecnologías para garantizar el suministro de agua a los centros poblacionales. En la era de bronce (2,800 a 1,100 A.C.), se desarrollaron los “qanats” (canales subterráneos artificiales en Persia), los cuales contenían un sistema de túneles y pozos diseñados para coleccionar y transportar agua a lo largo de grandes distancias (Mays et al., 2007). El historiador Pierre Grimal, en su libro “La civilización Romana”, indica la construcción de once acueductos para abastecer a Roma en el año 144 antes de Cristo.

El agua también es sinónimo de salud; el Dr. H. Mahler, director general de la Organización Mundial de Salud (WHO), en el foro de desarrollo de 1987 indicó “El número de grifos por cada mil habitantes es mucho mejor indicador del estado de salud de un país que el número de camas en los hospitales” (Billava & Kulkarni, 2012).

Por tales motivos, los ingenieros idearon una forma de suministrar agua potable a los centros poblacionales consistentes en diversas obras hidráulicas y de ingeniería que hacen posible el canalizar, sanear, conducir y distribuir el agua en cantidad, oportunidad y calidad.

El objetivo primordial de un sistema de abastecimiento es el satisfacer en tiempo, forma, calidad y cantidad, el agua necesaria para fines domésticos, industriales, comerciales, sanitarios, recreativos, etc. El sistema debe ser capaz de satisfacer las demandas a cualquier hora, en cualquier lugar, con la presión y calidad necesaria.

Un sistema de abastecimiento de agua potable consiste principalmente de obras de captación, almacenamiento, conducción, bombeo, tratamiento y distribución (CNA, 2007).

II.1.1) Componentes de un sistema de abastecimiento

Un sistema de abastecimiento de agua potable se compone de manera general de las siguientes partes (CNA, 2007):

- Obra de captación
- Obra de purificación
- Obra de regulación y control
- Obra de conducción
- Obra de distribución

La obra de captación comprende a la obra o conjunto de obras, y el equipamiento necesario, para captar el agua de una fuente natural de abastecimiento. Dependiendo de la fuente se clasifica el tipo de obra de captación. La obra de captación puede ser de una fuente de agua atmosférica, superficial, sub-superficial o subterránea (CNA, 2007). La naturaleza de la obra de captación depende directamente de las características particulares de cada fuente de extracción y de la zona donde se consumirá dicho líquido.

Las obras de captación de agua atmosférica comprenden el aprovechamiento del agua proveniente de la atmósfera, tal es el caso de la precipitación pluvial, nieve, granizo y escarcha.

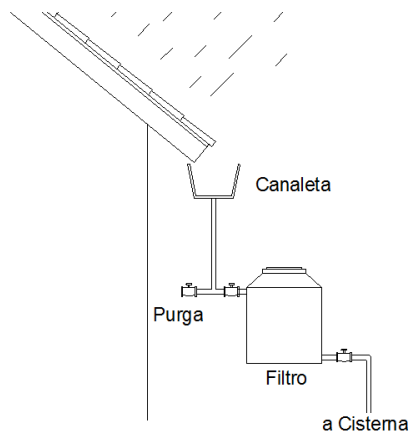


Fig. II.A.- Captación pluvial en casa-habitación

Se le denomina aguas superficiales a aquellas cuya superficie libre está sometida a la presión atmosférica, estas pueden provenir de corrientes temporales o perennes. Dentro de este rubro se contempla al agua proveniente de manantiales, ríos, presas de diferentes tipos y embalses naturales o artificiales.

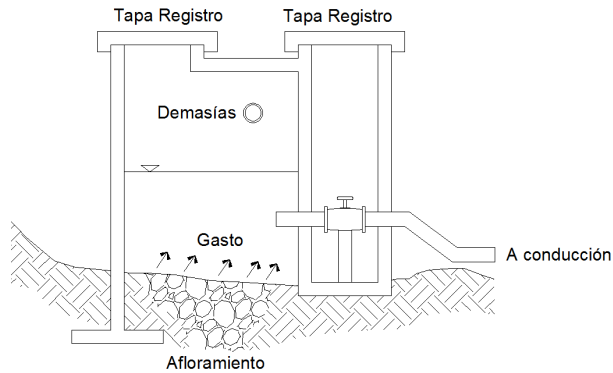


Fig. II.C.- Corte de obra de toma en manantial con afloramiento vertical

Las obras de captación de agua sub-superficial comprenden a galerías filtrantes, pozos someros, pozos radiales y sistemas de puyones. Estas obras captan el agua que se infiltra a una escasa profundidad. Por el efecto de la infiltración, esta agua es generalmente de buena calidad (CNA, 2007).

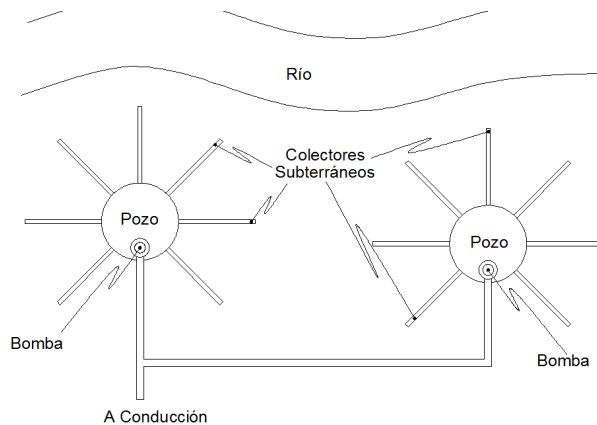


Fig. II.D.- Planta de Pozo radial o Ranney.

Aquellas obras en las que se requiere la perforación de pozos profundos para la extracción de agua de un acuífero se denominan obras de

captación de agua subterránea. Estos tipos de pozos pueden ser profundos o artesianos.



Fig.II.E.- Pozo Profundo y Artesiano.

La obra de purificación hace referencia a la infraestructura y al equipamiento necesario para purificar el agua extraída de la fuente natural de abastecimiento. Obviamente, una fuente natural de abastecimiento debe proveer agua limpia y apta para consumo humano, pero se necesita someterla a un proceso de desinfección para garantizar su calidad.

Dependiendo de la calidad del agua, se elige el método de tratamiento. Si solamente se desea eliminar la turbidez o bacterias, basta con una planta de filtración. En caso de que el agua tenga gran cantidad de sales, se trata con una planta suavizadora para eliminar su dureza. También hay plantas supresoras de hierro y manganeso, las cuales causan la precipitación de dichos elementos facilitando su remoción. En cualquier caso, se debe valorar la calidad del agua de la fuente natural de abastecimiento y ver si es viable el aprovechamiento de dicha fuente.

La obra de regulación y control comprende las obras y equipamiento necesarios que permiten el suministro de un gasto determinado. La fuente de abastecimiento no siempre dotará la misma cantidad de agua todo el tiempo, de igual manera, la población no demandará la misma

cantidad de agua a lo largo del día o del año. Es por esto que se debe contar con la infraestructura adecuada para evitar daños causados por un exceso de agua. Por otro lado, se debe garantizar un gasto de demanda aun cuando la fuente de abastecimiento no lo proporcione. Una obra característica de regulación y control es un tanque de regularización, su finalidad es la de recibir un gasto variable y convertirlo en un gasto constante. De esta forma se garantiza un suministro determinado y un funcionamiento hidráulico apropiado.

Una obra de conducción es aquella que conduce el agua desde la obra de regulación hacia una red de distribución. Ésta consta de un tubo de gran diámetro por el cual fluye el agua hasta la red de distribución. Dicho tubo no presenta bifurcaciones y debe garantizar el no alterar la calidad del agua. Dependiendo de la ubicación de la población a abastecer y de la obra de regulación, es que se determina si el agua se transporta por gravedad o si es necesario un equipo de bombeo.

El entramado de tuberías que abastecen el agua a cada propiedad de una población se denomina red de distribución. Su función principal es la de abastecer en tiempo, forma, calidad y cantidad el agua requerida por los consumidores. Dependiendo de la topografía del lugar, de la traza del asentamiento urbano, de la ubicación de tanques de almacenamiento y de las características hidráulicas de la obra es que se determina la configuración de la red de distribución.

II.1.2) Tipos de Redes de Distribución

Como ya se mencionó antes, hay diversos factores que determinan la configuración de la red de distribución. Por su forma, las redes de distribución se dividen en tres grandes grupos: redes abiertas, redes cerradas y combinadas.

Se denomina redes abiertas a aquellas en las que la tubería no forma un circuito; los tramos de tubería contienen ramales con extremos muertos que no se conectan a ningún otro tubo. Estas redes se presentan cuando no es posible el conectar dos tuberías paralelas entre sí por alguna restricción por motivos de propiedad, por no contener lotes cabeceros o por no considerarse económicamente viable.

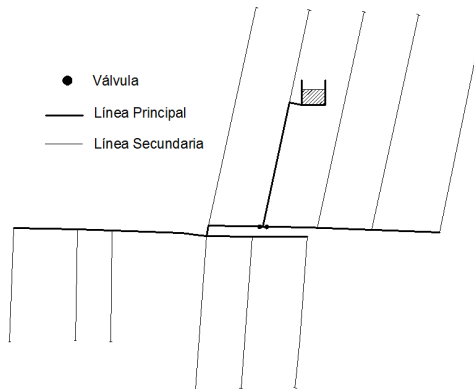


Fig. II.F.- Ejemplo de una red abierta.

Las redes cerradas son aquellas que tienen una configuración similar a una malla o retícula. La tubería se coloca de tal manera que forma un circuito. Una red cerrada es una mejor opción que una red abierta pues puede alimentar a un punto por varios tramos de tubería (al menos dos). Ésta es la principal ventaja de una red cerrada sobre una abierta.

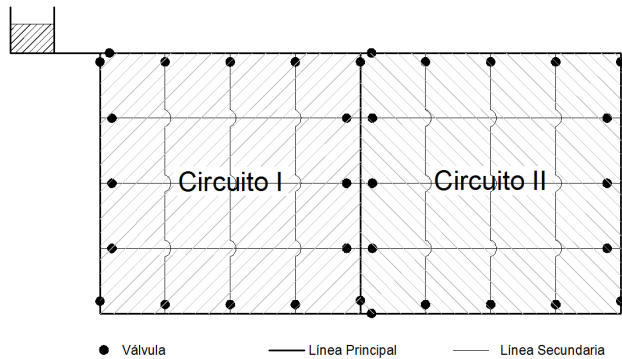


Fig. II.G.- Ejemplo de una red cerrada.

Cuando por algún motivo, en una red cerrada, se desea alimentar a una zona que sólo tiene un punto de acceso, se opta por una red combinada. Una red combinada se presenta cuando se mezcla una red cerrada con una red abierta por querer dotar algún punto en especial que no permite el trazo de una malla o circuito.

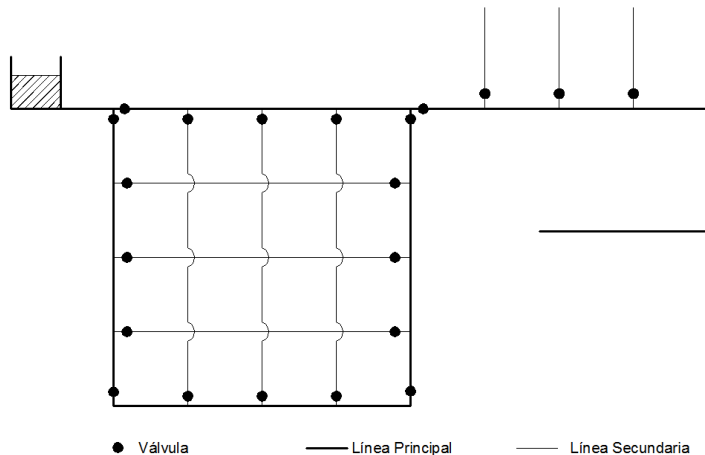


Fig. II.H.- Ejemplo de una red combinada.

Para elegir el mejor tipo de red de distribución se deben analizar múltiples factores además de los ya mencionados como: posibles problemas en presiones causados por la topografía, punto de

alimentación de la red de distribución, elementos adicionales en la red de distribución como puntos de tratamiento y tanques de almacenamiento intermedios, etc.

II.1.3) Componentes de una red de distribución

Los componentes de una red de distribución son los siguientes (CNA, 2007):

**** Tuberías:** Son aquellos elementos de sección circular por los que fluye el agua. Su finalidad es la de canalizar el agua evitando su contaminación, manteniendo una velocidad y una presión determinada. Se tienen tuberías con diferentes materiales como PVC, fierro fundido, asbesto cemento, concreto, acero, CPVC, etc. En el mercado se cuenta con una gran gama de tuberías disponibles en diferentes materiales y diámetros. La designación del diámetro depende casi exclusivamente del funcionamiento hidráulico: de las pérdidas admisibles de presión, de la presión deseada en algún punto, de la velocidad máxima admisible del líquido, etc. La determinación del material de la tubería depende de las características de la obra (clima, tránsito, etc.), de la localización y exposición de la tubería, de la velocidad máxima del líquido, entre otros.

Dependiendo de su funcionamiento hidráulico en la red de distribución, las tuberías pueden conformar una red primaria o una red secundaria. Se denomina red primaria a aquella que transporta los mayores caudales de agua y que alimenta a la red secundaria. La determinación de la red primaria se realiza en el cálculo hidráulico al establecer circuitos de flujo o líneas alimentadoras.

La red secundaria o ramales, hace alusión a aquellos tramos de tubería que reciben el agua de la red primaria y la distribuyen a los consumidores a través de las tomas domiciliarias. La red secundaria

contiene tuberías de menor tamaño que la red primaria, pero tiene una mayor longitud.

**** Piezas Especiales:** Se denomina piezas especiales a todos aquellos accesorios que por sí mismos, o en conjunción con otros elementos, permiten realizar cambios de diámetro, cambios de dirección, uniones, intersecciones, derivaciones, controlar el flujo, reducir el gasto, liberar presiones excesivas, expulsar el aire en la tubería, generar un punto terminal, etc. A los puntos donde se utilizan piezas especiales también se les conoce como cruceros. El material del que están hechos y la elección del mismo, depende de los mismos factores que ya se mencionaron antes con la tubería.

**** Válvulas:** Aunque también se pueden incluir dentro de las piezas especiales, las válvulas son aquellos dispositivos mecánicos que permiten el control, regulación, corte y disminución del flujo en un punto determinado. En general, y de acuerdo a su funcionamiento hidráulico, se clasifican en: válvulas de seccionamiento y válvulas de control. Las válvulas de seccionamiento son aquellas que permiten detener el flujo del agua en un punto dado. Las válvulas de control permiten regular el flujo, la presión, el ingreso o salida de aire en la tubería.

**** Hidrantes:** Son aquellas tomas en las que se puede extraer agua de la red. Se tienen dos tipos de hidrantes: los destinados al combate de incendios y los destinados a proveer agua a la población.

**** Tomas domiciliarias:** Es la conexión existente entre la red municipal de distribución y un predio, su función es la de dotar al predio de agua. Consiste por diversas piezas especiales y mangueras. El tipo de toma domiciliaria depende de la demanda que requiera el consumidor. En este punto se destaca la ubicación del medidor de agua potable, el cual permite aforar el gasto consumido por el usuario.

II.2) Problemática actual de los sistemas de agua potable en México.

De acuerdo con la ONU, se tiene una presión moderada sobre los recursos hidrológicos en México (Tello, 2007). Un panorama general del estado imperante en México se indica en la publicación “Situación del Subsector Agua Potable, Alcantarillado y Saneamiento. Edición 2011” publicado por la Comisión Nacional del Agua (CNA, 2011).

Algunos factores que dificultan la cobertura del agua potable en México son la orografía accidentada, el hecho de que el 25% de la población nacional se concentre en localidades menores a los 2,500 habitantes, el crecimiento poblacional, la concentración de la población en el centro de la república, entre otros.

De acuerdo con datos del XIII Censo de Población y Vivienda realizado por el INEGI, 10 millones de personas no cuentan con servicio de agua entubada correspondiente al 9.10% de la población nacional.

En 23 estados de la república se tienen coberturas superiores a la media nacional, destacando los Estados de Aguascalientes, Coahuila, Colima y Tlaxcala con porcentaje superior al 98%. En Chiapas, Guerrero y Oaxaca la cobertura es inferior al 78%.

De los 329.3 m³/s de agua suministrada a nivel nacional, se estima que 205 m³/s (62.2%) provienen de fuentes subterráneas; el resto del suministro se obtiene de fuentes superficiales.

Los centros urbanos de mayor desarrollo económico se concentran en el centro y norte de la República a pesar de que en estas zonas se cuentan con pobres recursos hídricos. Por este motivo, se tiene que canalizar agua de otros lugares para abastecer a dichos centros urbanos. Mientras tanto, los habitantes de la zona sur de la República son los que disponen de menos agua y pagan más por ella, a pesar de que esta es la zona con más agua en la República (Tello, 2007).

Las zonas urbanas presentan los niveles más altos en la cobertura del servicio con 9.20 millones, equivalentes al 95.40%. En las zonas rurales se tiene una cobertura del 77.20%. Al 2010, se tiene un porcentaje de agua suministrada y desinfectada del 97.40%

Al 31 de diciembre de 2010, existen 645 plantas potabilizadoras en operación con una capacidad instalada de 135 m³/s que procesan un caudal de 91.7 m³/s, principalmente aguas superficiales (CNA, 2011). Se estima que solamente el 5% de los cuerpos de agua en México presentan excelente calidad (Tello, 2007).

Especialistas indican que las plantas potabilizadoras en México no funcionan adecuadamente, además de que el agua se contamina en las tuberías, cisternas o tinacos (Tello, 2007)

En cuestión de fugas, las redes de abastecimiento de agua potable en México presentan volúmenes de entre el 30% y 40% (Madrigal et al., 2012). La edad de las tuberías, el hundimiento del suelo en zonas donde se ha sobreexplotado el agua y la excesiva presión causan la aparición de nuevas fugas.

II.3) Fugas en las redes de Agua Potable.

Una fuga se puede definir como la diferencia entre la cantidad de agua suministrada a la red menos el agua suministrada a los consumidores. Los rebales en los tanques de almacenamiento y regulación, las tomas clandestinas, las filtraciones que emanan las tuberías, etc., forman parte de las fugas de agua en la red.

Una fuga puede ocasionarse por múltiples factores, entre algunos de ellos están los siguientes (Farrer, 1979):

****Alta presión en la red:** La presión afecta principalmente a los cruceros de la red. Cuando se tiene una fisura menor, la presión hace que ésta crezca. Una fuga depende directamente de la presión. Esto es:

$$Q = K(H)^X$$

En la que Q hace alusión al gasto, H es la presión, K depende de la longitud, diámetro y fricción del tubo y X es un exponente que está en función de la apertura de la fuga (Mora-Rodríguez, 2011). Por lo tanto, se puede concluir que, a mayor presión, mayor volumen de fuga.

**** Corrosión:** Esta puede ser interna o externa. La primera la ocasiona el líquido que está siendo transportado, tal es el caso de las aguas corrosivas que debilitan las tuberías metálicas y generan fugas. La corrosión externa es aquella que genera el suelo sobre la tubería, degradándola hasta debilitarla.

**** Tráfico vehicular:** Las tuberías cuyas uniones son rígidas, son las más susceptibles a dañarse por los efectos del tráfico vehicular. También se puede presentar el caso de que la tubería, por un error en la instalación, no se encuentre a una profundidad necesaria y tenga que soportar las cargas del tráfico.

**** Movimientos del suelo:** Las tuberías presentan fugas cuando algún sismo de gran intensidad les genera un desplazamiento o esfuerzo. También se tienen suelos que, por sus características propias, cambian su estructura y dañan a la red de agua potable, tal es el caso de las arcillas.

**** Defectos en el material:** Un material de baja calidad merma la vida útil de una tubería, genera reparaciones costosas y frecuentes que redundan en un desperdicio de agua. En el costo por reparación de una

tubería en una calle asfaltada, el 24% corresponde solamente a la tubería, siendo el 76% restante, los gastos de mano de obra y obra civil.

**** Deficiencias en la instalación y operación:** La mano de obra no calificada es otro motivo para que la vida útil de la tubería no sea la adecuada, tanto en la instalación como en la operación de la red. También se puede incluir en este punto la falta de equipo o herramienta para la instalación de la tubería.

**** Golpe de ariete:** El golpe de ariete presentado por cortes eléctricos o deficiencias en la operación de la red generan sobrepresiones que producen fracturas en las tuberías y desplazamientos de los atraques.

**** Edad de la tubería:** Con la edad se empieza a presentar desgaste en la tubería volviéndose más vulnerable a presentar fugas. Los niveles de falla aumentan con la edad de la tubería (Burn et al., 1999).

El problema de las fugas es muy complejo pues la gran mayoría de estas no son superficiales o apreciables a simple vista. En muchas ocasiones una fuga no se manifiesta y, si lo hace, se manifiesta en un lugar muy distante del que se encuentra. El principal problema no es que tanta agua sale por la fuga, sino el tiempo que se tarda en detectarla (Gama et al., 2010).

Existen varios métodos de detección de fugas como: (Burn et al., 1999)

**** Métodos acústicos.** Consiste en la detección de fugas por medio de ruido o vibraciones. Estos son generados por el agua cuando escapa del tubo a presión. Se tienen definidas las diferentes frecuencias que genera el agua. Su efectividad depende mucho de la pericia del usuario

**** Inyección de gas trazador en el agua.** Se inyectan gases no tóxicos, insolubles en el agua y más ligeros que el aire. Comúnmente se usa helio

e hidrógeno. El gas se inyecta y sale por las grietas, fisuras o zonas porosas de la tubería, atraviesa el suelo y es detectado por sensores.

**** Termografía.** El agua que escapa por la tubería cambia la temperatura del suelo adyacente a este. Este cambio de temperatura se puede detectar por escáneres infrarrojos.

**** Radar.** Se envía un pulso electromagnético al suelo por medio de un radar. El pulso rebota parcialmente cuando encuentra una diferencia entre dos materiales. Con estos datos se puede calcular la profundidad del cambio. Las fugas se pueden localizar mediante los espacios en el suelo que el agua haya generado al momento del flujo o mediante el cambio de densidad por un suelo saturado por el agua.

**** Registradores de presión y flujo.** Los registradores de flujo y presión pueden ir en el interior o en el exterior de la tubería. Pueden trabajar por detección de ruido, mediante algún dispositivo mecánico, etc. Se tienen muchas variedades de registradores, estos pueden ser automatizados, con conexión a SMS, etc.

De los métodos mencionados, los de inspección electromagnética, de termografía y de radar son muy costosos. Los métodos con sustancias trazadoras y con sensores dentro de la tubería aumentan el riesgo de contaminación por la intrusión de elementos y sustancias en el agua. El método acústico es el más usado en la actualidad (Burn et al, 1999), pero su desventaja es que, factores externos como ruido ambiental, por tránsito vehicular, por equipos de bombeo, etc, tiende a dificultar su efectividad (Tafari et al., 1997). Por su parte, las mediciones de gasto en una red de tuberías de agua potable son complicadas de realizar (Fuentes-Mariles et al., 2010).

CAPÍTULO III

>>> PARTICLE SWARM OPTIMIZATION (PSO)

“La única verdad científica sobre la que estoy totalmente seguro es que somos unos ignorantes sobre todo lo relativo a la naturaleza”.

-Lewis Thomas.

III.1) Generalidades

En los últimos años, se han desarrollado técnicas computacionales para emular la inteligencia natural y la organización de estructuras biológicas. ¿Cómo un pájaro evita chocar con otros?, ¿Cómo las hormigas saben qué camino tomar?, ¿Por qué los peces forman cardúmenes?, ¿Por qué los animales han evolucionado de la forma que lo han hecho hasta nuestros días?. Preguntas como estas son las que pretende responder la computación basada en procesos biológicos.

Profesionales en psicología han tratado de encontrar reglas, comportamientos y normas que definan como trabaja la psique para poder explicar el origen de los fenómenos y la formación de las estructuras sociales. En este punto convergen especialistas de diversas áreas del conocimiento: computación, ingeniería, biología, psicología. Todos estos profesionales buscan una manera de comprender la

inteligencia dominante en los organismos vivos para poder aplicarla en sus respectivos campos de estudio.

La computación basada en organismos vivos regularmente contiene una variable estocástica. Esta aleatoriedad es una forma de explicar algo que el hombre todavía no ha podido llegar a comprender (Kennedy & Eberhart, 2001). Cuando el ser humano no se puede explicar algo, se recurre a la aleatoriedad. Eventos que consideramos aleatorios alguna vez, ahora están perfectamente comprendidos y analizados, esto es, no existe tal cosa como la aleatoriedad en el Universo.

Los métodos heurísticos computacionales son una nueva forma de abordar un problema extremadamente complejo. Cuando un problema no se puede resolver o resulta muy complejo de solucionar por un método analítico o determinístico, se realiza una simulación de la realidad: un método heurístico. Así como lo expresan Romanycia & Pelletier en su artículo “What is a heuristic?” (1985):

“la heurística en inteligencia artificial es cualquier dispositivo, sea un programa, regla, pedazo de conocimiento, etc., el cual no es confiable por completo en que proveerá una solución práctica, pero el cual da una razón para creer que será útil, y el cual se agrega para la solución de un problema con la expectativa de que, por lo general, ayude con la solución de este”.

Siendo que los métodos computacionales heurísticos contienen variables estocásticas, se resuelven por medio del método “prueba y error”. El mayor problema en estos métodos es el tiempo necesario para encontrar una solución aceptable. Se ha buscado la forma de encontrar los lineamientos óptimos para resolver un problema de manera eficiente y rápida.

A manera de ejemplo, considerando el análisis del diseño de una red de agua potable muy sencilla, cuya configuración solamente comprenda 15 tramos de tubería con opción de utilizar 4 diámetros diferentes, tendríamos 4^{15} permutaciones posibles. Esto es, tendríamos que revisar 1'073'741,824 redes hidráulicas provenientes de todas las permutaciones posibles, con base en los 4 diámetros diferentes, para poder determinar cuál red es la idónea. Considerando que un equipo de cómputo analice una red por segundo, se necesitarían 34 años, 1 hora, 9 minutos y 18 segundos para resolver dicho problema. Obviamente, la complejidad crece exponencialmente si aumentamos el número de tramos de tubería, los diámetros que podríamos utilizar o el material del que está compuesta la tubería.

Por tal motivo, se necesitan definir métodos que nos ayuden a optimizar la búsqueda de soluciones viables para un problema. También se necesitan identificar los criterios a buscar para descartar todas aquellas soluciones inviables en un proceso de búsqueda.

Otra cuestión a tener en cuenta en los métodos heurísticos es que, como contienen variables aleatorias, es casi imposible lograr obtener el mismo resultado dos veces. La aleatoriedad genera que el método contenga una parte “innovadora”, esta es la causante de que tal vez nunca se pueda volver a obtener el mismo procedimiento.

III.2) Descripción de la PSO

Particle Swarm Optimization (PSO) es un proceso estocástico y heurístico que simula la inteligencia natural emergente en sociedades organizadas ante un problema. Se inspiró en el vuelo que realizan las aves que migran a un rumbo desconocido por cuestiones climáticas.

James Kennedy, un psicólogo social, junto con Russel C. Eberhart, un ingeniero eléctrico, desarrollaron dicha técnica en 1995 (Bai, 2010). La

PSO empezó como una simulación de un medio social simplificado. La intención original era el graficar la coreografía de una parvada. Sus raíces se basan en la inteligencia artificial y en la computación evolutiva.

Inicialmente, se intentó emular a la parvada indicándole al algoritmo que, todas las partículas debían igualar la velocidad de sus vecinos, en la dirección X y Y. Los autores notaron que, en un momento determinado, el algoritmo se estancaba generando un movimiento uniforme y sin variación. Fue entonces cuando decidieron agregar un factor aleatorio llamado “locura”. Este factor le daba una apariencia más natural al algoritmo.

Mediante simulaciones y la introducción de nuevas variables, Kennedy & Eberhart se dieron cuenta de que, se podía remover la variable “locura”, al introducir una influencia social y experiencia personal. Al hacer esto, el código se empezó a comportar como un enjambre en lugar de hacerlo como una parvada; esta característica le dio su nombre a la metodología (Kennedy & Eberhart, 1995).

La inteligencia emergente es aquella que surge de elementos de menor complejidad, en la que se puede incluir diversos elementos: lenguaje, memoria, razonamiento (Miner et al., 2009). Existen otros factores muy importantes que generan dicha inteligencia, como lo es la interacción social. Todavía se desconoce mucho del tema, pero se ha visto que la falta de interacción social repercute directamente en el desarrollo de la inteligencia. Incluso, se tienen registrados casos en los que, el desapego de la madre hacia su niño, causa deficiencias en el desarrollo cognitivo del infante (Dörr & Banz, 2010).

Se han desarrollado estudios tratando de entender que genera el conocimiento en un individuo. Sin importar la diferencia en los pasos o nombres, no se puede pasar por alto que la observación, la interacción social y el análisis forman parte del aprendizaje; se ha observado que el

aprendizaje humano es fuertemente formado de acuerdo con las expectativas que rigen a una cultura (Engeström & Sannino, 2012).

Ante un factor adverso (muchas veces no esperado), una organización biológica, basada en la experiencia y en la percepción de cada uno de sus integrantes, debe decidir qué acción tomar. La PSO se basa en tres principios fundamentales del aprendizaje: la evaluación, la comparación y la imitación (Kennedy & Eberhart, 2001).

La evaluación es un proceso clave en la optimización. Sólo mediante la evaluación se puede discernir si el camino que se ha tomado es el correcto o qué tanto se ha mejorado con una u otra decisión. Esto ayuda a evitar repetir los errores o las malas decisiones que se han tomado con anterioridad. Las partículas de la PSO deben ser capaces de evaluarse a sí mismas. Solamente mediante la evaluación se tiene una verdadera capacidad de análisis de los resultados obtenidos, de esta manera, se genera un aprendizaje.

Los integrantes de cualquier organización social siempre se están comparando con sus semejantes. Las calificaciones académicas, índices de pobreza, nivel de riqueza, coeficiente intelectual, edad, etc., son medios de comparación para ubicar a un individuo dentro de una población. A través de la comparación se puede observar que medidas son más efectivas para lograr un mejor desempeño. Quizá sea por cuestiones evolutivas (la supervivencia del más apto), pero la comparación genera un estímulo hacia la superación (Corcoran et al., 2011). En la PSO, las partículas se comparan con sus vecinos para determinar que tan bien o tan mal se han ido desempeñando y poder elevar así su rendimiento.

En el aprendizaje, la imitación forma parte fundamental para el desarrollo de nuevas habilidades o aptitudes. Desde la infancia, se tiende a imitar las acciones que se ven y eso va generando habilidades y patrones de comportamiento. Un niño imita lo que ve, de ahí va

asimilando como funciona una sociedad. Por naturaleza, tendemos a imitar a nuestros vecinos con mejor desempeño, generándose así, una superación personal. La imitación en la PSO permite el aventajar un camino a recorrer, gracias a las experiencias que se comparten entre partículas vecinas. Éste es un punto clave en la metodología, pues una computadora podrá evaluar y comparar, pero adolece de imitación y aceptación de nuevas teorías e ideas.

La dinámica que sigue la PSO es la siguiente: Un grupo social tiene un problema determinado. Cada integrante del grupo obtiene una solución al problema (evaluación). Los integrantes analizan cuál fue la mejor solución de todas las propuestas (comparación). Todos los integrantes propondrán una nueva solución, pero esta vez tienen una influencia adicional: la social (imitación). La experiencia propia y la influencia social influyen para tomar la próxima decisión.

En la figura III.A se puede ver como el movimiento de cada partícula en la PSO tiene dos influencias, la experiencia personal (P_{best}) y la influencia social (G_{best}). Si sólo se tomara en cuenta la experiencia propia, se llegaría un óptimo local. Por tal motivo, la siguiente posición a tomar por la partícula, será una combinación del mejor resultado obtenido por la partícula y por el grupo.

Haciendo una analogía: Un grupo de amigos se van de campamento al bosque. La primera ocasión que van a acampar no saben bien que cosas llevar; estos pueden llevar objetos innecesarios y omitir algunos imprescindibles. Probablemente, la primera vez pasen hambre, incomodidad o frío, tal vez no hayan elegido la ropa adecuada o hayan llevado una cantidad insuficiente o inadecuada de alimento. Al haber experimentado algún inconveniente, habrán evaluado qué elementos hicieron falta y cuáles fue buena idea el haber llevado.

Al regreso del campamento, platicarán sus experiencias con sus familiares y conocidos, quienes les darán nuevos consejos para acampar

(comparación). La próxima vez que vayan a acampar, no sólo tendrán la experiencia propia del campamento anterior (factor individual), sino la experiencia platicada por sus familias (factor social). Con estos datos, ellos podrán determinar qué elementos necesitan para acampar y hacer su estadía cada vez más confortable.

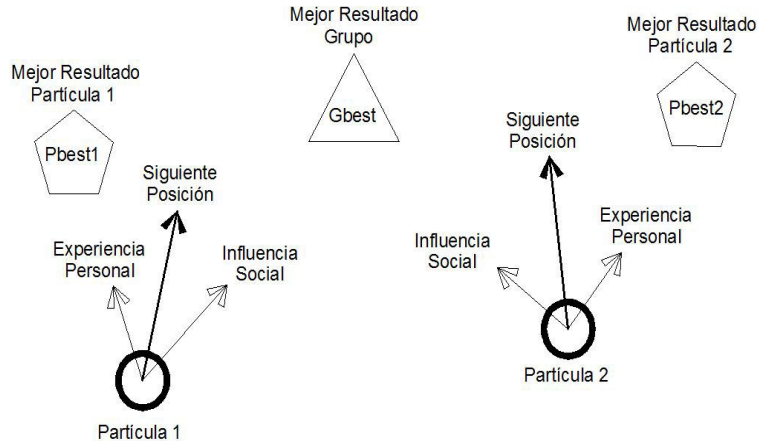


Fig III.A.- Influencias sobre una partícula

En los procesos sociales se tiene una capacidad de decisión propia. No todas las personas o integrantes de un grupo son fácilmente influenciables. Existen personas altamente influenciables que harán todo lo que un grupo les dice que hacer, inclusive por encima de sus propias ideas. Habrá otras personas obstinadas que harán lo que ellos creen sin importar lo que los demás le recomienden hacer. La PSO intenta emular esta diversidad, permitiéndoles a los integrantes el dar mayor peso al factor individual (experiencias propias) o al factor social (influencia del grupo).

Entre las ventajas más notables de la PSO están las siguientes: 1) Se puede aplicar en investigación científica o problemas de ingeniería. 2) El cálculo de la PSO es muy sencillo y de fácil implementación. 3) La PSO

permite el uso de números reales, lo que ahorra tiempo por no tener que decodificar los resultados (Bai, 2010).

Entre las mayores desventajas de la PSO están las siguientes: 1) Como la mayoría de las técnicas estocásticas, la PSO sufre de convergencia prematura, especialmente en la solución de problemas complejos de múltiples dimensiones (Wei & Cao, 2011). 2) El método no puede trabajar fuera de problemas de optimización (Bai, 2010).

La función principal de la PSO es el encontrar un máximo o un mínimo en un espacio de búsqueda; esto es, la PSO nunca intenta resolver un problema. Por tal motivo, el problema en cuestión, deberá plantearse de tal forma que nos proporcione una solución al encontrar un máximo o un mínimo.

III.3) Espacio de búsqueda, dimensiones, tipos de variables y partículas

Antes de pasar al procedimiento de cálculo de la PSO, se tendrá que identificar cuál es el Universo de soluciones posibles (espacio de búsqueda), los elementos que deberá contener una solución (dimensiones) y si se trata de un problema con variables continuas o discretas.

Para todo problema existe un espacio de búsqueda. Además de elegir las soluciones posibles, se debe identificar el número de elementos que debe tener la solución (dimensiones). Por ejemplo, si se quiere armar una dieta alimenticia diaria con base en una ingesta calórica, el espacio de búsqueda contendrá todos los alimentos que pueden conformar una dieta: ensalada, huevos con jamón, licuado, fruta picada, carne asada, quesadillas, pescado frito, arroz, frijoles, jugo, pan tostado, caldo de pollo, camarones, etc.

Habiendo identificado el universo de soluciones posibles, se tendrá que determinar el número de elementos que debe tener una solución (dimensiones). Si se busca una dieta diaria, el número de elementos que tendrá una solución serán tres (desayuno, comida y cena). De esta manera, se tendrán las siguientes soluciones posibles: 1) Huevos con jamón, pescado frito, quesadillas 2) Quesadillas, carne asada, fruta picada, 3) Huevos con jamón, carne asada, fruta picada, 4) Licuado, pescado frito, fruta picada, 5) Pan tostado, camarones, huevos con jamón y así sucesivamente.

Habrán algunos problemas en los que no importe el orden ni la repetición de los elementos (permutación), habrá otros casos en los que el orden y la no repetición sea crucial (combinación). Siguiendo con el ejemplo de la dieta alimenticia, muy probablemente no se deseen camarones como cena por su alto contenido calórico, tampoco se querrá comer sólo un licuado, mucho menos se deseará desayunar, comer y cenar pan tostado. Por este motivo, se tiene que tener muy bien identificado el problema que se está intentando resolver.

En este ejemplo hablamos de 3 dimensiones para cada solución (desayuno, comida y cena); esto es, cada propuesta planteada tendrá tres elementos. En caso de que solamente se busquen soluciones para la comida, se hablaría de una sola dimensión. Es importante hacer notar que no todos los problemas tienen las mismas dimensiones. En ingeniería, la mayor parte de los problemas tienen muchas dimensiones.

Ahora bien, existen dos tipos de problemas: los que tienen una solución con variables continuas y los que se resuelven con variables discretas. Una variable continua es aquella en la que se puede establecer un rango numérico y no importa qué valor tome, pues cualquier número puede solucionar dicho problema.

Un ejemplo de un problema que se resuelve con variables continuas es la determinación de cómo puede gastar dinero un individuo al salir por la

noche. Si se tienen \$800 para gastar en una noche y se quiere ir a dos lugares, se tendrán las siguientes soluciones: 1) \$0 en el primer lugar y \$800 en el segundo, 2) \$100 en el primer lugar y \$700 en el segundo, 3) \$234.56 en el primer lugar y \$ 565.44 en el segundo, etc. En este ejemplo, el espacio de búsqueda comprende todas las cantidades existentes entre \$0 y \$800; se tienen 2 dimensiones.

Un problema que se resuelve con variables discretas es aquél en el que la variable no puede tomar todos los valores posibles dentro de un rango porque algunas cantidades no son viables o son inexistentes.

Suponiendo que 5 amigos quieren ir en bote a una isla pero el bote solamente tiene 3 plazas y aguanta 200 kg. ¿Cómo se deberán organizar para ir todos a la isla si su peso es: Pablo (65 kg), Pedro (80 kg), Sergio (90 kg), Ana (60 kg), Martha (70 kg)?. Este es un ejemplo de un problema con variables discretas, no se puede hacer un viaje con Pedro, Sergio y una mitad de Ana. Por esta razón, la variable solamente podrá tener los siguientes valores: 60, 65, 70, 80 y 90. La variable no podrá seleccionar ningún otro valor diferente. Las dimensiones de este problema son tres.

Habiendo identificado el problema, se debe determinar el número de partículas que buscarán una solución. Una partícula es una solución potencial. En la PSO, el usuario puede indicar el número de partículas que atacarán el problema. Cada partícula obtiene una solución del problema y al final se determina cuál obtuvo la mejor. No necesariamente un mayor número de partículas obtendrá un mejor resultado del problema. En el punto III.6 se darán más detalles de este punto.

En la figura III.B se pueden observar dos partículas en un espacio de búsqueda. En este ejemplo, las partículas poseen 14 dimensiones cada una. Por lo que, en lugar de aparecer 2 puntos, se grafican 14 puntos por partícula. Los puntos rojos hacen alusión al mejor resultado encontrado hasta el momento, los puntos azules representan otras propuestas de

solución que no han obtenido un buen resultado pero que han sido sujetos de análisis. Como se puede apreciar, el eje Y hace alusión al espacio de búsqueda que, en este caso en particular, comprende todos los números reales de 0 a 25. El eje X representa el número de iteración en el que las partículas buscaron en dichos puntos.

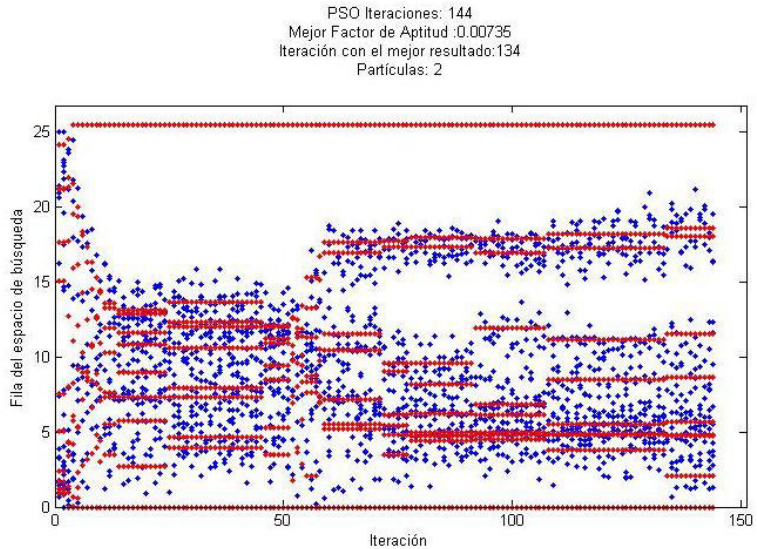


Fig. III.B.- Espacio de Búsqueda

III.4) Función Objetivo

Como ya se comentó con anterioridad, la PSO no busca la solución de un problema, solamente busca máximos o mínimos en un espacio de soluciones.

La función objetivo no forma parte de la PSO, pero se requiere para poder evaluar el desempeño de las partículas y poder expresar la solución de un problema como un máximo o un mínimo.

La función objetivo (también llamada factor de aptitud), es una expresión matemática que se tiene que idear para poder evaluar las partículas. La PSO permite la búsqueda de máximos y mínimos en un problema. ¿Cuándo serán máximos y cuándo mínimos?, pues depende de lo que se esté buscando.

Retomando el ejemplo de la dieta, si se busca una dieta de 1080 calorías diarias, se puede obtener la diferencia entre las calorías que proporciona una solución y las que se desean obtener:

$$Fa = (1080 - \text{Calorías aportadas por solución})^2$$

Suponiendo que la PSO brinda las siguientes dietas, y evaluando los factores de aptitud para cada una de ellas se tiene lo siguiente:

Solución 1 = 1070 cal ; $Fa = (1080 - 1070)^2 = 100$

Solución 2 = 1000 cal ; $Fa = (1080 - 1000)^2 = 6,400$

Solución 3 = 1079 cal ; $Fa = (1080 - 1079)^2 = 1$

Como se puede ver, mientras más se acerque la solución brindada por la PSO al resultado que se busca, el factor de aptitud será menor. Este será entonces un problema donde se buscará un mínimo; mientras menor sea el factor de aptitud de una partícula, mejor resultado será. El resultado ideal se obtendrá cuando el factor de aptitud sea cero. Entonces, una partícula con el menor factor de aptitud encontrado hasta el momento, será la partícula que haya obtenido el mejor resultado.

Ahora, este mismo problema se puede expresar como la búsqueda de un máximo si se cambia la fórmula del factor de aptitud:

$$Fa = \frac{1}{(1080 - \text{Calorías aportadas por solución})^2}$$

A pesar de ser el mismo problema, se puede observar que los mejores resultados tendrán un valor cercano a uno. Los valores que contengan mayor error tendrán factores de aptitud más pequeños.

Al ser imprescindible la función objetivo para usar la PSO, se debe buscar que esta no sea muy compleja y genere un “cuello de botella” que limite el algoritmo. De la misma manera, una buena función objetivo es crucial para obtener una buena solución al problema en cuestión.

III.5) Procedimiento de la PSO

El procedimiento de la PSO se puede dividir en 3 pasos principales: Inicialización de partículas, evaluación de partículas, obtención de nuevas posiciones y velocidades.

La inicialización de partículas hace alusión a determinar una posición inicial X_{ij} y una velocidad V_{ij} para cada dimensión j de cada partícula i . Han existido múltiples propuestas para inicializar las partículas en la PSO (que se discutirán en el subcapítulo III.6), un método general es el siguiente:

$$X_{ij} = rand() * espbusq$$

Siendo:

X_{ij} = Posición de la dimensión j que es parte de la partícula i .

$rand()$ = Número aleatorio con distribución normal entre cero y uno.

$espbusq$ = Rango de valores del espacio de búsqueda.

En la ecuación anterior se define la posición para una dimensión de la partícula en cuestión. Se obtiene un número aleatorio y se multiplica por el valor máximo del espacio de búsqueda, de esta forma se obtiene una posición dentro de éste. Obviamente, esta operación se deberá hacer ij

veces (siendo i el número de partículas y j las dimensiones del problema).

Si por algún motivo, se desea que la posición se seleccione aleatoriamente de entre un rango $[X_{\min}, X_{\max}]$, la fórmula a usar será:

$$X_{ij} = rand() * (X_{\max} - X_{\min}) + X_{\min}$$

Donde:

X_{ij} = Posición de la dimensión j que es parte de la partícula i .

$rand()$ = Número aleatorio con distribución normal entre cero y uno.

$X_{\max}$ = Máxima posición que puede adoptar la dimensión.

$X_{\min}$ = Mínima posición que puede adoptar la dimensión.

La velocidad en una partícula se puede expresar como un cambio o un incremento en la posición ($V = X + \Delta X$). La velocidad le indica a la partícula, que tanto se debe mover de su posición actual en la siguiente iteración. Este proceso es muy importante, pues permite el encontrar una buena solución en un menor tiempo. Para evitar posibles confusiones, se decidió descartar el término “incremento en la posición” y asignar “velocidad” (Kennedy & Eberhart, 2001).

Para inicializar la velocidad de la partícula, se debe definir antes una velocidad máxima y una velocidad mínima. Se ha observado que, debido a la aleatoriedad de la PSO, la velocidad tiende a adquirir una gran aceleración (Kennedy & Eberhart, 2001). Esto ocasionaría que las partículas chocaran contra los límites del espacio de búsqueda.

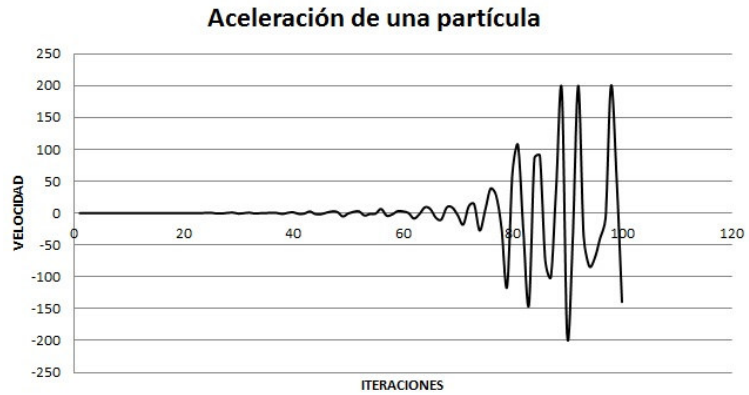


Fig. III.C.- Aceleración de una partícula sin restricción.

La velocidad ha sido tema de debate entre muchos autores (Helwig et al., 2009., Cai et al., 2009., Barrera & Coello, 2009., Kennedy et al., 2001). Una buena velocidad le asegura a la partícula que explorará el espacio de búsqueda y podrá obtener un mejor resultado global. Se verá este tema más a detalle en el subcapítulo III.6. De manera general, se pueden establecer los límites de la velocidad como:

$$V_{\max} = X_{\max}$$

$$V_{\min} = -V_{\max}$$

Donde:

$V_{\max}$ = Velocidad máxima que podrá alcanzar una partícula.

$V_{\min}$ = Velocidad mínima que podrá tener una partícula.

$X_{\max}$ = Máximo valor que una partícula puede asumir como posición.

Ahora bien, para inicializar las partículas, se les debe asignar una velocidad inicial. De manera general, se tiene que:

$$V_{ij} = rand() * (V_{\max} - V_{\min}) + V_{\min}$$

Donde:

V_{max} = Velocidad máxima que podrá alcanzar una partícula.

$rand()$ = Número aleatorio con distribución normal entre cero y uno.

Se define la velocidad máxima y mínima para una dimensión de la partícula. Se obtiene un número aleatorio y se multiplica por el rango comprendido entre las velocidades permitidas, posteriormente se le suma la velocidad mínima. De esta forma se obtiene un valor entre la velocidad máxima y la velocidad mínima. Esta operación se deberá hacer ij veces.

La fig. III.D. muestra como cada dimensión de una partícula posee diferente velocidad y posición. También se puede apreciar que, en la siguiente iteración, se habrán actualizado todas las posiciones y velocidades para todas las dimensiones de todas las partículas. Obviamente, todas las velocidades estarán dentro de los límites V_{min} y V_{max} . El conjunto de datos X_{10} , X_{11} y X_{12} proporcionan una sola solución para la partícula 2 en la iteración $i + 1$.

Habiendo inicializado todas las partículas, se procede a la evaluación de las mismas y se identifica cuál ha tenido el mejor desempeño. En este punto entra la función objetivo explicada en el subcapítulo III.4.

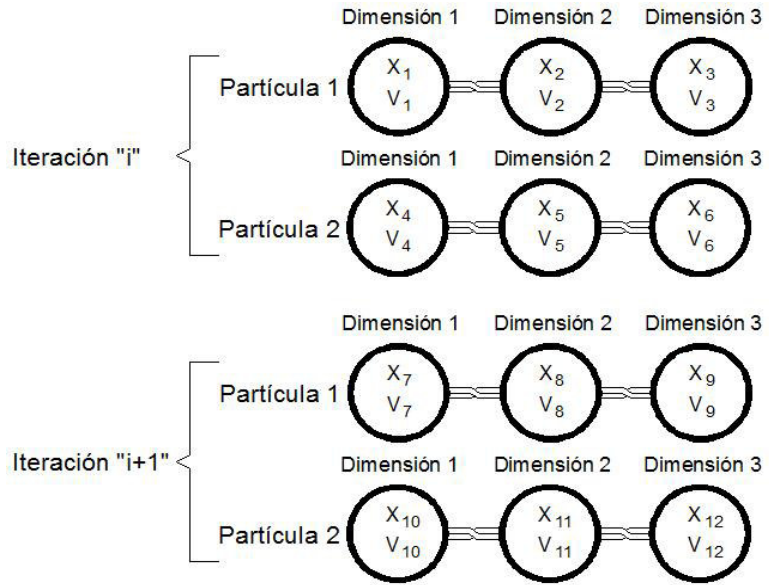


Fig III.D.- Cambio de posiciones y velocidades por dimensión

Después de haber evaluado todas las partículas y haber identificado cuál fue la mejor de todas. Se procede a calcular sus nuevas posiciones y velocidades. La velocidad de la partícula se actualiza de la siguiente manera:

$$V_{ij(t+1)} = V_{ij(t)} + \varphi_1 [P_i - X_{ij(t)}] + \varphi_2 [P_g - X_{ij(t)}]$$

Donde:

$V_{ij(t+1)}$ = Velocidad calculada para la siguiente iteración.

$V_{ij(t)}$ = Velocidad actual.

φ_1 = Confianza individual.

P_i = Mejor posición obtenida por el individuo hasta el momento.

$X_{ij(t)}$ = Posición actual.

φ_2 = Influencia social.

P_g = Mejor posición del grupo obtenida hasta el momento.

La velocidad contiene 3 componentes principales. El primer componente $\{ V_{ij(t)} \}$ hace alusión a la inercia que ya lleva la partícula. El segundo componente $\{ \varphi_1 [P_i - X_{ij(t)}] \}$ es la experiencia individual. El tercer componente $\{ \varphi_2 [P_g - X_{ij(t)}] \}$ es la sugestión social.

El factor de inercia se refiere a que tan rápido ha ido viajando la partícula anteriormente, esta característica contribuye a que la partícula vaya adquiriendo mayor velocidad con cada iteración. El segundo componente analiza la posición que tiene esa partícula contra la mejor posición que ha alcanzado, este se multiplica por un factor aleatorio que podrá darle mayor o menor ponderación. El tercer término es el social, éste analiza la mejor posición que ha alcanzado el grupo contra la posición que tiene actualmente la partícula. El componente social, al igual que el individual, se multiplica por un número aleatorio.

La idea principal de multiplicar los componentes por un número aleatorio es el que a veces se le dé una mayor importancia a la influencia del grupo y, en otras ocasiones, se le dé una mayor ponderación a la experiencia propia. Como regla general, se pueden asignar los siguientes valores a los factores aleatorios (Kennedy & Eberhart 2001):

$$\varphi_1 = 2 * rand()$$

$$\varphi_2 = 2 * rand()$$

Donde:

φ_1 = Confianza individual.

φ_2 = Influencia social.

rand() = Número aleatorio con distribución normal entre cero y uno.

Como la velocidad tiende a aumentar dramáticamente, se recomienda controlarla mediante la siguiente condición:

$$V \max < Vij_{(t+1)} < V \min$$

Por lo que tendremos que establecer:

$$Si \quad Vij_{(t+1)} < V \min \rightarrow Vij_{(t+1)} = V \min$$

$$Si \quad Vij_{(t+1)} > V \max \rightarrow Vij_{(t+1)} = V \max$$

Dentro de la búsqueda de soluciones, se desea que la partícula tenga una combinación entre exploración y explotación. La exploración es la característica que le permite a la partícula el explorar muchos puntos en el espacio de búsqueda sin concentrarse en un área determinada. Cuando se ha explorado lo suficiente, se desea que la partícula realice la explotación de la zona. Una explotación es una búsqueda exhaustiva en un área local.

Si una partícula tiene buena exploración y mala explotación estará “brincando” de un lado a otro sin producir un buen resultado. Si una partícula tiene buena explotación y mala exploración, se concentrará mucho en un área y dará un buen resultado local que no garantiza que sea el mejor de todos. Se busca que las partículas tengan un buen balance entre ambas características, de ahí la importancia de regular la velocidad con la que se mueven las partículas.

Teniendo la velocidad calculada, se procede a determinar la nueva posición de la partícula:

$$Xij_{(t+1)} = Xij_{(t)} + Vij_{(t+1)}$$

Donde:

$Xij_{(t+1)}$ = Posición de la partícula en la siguiente iteración.

$Xij_{(t)}$ = Posición actual de la partícula.

$Vij_{(t+1)}$ = Velocidad de la partícula en la siguiente iteración.

Habiendo determinado la nueva posición de la partícula, el ciclo vuelve a comenzar: se evalúan las nuevas posiciones de las partículas, se asignan nuevas velocidades, se asignan nuevas posiciones, etc., esto se hace hasta que se alcance el factor de aptitud deseado o hasta un n número de iteraciones indicadas por el usuario.

En resumen, el pseudo código de la PSO es el siguiente:

Definir espacio de búsqueda [X_{min} , $X_{máx}$]

Definir velocidades máximas y mínimas [V_{min} , V_{max}]

Definir el número de partículas “ i ”

Inicializar las posiciones X_{ij} y las velocidades V_{ij} de todas las partículas

```

for  $i = 1$  until “número de partículas”
    for  $j=1$  until “número de dimensiones”
         $X_{ij} = (X_{max}-X_{min})*rand()+X_{min}$ 
         $V_{ij} = (V_{max}-V_{min})*rand()+V_{min}$ 
    end
end

```

While $iteraciones < iteraciones\ definidas\ por\ el\ usuario$

```

    Ejecutar la función objetivo y determinar  $P_i$  de cada partícula y  $P_g$ 
    if factor aptitud ==0 then “fin de ciclo”

```

Calcular velocidades V_{ij} y las posiciones X_{ij} de todas las partículas

```

for  $i = 1$  until “número de partículas”
    for  $j=1$  until “número de dimensiones”
         $C_1 = rand()*2$  ;  $C_2 = rand()*2$  ;
         $\phi_1 = C_1*rand()$  ;  $\phi_2 = C_2*rand()$  ;
         $V_{ij(t+1)} = V_{ij(t)} + \phi_1 [ P_i - X_{ij(t)} ] + \phi_2 [ P_g - X_{ij(t)} ]$ 
        if  $V_{ij(t+1)} > V_{max}$  then  $V_{ij(t+1)} = V_{max}$ 
        if  $V_{ij(t+1)} < V_{min}$  then  $V_{ij(t+1)} = V_{min}$ 
         $X_{ij(t+1)} = X_{ij(t)} + V_{ij(t+1)}$ 
    end
end

```

end
iteraciones = iteraciones +1
loop

III.6) Factores que impactan en el desempeño de la PSO

De manera general se mencionarán algunos factores que impactan el desempeño de la PSO. Se enunciará su importancia y se citarán algunos autores que han tratado dicho tema.

- **Número de Partículas**

El número de partículas hace alusión a las posibles soluciones simultáneas que se buscan durante una iteración. A pesar de lo que se pudiera pensar, un mayor número de partículas no necesariamente refleja un mejor resultado.

El número de partículas depende de las dimensiones del problema y de las características de este. Zhang et al. (2005) indican que, para problemas con muchas dimensiones, se recomienda usar 50 partículas, mientras que para problemas con pocas dimensiones, 30 partículas son una buena opción.

Kennedy & Eberhart indican que no existe una respuesta certera en este aspecto pero que el uso de un enjambre entre 10 y 50 partículas suele trabajar bastante bien (Kennedy & Eberhart, 2001).

Maurice Clerc estableció un Standard PSO (SPSO) en el cual el tamaño del enjambre se calcula mediante la fórmula (Clerc, 2010):

$$S = \text{Int}(10 + 2\sqrt{D})$$

Donde:

S= Tamaño del enjambre.

Int= Función que devuelve un número entero.

D= Dimensiones del problema

Según estudios realizados por Shi y Eberhart, el tamaño del enjambre no repercute seriamente en la efectividad de la PSO (Ratnaweera et al., 2004).

Como se puede apreciar, el número de partículas depende del tipo de problema que se esté tratando. De cualquier forma, se debe tener en cuenta de que un mayor número de partículas generará un mayor tiempo de cómputo, pues en cada iteración, cada partícula provee una solución potencial que deberá ser evaluada por la función objetivo. Por este motivo el número de partículas debe ser tal que provea un equilibrio entre eficiencia y rapidez.

- Forma del enjambre.

La forma del enjambre hace alusión al intercambio de información entre partículas durante el vuelo. Esto influye en la componente social que modifica la velocidad de cada partícula en su búsqueda.

Se tiene la formación estrella o gbest. Esta formación permite que todas las partículas comparen sus posiciones actuales; así se elige la mejor posición alcanzada por cualquier partícula. De esta manera, todo el enjambre es influenciado por la posición de la partícula con la mejor posición actual. Esta formación favorece la explotación pues todas las partículas comparten la misma información (Parsopoulos & Vrahatis, 2008).

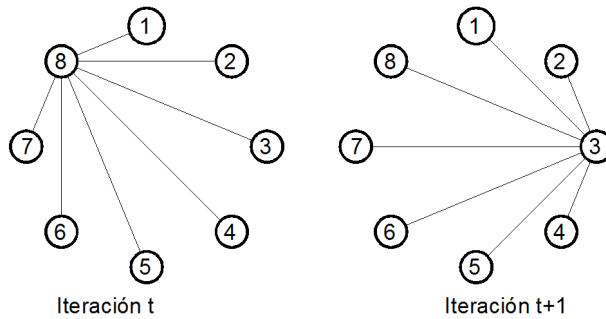


Fig.III.E.- Formación estrella o Gbest.

La formación anillo es aquella en la que las partículas se comunican solamente con sus dos vecinos adyacentes. De esta forma, la partícula n solamente comparará su posición contra sus vecinos $n-1$ y $n+1$. Esta formación favorece la comunicación local entre las partículas evitando una convergencia prematura de la PSO (Bratton & Kennedy, 2007). Esta formación favorece la exploración (Parsopoulos & Vrahatis, 2008).

Maurice Clerc propone que, si el enjambre no obtiene soluciones globales, cada partícula seleccione aleatoriamente a los K vecinos con los que ejercerá su influencia social; sugiere que K adopte el valor de 3 (Clerc et al., 2010). Esto es, cada partícula tendrá comunicación con solamente 3 vecinos.

También se ha propuesto la creación de múltiples sub-enjambres y que solamente algunas partículas de cada uno de estos tengan comunicación con los demás sub-enjambres (Mendes et al., 2004).

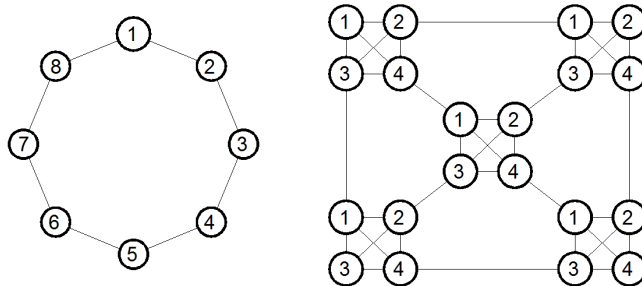


Fig. III.F.- Formación anillo (izquierda), formación con sub-enjambres (derecha).

Una técnica más inspirada en la naturaleza consiste en la migración de partículas. En esta se generan múltiples sub-enjambres y después de ciertas condiciones establecidas, existe migración de partículas de un sub-enjambre a otro, así se establece una riqueza adicional por diversidad (Ma et al., 2012).

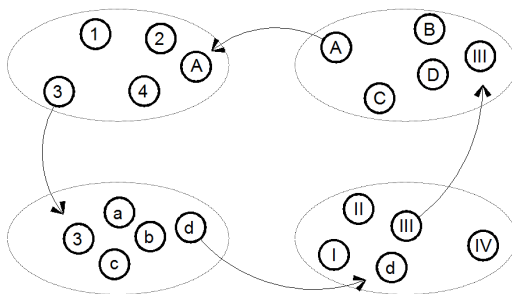


Fig.III.G.- Sub-enjambres con partículas migratorias.

Se tiene también una propuesta de un vecindario dinámico. Este método consiste en que, en cada generación, las partículas calculen la distancia existente entre ellas y las demás, posteriormente se eligen las n partículas que estén más cerca de ellas (se recomienda usar $n=2$). Este método funciona mejor con una mayor población (Hu & Eberhart, 2002).

- Inicialización en la posición de partículas.

La inicialización de partículas comprende la primera posición que tendrán las partículas en el espacio de búsqueda. Una buena inicialización permite ubicar de manera mejor distribuida a las partículas. Esto se hace con la finalidad de que sea más probable el encontrar la solución.

La PSO clásica ubica a las dimensiones de las partículas de manera aleatoria en el espacio de búsqueda $[X_{\max}, X_{\min}]$. Esto es (Kennedy & Eberhart, 2001):

$$X_{ij} = rand() * (X_{\max} - X_{\min}) + X_{\min}$$

La inicialización por el método de Hammersley-Halton, se usa para distribuir uniformemente un número de partículas en un espacio determinado. El método aleatorio proporciona resultados con ruido, los métodos con muestreo regular generan una tendencia que requiere de varios muestreos para su reducción. El método de Hammersley-Halton genera vectores direccionados con distribución uniforme, los cuales tienen una mejora significativa en términos de error (Wong et al., 1997).

Otro método de inicialización es el de teselados centroidales de Voronoi. Este método puede ser explicado como la partición del espacio de búsqueda en compartimientos. Un grupo de puntos en el espacio son designados para ser “los generadores”, los cuales se encuentran en el centro de sus compartimientos correspondientes. Una desventaja de este método es que requiere de un costo computacional alto (Richards & Ventura, 2004).

El método del Caos genera posiciones a través de la siguiente fórmula:

$$X_{k+1} = L * X_k * (1 - X_k)$$

Donde L es un parámetro de control que varía entre $[3.56, 4.00]$, de esta manera se obtiene una variable caótica X . El valor inicial X_0 debe ser un valor arbitrario entre $(0,1)$ pero nunca debe ser igual a 0.25 , 0.50 ni 0.75 (Ren & Zhong, 2011).

El método de inicialización opuesta pertenece al aprendizaje basado en lo opuesto (Opposite Based Learning). Se tienen pruebas matemáticas que demuestran que este método aumenta las probabilidades de encontrar una solución al ubicar a las partículas cerca de un óptimo global (Gutiérrez et al., 2011). La fórmula es la siguiente:

$$X_0 = X_{\min} + X_{\max} - X$$

Primero se inicializa aleatoriamente el enjambre con una distribución uniforme entre cero y uno. Entonces la población opuesta se calcula reflejando la anterior con base en el valor promedio de cada dimensión.

- Inicialización en la velocidad de partículas.

Ya se mencionó la inicialización de posiciones en las partículas, de la misma manera se debe elegir una velocidad inicial en las dimensiones de las partículas. Es importante mencionar que este punto no es tan trascendente como la inicialización de posiciones.

La PSO clásica determina las velocidades en cada una de las dimensiones de las partículas de manera aleatoria entre el rango permitido de velocidades $[V_{\max}, V_{\min}]$. Esto es (Kennedy & Eberhart, 2001):

$$V_{ij} = rand() * (V_{\max} - V_{\min}) + V_{\min}$$

El método “Half-Diff” es un método que sirve para determinar la velocidad inicial de las dimensiones de las partículas. Su fórmula es (Clerc, 2010):

$$V(0)_d = 0.50[U(X \min_d, X \max_d) - X(0)_d]$$

- Velocidad.

La velocidad hace alusión al cambio de posición de las partículas, se puede expresar como la diferencia entre la posición actual y la posición anterior (Kennedy & Eberhart, 2001).

Como se vio en el subcapítulo anterior, la velocidad de la PSO se calcula a través de la velocidad anterior V_t , los coeficientes C_1 y C_2 , la posición anterior de la partícula X_t , la mejor posición del grupo P_g y la mejor posición de la partícula P_i .

$$V_{ij(t+1)} = V_{ij(t)} + \phi_1[P_i - X_{ij(t)}] + \phi_2[P_g - X_{ij(t)}]$$

Los valores límites para la velocidad se fijan con V_{max} , la cual es la máxima velocidad de vuelo que puede adquirir una partícula durante su búsqueda. Como se mencionó anteriormente, el no limitar la velocidad de vuelo causa una explosión en esta.

El valor de V_{max} a menudo se fija entre el 10% y 20% del rango dinámico de la variable en cada dimensión (Eberhart & Shi, 2001). En el libro “Swarm Intelligence” se recomienda que V_{max} adopte el valor de 4, aunque también se indica que, para determinar este factor, se debe tener un conocimiento previo del problema (Kennedy & Eberhart, 2001).

Con la finalidad de eliminar el término V_{max} , se introdujo el concepto de inercia W en 1998 por Shi & Eberhart. A pesar de que no se pudo eliminar el factor de V_{max} , se pudo observar que la PSO funciona eficientemente si se iguala la velocidad máxima con el máximo valor del espacio de búsqueda en cada dimensión, esto es $V_{max} = X_{max}$ (Eberhart & Shi, 2001). Desde entonces se han propuesto diferentes maneras de

manipular el factor de inercia W . Considerando el factor de inercia W , la velocidad se expresa de la siguiente manera:

$$V_{ij(t+1)} = W_{ij} * V_{ij(t)} + \varphi_1 [P_i - X_{ij(t)}] + \varphi_2 [P_g - X_{ij(t)}]$$

La finalidad de manipular el factor de inercia W consiste en proporcionarle a la PSO un balance entre exploración y explotación. Shi & Eberhart implementaron una fórmula para que W decrezca linealmente de 0.90 a 0.40 con el tiempo (Eberhart & Shi, 2001). La fórmula se expresa de la siguiente manera (Ratnaweera et al., 2004):

$$W = (W_1 - W_2) * \frac{(MaxIter - iter)}{MaxIter} + W_2$$

Donde W_1 es el máximo valor del factor de inercia (0.90), W_2 es el mínimo valor de inercia (0.40), $MaxIter$ es el número máximo de iteraciones a alcanzar, $iter$ es el número de iteración actual.

Se encontró que el variar linealmente la inercia no es muy efectivo al tratar con sistemas dinámicos. En este caso, la determinación de W se realiza de la siguiente manera:

$$W = 0.50 + \frac{rand()}{2}$$

Donde $rand()$ es un número aleatorio con distribución normal entre el rango [0,1]. Esta fórmula fue inspirada por el factor de constricción de Maurice Clerc, y se recomienda que C_1 y C_2 se fijen en 1.494 (Eberhart & Shi, 2001).

Los coeficientes de aceleración C_1 y C_2 le dan mayor peso a la componente social o a la personal. Normalmente estos coeficientes tienen un valor cercano a 2 (Eberhart & Shi, 2001).

Ratnaweera et al. (2004), proponen una PSO cuyos coeficientes de aceleración C_1 y C_2 varíen con el tiempo. Esto es:

$$C_1 = (C_{1f} - C_{1i}) \frac{iter}{MaxIter} + C_{1i}$$

$$C_2 = (C_{2f} - C_{2i}) \frac{iter}{MaxIter} + C_{2i}$$

Donde C_{1f} , C_{1i} , C_{2f} , C_{2i} son constantes, $iter$ es el número de iteración actual y $MaxIter$ es el número máximo de iteraciones. Se observó una mejor obtención de resultados si C_1 varía de 2.50 a 0.50 y C_2 varía de 0.50 a 2.50 (Ratnaweera et al., 2004).

Maurice Clerc propuso un factor de constricción para asegurar la convergencia de la PSO.

$$V_{i+1} = K * [V_i + C1 * rand() * (P_i - X_i) + C2 * rand() * (P_g - X_i)]$$

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}; \quad \varphi = C1 + C2; \quad \varphi > 4.0$$

Normalmente se fija φ en 4.10, por lo tanto $K = 0.729$. Ahora, si $C1=C2=2.05$, entonces C_1 y C_2 son iguales a 1.49445 (Eberhart & Shi, 2001).

- Procedimiento de la PSO

A pesar de no ser un factor de la PSO, el procedimiento de esta ha sido modificado por varios autores con la finalidad de obtener mejores resultados a diversos problemas.

El método PSO con predicción de velocidad utiliza a la velocidad como pronosticador y a la posición como corrector (Cui & Zeng, 2005). Ratnaweera et al. (2004), proponen la PSO Jerárquica con aceleración variante. En esta, se establece una variación del factor de inercia y además se modifica la velocidad de una dimensión de una partícula si se cumple una probabilidad de mutación.

La PSO con división de trabajo propone varios cambios pues genera cuatro sub-enjambres con diferentes fórmulas para el cálculo de la velocidad. Además, genera la posibilidad de obtener un mejor global mediante una distribución gaussiana y mediante la reinicialización de partículas (Wei & Cao, 2011).

CAPÍTULO IV

>>> EPANET

*“Una máquina puede hacer el trabajo de cincuenta hombres ordinarios. Ninguna máquina puede hacer el trabajo de un hombre extraordinario”.
-Elbert Hubbard.*

IV.1) Generalidades

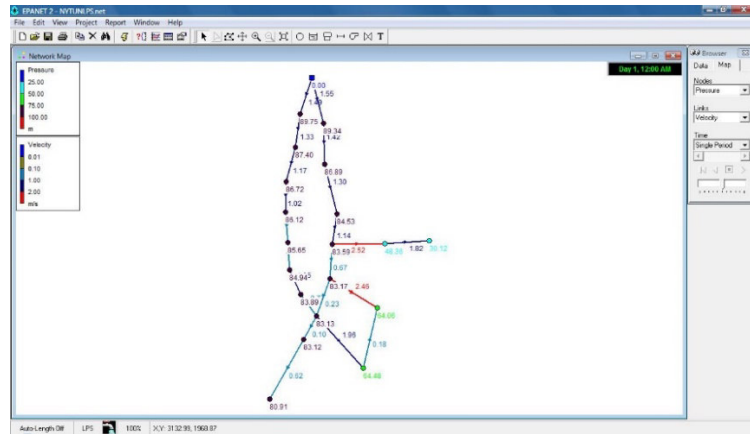
EPANET es un programa para computadora de acceso público desarrollado por la división de Suministro de Agua y Recursos Naturales de la Agencia de Protección al Ambiente de los Estados Unidos de Norteamérica (EPA).

EPANET permite realizar el análisis hidráulico y de calidad del agua en redes presurizadas. Este programa fue creado en 1993 y, en 1999, se tuvo una nueva versión del software (Ormsbee, 2006). El programa se puede descargar de manera gratuita del sitio web de la EPA.

El diseño de este software se realizó con la finalidad de ser una herramienta de investigación para ayudar a entender el movimiento y el destino de los constituyentes del agua en una red presurizada (Rossman, 1999).

Las capacidades de modelación hidráulica en EPANET son (Rossman, 2000):

- a) No se tiene un límite en el tamaño de la red a analizar.
- b) Se calculan pérdidas por fricción mediante la fórmula de Hazen-Williams, Darcy-Weisbach o Chezy-Manning.
- c) Considera pérdidas menores por acoplamientos, cambios de dirección, etc.
- d) Puede modelar bombas con velocidad variable o constante.
- e) Calcula la energía y costo de bombeo.
- f) Modela diversos tipos de válvulas como no retorno (check), controladora de flujo, reguladora de presión, de aislamiento, etc.
- g) Permite que los tanques de almacenamiento tengan cualquier forma (por ejemplo, el diámetro puede variar con la altura).
- h) Considera la posibilidad de establecer diferentes categorías en la demanda de los nodos, cada una con su propio patrón de variación en el tiempo.
- i) Permite el modelado de emisores dependientes de la presión, como rociadores o aspersores.
- j) Permite basar la operación del sistema mediante el nivel de agua en los tanques, mediante un temporizador, mediante ambos o con reglas de funcionamiento más complejas.



EPANET permite la introducción gráfica de tuberías, válvulas, nodos, tanques, reservorios y emisores. De la misma forma, se pueden introducir datos adicionales como variación en las demandas en el tiempo, curvas de bombeo, variación en los niveles de los tanques, etc.

Mediante un archivo txt, EPANET permite la introducción de datos para graficar una red de agua potable, para el cálculo de ésta o para ambas. Gracias al archivo ejecutable “epanet2d.exe”, se puede obtener un reporte con el análisis hidráulico sin necesidad de dibujar la red o incluso sin ejecutar el EPANET en ambiente Windows.

El método usado en EPANET, para la resolución de las ecuaciones de continuidad del flujo y pérdida de carga, es un método híbrido nodo-circuito llamado “Método del Gradiente” (Rossman, 2000).

IV.2) Antecedentes

En 1936 un profesor de la Universidad de Illinois, el ingeniero en estructuras Hardy-Cross, inventó un método para determinar la distribución de momentos en una estructura estáticamente

determinada. Posteriormente, aplicó dicho método en la solución de redes de distribución de agua potable con circuitos (Savic, 2007).

El método de Cross tiene dos metodologías: la corrección de caudales en tuberías y la corrección de cargas en los nodos. El método se consideró eficiente, pero se requería tener una muy buena primera estimación de las condiciones de la red para que este convergiera (Cross, 1936).

Con la aparición de las computadoras, se observaron algunas deficiencias en el método de Cross pues, dependiendo de la complejidad de la red en cuestión, tomaba mucho tiempo la convergencia de datos o, incluso, no se tenía convergencia alguna (Ormsbee, 2006).

Desde entonces, se desarrollaron nuevos métodos para combinar el análisis de flujo con las cargas hidráulicas. De esta manera, el análisis de redes se vio resumido en los siguientes métodos: Hardy-Cross, Newton-Raphson, Teoría líneal y, finalmente, gradiente hidráulico (Saldarriaga, 1998).

Estos métodos son iterativos y, aprovechan las capacidades de cálculo que tienen las computadoras en la actualidad. Se inicia con una estimación inicial de las condiciones de la red y se analizan hasta que no se tenga una variación sustancial en las presiones y/o gastos calculados.

El método de Cross con corrección de caudales se basa en suponer la magnitud y dirección de los gastos que fluyen por las tuberías de la red. Conforme el análisis avanza, se van ajustando los gastos supuestos. Este procedimiento requiere menos iteraciones que el método de Cross con corrección de cargas, pero su principal desventaja es la necesidad de que el calculista proporcione una buena estimación de los gastos iniciales en las tuberías.

El método de Cross con corrección de cargas (cabezas) se creó gracias a una modificación de R.J. Cornish. En este método, se suponen las cargas

en los nodos y se van corrigiendo conforme se avanza en el análisis. Este proceso se repite hasta que la ecuación de continuidad adquiere valores muy cercanos a cero en todos los nodos. La convergencia de este método, comparado con el de Cross con corrección de caudales, es más lenta. A pesar de esto, este método resulta más útil que el de corrección de caudales (Saldarriaga, 1998).

En 1969 Alvin Flower y Robert Epp implementaron las fórmulas de Newton-Raphson para resolver las correcciones en el método de Cross (Ormsbee, 2006). Inicialmente inventado por Sir Isaac Newton y Joseph Raphson para encontrar las raíces de una función, este método permite la solución de ecuaciones no lineales explícitas o no explícitas. Al comienzo se suponen las cargas en todos los nodos de la red. El proceso termina cuando la variación en las cargas entre iteraciones, es prácticamente nula.

D.J. Wood y C.O. Charles desarrollaron el método de la energía lineal en 1972. Se basa en la linealización de las ecuaciones de energía en cada una de las tuberías de la red. La conservación de la masa y de la energía son resueltas simultáneamente para obtener el gasto en cada tubo. Debido a sus características, este método no necesita una estimación inicial aproximada de gastos; se pueden suponer los gastos que se deseen sin que esto influya en el tiempo de solución. Tiene una mejor convergencia que el método de Cross y de Newton-Raphson (Saldarriaga, 1998).

En 1987 Todini y Pilati propusieron una modificación al método de Newton-Raphson. Este nuevo método considera que se cumplen la ley de la conservación de la masa en todos los nodos de la red y la ley de la conservación de la energía en todos los circuitos (Saldarriaga, 1998). A este nuevo procedimiento se le llamó el método del gradiente. Al usar matrices inversas, se requieren menos recursos computacionales para su almacenamiento, se realiza el cálculo en un menor tiempo y permite la fácil inclusión de elementos como válvulas (Rossman, 1999). El método

del gradiente es el más utilizado en los software comerciales (Savic, 2007)

IV.3 Método del Gradiente (Ejemplo Práctico)

Considerando a “NT” como el número de tuberías en la red a analizar, a “NN” como el número de nodos sin carga conocida en la red, y a “NS” como el número de nodos con carga conocida, podemos describir el procedimiento del método del gradiente de la siguiente manera:

1. Identificación de “ H_0 ”. Nodos con carga conocida fija. Tamaño $NS \times 1$.
2. Obtención de la matriz de conectividad “ A_{12} ”. Dicha matriz comprende la conectividad que tienen los nodos a través de las tuberías. Esta tiene un tamaño de $NT \times NN$.
3. Obtención de la matriz transpuesta “ A_{21} ” de tamaño $NN \times NT$.
4. Obtención de la matriz topológica “ A_{10} ”. Esta comprende a los nodos con cabeza fija, representándolos con un -1. Tamaño $NT \times NS$.
5. Suposición de los caudales en los tubos de la red para la formación de la matriz “Q” (No importa el caudal que se proponga). Tamaño $NT \times 1$.
6. Formación de la matriz consumo “q”. Esta contiene la demanda en los nodos. Su tamaño es $NN \times 1$.
7. Creación de la matriz identidad “N”. Esta contiene en su diagonal el valor del exponente de la fórmula que se esté utilizando para calcular las pérdidas. Tamaño $NT \times NT$.
8. Obtención de la matriz identidad “I”. Tamaño $NT \times NT$.
9. Cálculo de la matriz de pérdidas “ A_{11} ”. En esta matriz unitaria, el valor de la diagonal es “ αQ ”, siendo $\alpha = hf/Q^2$. Tamaño $NT \times NT$.
10. Obtención de la matriz transpuesta A_{11}' de tamaño $NT \times NT$.

11. Obtención de las nuevas cargas mediante la fórmula:

$$H_{i+1} = -\text{inv}\{A_{21} * \text{inv}[N * A_{11}'] * A_{12}\} * \{[A_{21} * \text{inv}(N * A_{11}')] * [A_{11} * Q + A_{10} * H_0] - [A_{21} * Q - q]\}$$
12. Obtención de los nuevos gastos mediante la fórmula:

$$Q_{i+1} = [I - \text{inv}(N * A_{11}') * A_{11}] * Q - \text{inv}(N * A_{11}') * \{A_{12} * H_{i+1} + A_{10} * H_0\}$$
13. Recalcular matriz unitaria de pérdidas "A₁₁" con los nuevos gastos obtenidos (ver paso 9).
14. Recalcular las nuevas cargas mediante la fórmula descrita en el paso 11.
15. Si se tiene diferencia entre las cargas calculadas en el paso 14, con las obtenidas en el paso 11, se repiten los pasos 12, 13 y 14. En caso de que no haya diferencia, se obtienen las presiones restando las elevaciones de los nodos a los valores de la carga.

Para la explicación del método del gradiente, se tomará en cuenta la red mostrada en la figura V.B.

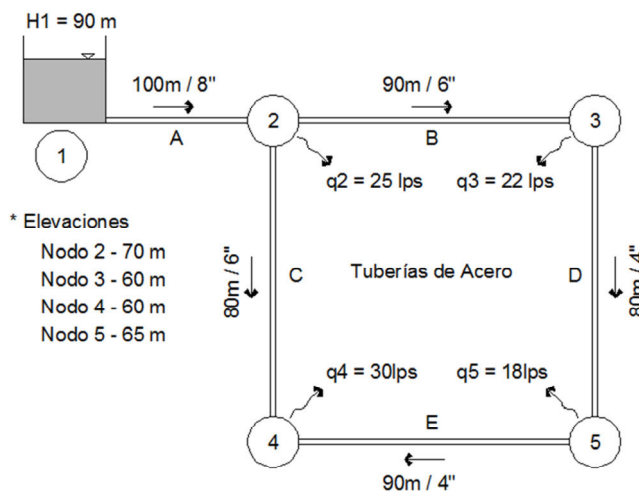


Fig. IV.B.- Red de 5 tubos para planteamiento de ecuaciones

De acuerdo a su geometría, se puede apreciar que:

NT = 5 ; Número de tubos (A, B, C, D, E).

NN = 4 ; Número de nodos sin carga conocida (2, 3, 4, 5).

NS = 1 ; Número de nodos con carga conocida (1).

H₀ = 90 ; Carga conocida fija (Nodo 1)

Ahora bien, para la obtención de la matriz de conectividad A₁₂, se debe expresar la unión entre nodos de la siguiente manera:

$$H_i - hf_{ij} = H_j$$

Siendo H_i la carga del nodo aguas arriba, H_j la carga del nodo aguas abajo y hf_{ij} las pérdidas de carga en el tubo que conecta al nodo i con el nodo j.

De la misma manera, se puede decir que:

$$hf_{ij} - H_i + H_j = 0$$

Aplicando esta expresión a todos los tubos de la red en cuestión, tenemos que:

$$hf_A + H_2 - H_1 = 0 \quad ; \text{ Para tubo A}$$

$$hf_B + H_3 - H_2 = 0 \quad ; \text{ Para tubo B}$$

$$hf_C + H_4 - H_2 = 0 \quad ; \text{ Para tubo C}$$

$$hf_D + H_5 - H_3 = 0 \quad ; \text{ Para tubo D}$$

$$hf_E + H_4 - H_5 = 0 \quad ; \text{ Para tubo E}$$

Anotando dichas condiciones en forma de matriz, tenemos que la matriz de conectividad A_{12} será:

$$A_{12} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{Bmatrix}$$

El tamaño de esta matriz es NTxNN, por tal motivo, la primera fila hace alusión al tubo A, la segunda al tubo B, etc. Las columnas hacen alusión a los nodos; la primera columna se refiere al nodo 2, la segunda columna se refiere al nodo 3, etc. No se incluye al nodo 1 en las columnas porque el nodo 1 se considera en NS (nodos con carga conocida) por tratarse del tanque de almacenamiento. Para la segunda fila (que se refiere al tubo B), la primera columna (que se refiere al nodo 2) tiene un valor de -1 porque es el signo que tiene H_2 en la ecuación del tubo B. Como H_3 tiene signo positivo en la ecuación del tubo B, la segunda columna de la segunda fila de la matriz es +1.

Habiendo obtenido la matriz de conectividad, se obtiene la matriz transpuesta A_{21} de tamaño NNxNT:

$$A_{21} = \begin{Bmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{Bmatrix}$$

La matriz topológica A_{10} comprende a los nodos que tienen una carga fija y se representan con -1:

$$A_{10} = \begin{Bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

En este caso, la matriz A_{10} lleva el valor -1 en la primera fila porque se hace alusión al nodo 1 (tanque de almacenamiento), el cual siempre tiene una carga fija. Los demás nodos se representan con cero por tener carga variable.

Se suponen los caudales para los tubos de la red, estos no se necesitan calcular. Suponiendo un gasto arbitrario de $0.10 \text{ m}^3/\text{s}$ en cada tubo, tenemos que la matriz Q es:

$$Q = \begin{Bmatrix} 0.10 \\ 0.10 \\ 0.10 \\ 0.10 \\ 0.10 \end{Bmatrix}$$

La matriz que contiene la demanda en la red, se denomina como q . El tamaño de esta matriz es $NN \times 1$. Es importante hacer notar que las demandas deben estar expresadas en m^3/s . Convirtiendo las demandas en m^3/s , se obtiene:

$$q = \begin{Bmatrix} 0.025 \\ 0.022 \\ 0.030 \\ 0.018 \end{Bmatrix}$$

Ahora bien, la pérdida por fricción en cada tramo i-j se puede expresar mediante la siguiente fórmula:

$$hf_{ij} = K_{ij} Q_{ij}^n$$

En la que Q hace alusión al gasto, K es una constante que depende de las características del tubo y n es un exponente que depende de la fórmula de pérdidas que se use (Sotelo, 1999).

Para Darcy-Weisbach tenemos que:

$$hf = \frac{fLQ^2}{2gDA^2}$$

Si aislamos las características del tubo como f = factor rugosidad del tubo, L = longitud del tubo, D = diámetro del tubo, A = Área del tubo y las resumimos en una constante K, tenemos que:

$$K = \frac{fL}{2gDA^2}$$

Por lo tanto:

$$hf = KQ^2; \text{ Por lo tanto } n=2$$

De la misma manera, podemos expresar las pérdidas con Hazen-Williams de la siguiente forma:

$$hf = \frac{LQ^{1.851}}{[0.279C_H D^{2.63}]^{1.851}}$$

$$Si \quad K = \frac{L}{[0.279 C_H D^{2.63}]^{1.851}}$$

$$hf = KQ^{1.851}; \text{ Por lo tanto } n=1.851$$

Si se opta por usar la ecuación de Darcy-Weisbach, se tiene un exponente $n=2$. Colocando dicho exponente en la diagonal de una matriz unitaria de tamaño $NT \times NT$ se obtiene:

$$N = \begin{Bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{Bmatrix}$$

Se estructura la matriz identidad "I" con tamaño $NT \times NT$:

$$I = \begin{Bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{Bmatrix}$$

Para formar la matriz de pérdidas A_{11} , se debe calcular αQ para cada tubo de la red en cuestión. El valor del gasto es aquél propuesto en la matriz Q. El valor de α se calcula mediante:

$$\alpha = \frac{hf}{Q^2}$$

Donde:

hf= Pérdidas por fricción

Q = Gasto en el tubo

Las pérdidas por fricción hf se obtienen mediante la fórmula de Darcy-Weisbach:

$$hf = f \frac{L}{D} \left(\frac{V^2}{2g} \right)$$

Donde:

hf= Pérdidas por fricción (m).

f = Factor de fricción.

L = Longitud media del tubo (m).

D = Diámetro (m).

V = Velocidad media (m/s).

g = Gravedad (m/s²).

El factor de fricción f está en función de la rugosidad ϵ y del número de Reynolds Re. Dicho factor de fricción se puede determinar de manera manual mediante el diagrama universal de Moody. Con la finalidad de evitar la introducción de los factores de fricción de manera manual, en este apartado se usa la ecuación de Swamee-Jain (Swamee & Jain, 1976).

$$f = \frac{0.25}{\left(\log_{10} \left[\frac{\epsilon}{3.7D} + \frac{5.74}{Re^{0.90}} \right] \right)^2}$$

Donde:

f= Factor de fricción para cualquier tipo y tamaño de tubo.

ϵ = Rugosidad absoluta del tubo (mm).

D = Diámetro de la tubería (m).

Re = Número de Reynolds.

La rugosidad absoluta de la tubería ϵ se determina con base en el material de fabricación, la frecuencia y alineamiento de las juntas, el tipo de costura o remachado, las incrustaciones y acumulamientos que se presenten y el estado físico en el que se encuentre el conducto (Sotelo, 1999). Para este ejemplo, $\epsilon = 0.075$ mm (Acero soldado de calidad normal nuevo).

El número de Reynolds se obtiene mediante la siguiente fórmula:

$$Re = \frac{VD}{\nu}$$

Donde:

Re = Número de Reynolds.

V= Velocidad media.

D= Diámetro del conducto.

ν = Viscosidad cinemática del fluido

La viscosidad cinemática depende del fluido en cuestión y de la temperatura de este. Para el agua a 20°C, $\nu = 1.007 \times 10^{-6}$ m²/s (Giles, 1994). La velocidad media se determina mediante el gasto y el área del tubo. Esto es:

$$V = \frac{4Q}{\pi D^2}$$

Donde:

V = Velocidad media.

Q = Gasto.

D = Diámetro del tubo.

Resumiendo las características de los tubos se tiene:

Tubo	Longitud (L) m.	Diámetro (D)		Gasto (Q) m ³ /s.
		plg.	m.	
A	100	8	0.2032	0.10
B	90	6	0.1524	0.10
C	80	6	0.1524	0.10
D	80	4	0.1016	0.10
E	90	4	0.1016	0.10

Con las fórmulas antes descritas se realizan los cálculos necesarios hasta la obtención de αQ .

Tubo	Velocidad (V) m/s	Número de Reynolds (Re)	Rugosidad (ϵ) m
A	3.084	626,594.264	0.000075
B	5.482	835,459.019	0.000075
C	5.482	835,459.019	0.000075
D	12.335	1,253,188.528	0.000075
E	12.335	1,253,188.528	0.000075

Tubo	Factor de Fricción (f)	Pérdidas (hf) m	Alfa (α)	αQ
A	0.0167	3.979	397.90	39.79
B	0.0173	15.692	1,569.19	156.92
C	0.0173	13.948	1,394.83	139.48
D	0.0187	113.876	11,387.56	1,138.76
E	0.0187	128.110	12,811.00	1,281.10

Los valores de αQ se substituyen en la diagonal de la matriz unitaria de pérdidas A_{11} , la cual tiene un tamaño NTxNT:

$$A_{11} = \begin{bmatrix} 39.79 & 0 & 0 & 0 & 0 \\ 0 & 156.92 & 0 & 0 & 0 \\ 0 & 0 & 139.48 & 0 & 0 \\ 0 & 0 & 0 & 1,138.76 & 0 \\ 0 & 0 & 0 & 0 & 1,281.10 \end{bmatrix}$$

Se obtiene la matriz transpuesta A_{11}' , la cual resulta ser idéntica a la matriz A_{11} :

$$A_{11}' = \begin{bmatrix} 39.79 & 0 & 0 & 0 & 0 \\ 0 & 156.92 & 0 & 0 & 0 \\ 0 & 0 & 139.48 & 0 & 0 \\ 0 & 0 & 0 & 1,138.76 & 0 \\ 0 & 0 & 0 & 0 & 1,281.10 \end{bmatrix}$$

El siguiente paso consiste en la obtención de las nuevas cargas mediante la fórmula:

$$H_{i+1} = -\text{inv}\{A_{21} * \text{inv}[N * A_{11}'] * A_{12}\} * [\{A_{21} * \text{inv}(N * A_{11}')\} * [A_{11} * Q + A_{10} * H_0] - \{A_{21} * Q - q\}]$$

Para una mejor comprensión de la fórmula, se decidió modificar su presentación a:

$$H_{i+1} = \beta * [\psi * \Omega - \theta]$$

Donde:

$$\beta = -\text{inv}\{A_{21} * \text{inv}[N * A_{11}'] * A_{12}\}$$

$$\psi = A_{21} * \text{inv}[N * A_{11}']$$

$$\Omega = A_{11} * Q + A_{10} * H_0$$

$$\Theta = A_{21} * Q - q$$

Es importante hacer notar que, como se trata de operaciones matriciales, el orden descrito es imperante. También se debe tener presente que A_{11}' no es lo mismo que $\text{inv}(A_{11})$.

De esta manera, el procedimiento para obtener β es el siguiente:

$$\begin{aligned} N * A_{11}' &= \\ &= \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix} * \begin{pmatrix} 39.79 & 0 & 0 & 0 & 0 \\ 0 & 156.92 & 0 & 0 & 0 \\ 0 & 0 & 139.48 & 0 & 0 \\ 0 & 0 & 0 & 1,138.76 & 0 \\ 0 & 0 & 0 & 0 & 1,281.10 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} N * A_{11}' &= \\ &= \begin{pmatrix} 79.60 & 0 & 0 & 0 & 0 \\ 0 & 313.80 & 0 & 0 & 0 \\ 0 & 0 & 279.00 & 0 & 0 \\ 0 & 0 & 0 & 2,277.50 & 0 \\ 0 & 0 & 0 & 0 & 2,562.20 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \text{inv}(N * A_{11}') &= \\ &= \text{inv} \begin{pmatrix} 79.60 & 0 & 0 & 0 & 0 \\ 0 & 313.80 & 0 & 0 & 0 \\ 0 & 0 & 279.00 & 0 & 0 \\ 0 & 0 & 0 & 2,277.50 & 0 \\ 0 & 0 & 0 & 0 & 2,562.20 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & \text{inv}(N^*A_{11}')= \\ & = \begin{pmatrix} 0.0126 & 0 & 0 & 0 & 0 \\ 0 & 0.0032 & 0 & 0 & 0 \\ 0 & 0 & 0.0036 & 0 & 0 \\ 0 & 0 & 0 & 0.0004 & 0 \\ 0 & 0 & 0 & 0 & 0.0004 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & A_{21}*\text{inv}(N^*A_{11}')= \\ & = \begin{pmatrix} 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} * \begin{pmatrix} 0.0126 & 0 & 0 & 0 & 0 \\ 0 & 0.0032 & 0 & 0 & 0 \\ 0 & 0 & 0.0036 & 0 & 0 \\ 0 & 0 & 0 & 0.0004 & 0 \\ 0 & 0 & 0 & 0 & 0.0004 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & A_{21}*\text{inv}(N^*A_{11}')= \\ & = \begin{pmatrix} 0.0126 & -0.0032 & -0.0036 & 0 & 0 \\ 0 & 0.0032 & 0 & -0.0004 & 0 \\ 0 & 0 & 0.0036 & 0 & 0.0004 \\ 0 & 0 & 0 & 0.0004 & -0.0004 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & A_{21}*\text{inv}(N^*A_{11}')*A_{12}= \\ & = \begin{pmatrix} 0.0126 & -0.0032 & -0.0036 & 0 & 0 \\ 0 & 0.0032 & 0 & -0.0004 & 0 \\ 0 & 0 & 0.0036 & 0 & 0.0004 \\ 0 & 0 & 0 & 0.0004 & -0.0004 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & A_{21}*\text{inv}(N^*A_{11}')*A_{12}= \\ & = \begin{pmatrix} 0.0193 & -0.0032 & -0.0036 & 0 \\ -0.0032 & 0.0036 & 0 & -0.0004 \\ -0.0036 & 0 & 0.0040 & -0.0004 \\ 0 & -0.0004 & -0.0004 & 0.0008 \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
& - \text{inv} \{ A_{21} * \text{inv}[N * A_{11}'] * A_{12} \} = \\
& = -\text{inv} \begin{pmatrix} 0.0193 & -0.0032 & -0.0036 & 0 \\ -0.0032 & 0.0036 & 0 & -0.0004 \\ -0.0036 & 0 & 0.0040 & -0.0004 \\ 0 & -0.0004 & -0.0004 & 0.0008 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
\beta & = - \text{inv} \{ A_{21} * \text{inv}[N * A_{11}'] * A_{12} \} \\
& = \begin{pmatrix} -79.60 & -79.60 & -79.60 & -79.60 \\ -79.60 & -375.30 & -95.70 & -243.70 \\ -79.60 & -95.70 & -344.20 & -212.60 \\ -79.60 & -243.70 & -212.60 & -1,434.80 \end{pmatrix}
\end{aligned}$$

Habiendo obtenido el valor de β , se calcula los valores de Ψ , Ω y Θ de la misma manera arriba descrita.

$$\Psi = A_{21} * \text{inv}[N * A_{11}']$$

$$\Psi = \begin{pmatrix} 0.0126 & -0.0032 & -0.0036 & 0 & 0 \\ 0 & 0.0032 & 0 & -0.0004 & 0 \\ 0 & 0 & 0.0036 & 0 & 0.0004 \\ 0 & 0 & 0 & 0.0004 & -0.0004 \end{pmatrix}$$

$$\Omega = A_{11} * Q + A_{10} * H_0$$

$$\Omega = \begin{pmatrix} -86.0210 \\ 15.6919 \\ 13.9483 \\ 113.8756 \\ 128.1100 \end{pmatrix}$$

$$\Theta = A_{21} * Q - q$$

$$\theta = \begin{pmatrix} -0.125 \\ -0.022 \\ 0.170 \\ -0.018 \end{pmatrix}$$

Entonces, las cargas calculadas serán:

$$H_{i+1} = \beta * [\Psi * \Omega - \Theta]$$

$$H_{i+1} = \begin{Bmatrix} 86.4189 \\ 78.0870 \\ 102.1940 \\ 67.7281 \end{Bmatrix}$$

La matriz H_{i+1} contiene la carga en los nodos 2 al 5 de la red, nuevamente el nodo 1 no figura por tener carga constante y considerarse en H_0 .

Teniendo las cargas en los nodos, se procede a calcular los gastos en las tuberías mediante la fórmula:

$$Q_{i+1} = [I - \text{inv}(N * A_{11}') * A_{11}] * Q - \text{inv}(N * A_{11}') * \{A_{12} * H_{i+1} + A_{10} * H_0\}$$

Simplificando la fórmula para una fácil comprensión, se tiene que:

$$Q_{i+1} = \rho - \eta * \delta$$

Donde:

$$\rho = [I - \text{inv}(N * A_{11}') * A_{11}] * Q$$

$$\eta = \text{inv}(N * A_{11}')$$

$$\delta = A_{12} * H_{i+1} + A_{10} * H_0$$

Obteniendo ρ , η , δ :

$$\rho = [I - \text{inv}(N * A_{11}') * A_{11}] * Q$$

$$\rho = \begin{pmatrix} 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \end{pmatrix}$$

$$\eta = \text{inv}(N * A_{11}')$$

$$\eta = \begin{pmatrix} 0.0126 & 0 & 0 & 0 & 0 \\ 0 & 0.0032 & 0 & 0 & 0 \\ 0 & 0 & 0.0036 & 0 & 0 \\ 0 & 0 & 0 & 0.0004 & 0 \\ 0 & 0 & 0 & 0 & 0.0004 \end{pmatrix}$$

$$\delta = A_{12} * H_{i+1} + A_{10} * H_0$$

$$\delta = \begin{pmatrix} -3.5811 \\ -8.3319 \\ 15.7751 \\ -10.3589 \\ 34.4658 \end{pmatrix}$$

Entonces, el gasto en los tubos será:

$$Q_{i+1} = \rho - \eta * \delta$$

$$Q_{i+1} = \begin{pmatrix} 0.0950 \\ 0.0765 \\ -0.0065 \\ 0.0545 \\ 0.0365 \end{pmatrix}$$

Habiendo calculado los nuevos gastos en los tubos, se finaliza la primera iteración del método del gradiente. La segunda iteración comienza mediante la obtención de una nueva matriz de pérdidas A_{11} .

Tubo	Longitud (L) m.	Diámetro (D)		Gasto (Q) m ³ /s
		plg.	m.	
A	100	8	0.2032	0.0950
B	90	6	0.1524	0.0765
C	80	6	0.1524	-0.0065
D	80	4	0.1016	0.0545
E	90	4	0.1016	0.0365

Tubo	Velocidad (V) m/s	Número de Reynolds (Re)	Rugosidad (ε) M
A	2.929	595,264.551	0.000075
B	4.196	639,126.149	0.000075
C	-0.359	54,304.836	0.000075
D	6.728	682,987.748	0.000075
E	4.508	457,413.813	0.000075

Tubo	Factor de Fricción (f)	Pérdidas (hf) m	Alfa (α)	αQ
A	0.0167	3.601	398.99	37.90
B	0.0175	9.290	1,585.39	121.36
C	0.0223	-0.077	1,791.98	11.72
D	0.0189	34.373	11,552.17	630.14
E	0.0192	17.605	13,180.21	481.69

Por lo tanto, la matriz A_{11} queda de la siguiente forma:

$$A_{11} = \begin{pmatrix} 37.90 & 0 & 0 & 0 & 0 \\ 0 & 121.36 & 0 & 0 & 0 \\ 0 & 0 & 11.72 & 0 & 0 \\ 0 & 0 & 0 & 630.14 & 0 \\ 0 & 0 & 0 & 0 & 481.69 \end{pmatrix}$$

Se calculan las nuevas cargas H_{i+1} . A partir de esta iteración se elimina el término Θ , esto es:

$$H_{i+1} = \beta * [\psi * \Omega]$$

Donde:

$$\beta = - \text{inv} \{ A_{21} * \text{inv}[N * A_{11}'] * A_{12} \}$$

$$\Psi = A_{21} * \text{inv}[N * A_{11}']$$

$$\Omega = A_{11} * Q + A_{10} * H_0$$

$$H_{i+1} = \begin{pmatrix} 86.3991 \\ 83.0895 \\ 85.8983 \\ 79.7674 \end{pmatrix}$$

Se calculan los nuevos gastos.

$$Q_{i+1} = \rho - \eta * \delta$$

$$Q_{i+1} = \begin{pmatrix} 0.0950 \\ 0.0519 \\ 0.0181 \\ 0.0299 \\ 0.0119 \end{pmatrix}$$

Hasta aquí llega la iteración 2. Se observa si el vector H_{i+1} sufrió cambios entre la iteración pasada y esta iteración. De ser así, se vuelve a iterar. Este proceso se repetirá n número de veces hasta que el vector H_{i+1} no presente cambio alguno (o que el cambio obtenido sea insignificante), sólo entonces se tendrá la solución.

A manera de resumen se muestra la siguiente tabla con los valores de los gastos y cargas que se obtuvieron en el proceso realizado para encontrar el resultado de este problema.

Iteración H_{i+1}	Gasto en Tubería (m^3/s)				
	A	B	C	D	E
1	0.095	0.0765	-0.0065	0.0545	0.0365
2	0.095	0.0519	0.0181	0.0299	0.0119
3	0.095	0.039	0.031	0.017	-0.001
4	0.095	0.0322	0.0378	0.0102	-0.0078
5	0.095	0.0321	0.0379	0.0101	-0.0079
6	0.095	0.0321	0.0379	0.0101	-0.0079
7	0.095	0.0321	0.0379	0.0101	-0.0079

Iteración H_{i+1}	Carga en Nodo (m)			
	2	3	4	5
1	86.4189	78.087	102.194	67.7281
2	86.3991	83.0895	85.8983	79.7674
3	86.3991	84.218	85.1565	82.8021
4	86.3991	84.7639	84.3474	84.0339
5	86.3991	84.6803	84.2961	83.3792
6	86.3991	84.6804	84.2959	83.379
7	86.3991	84.6804	84.2959	83.379

Como se puede apreciar en la tabla anterior, después de 7 iteraciones no existen cambios en las cargas de los nodos, esto indica que ahí deben detenerse las iteraciones. Teniendo las cargas en los nodos podemos determinar la presión mediante la fórmula:

$$P_i = H_i - E_i$$

Donde:

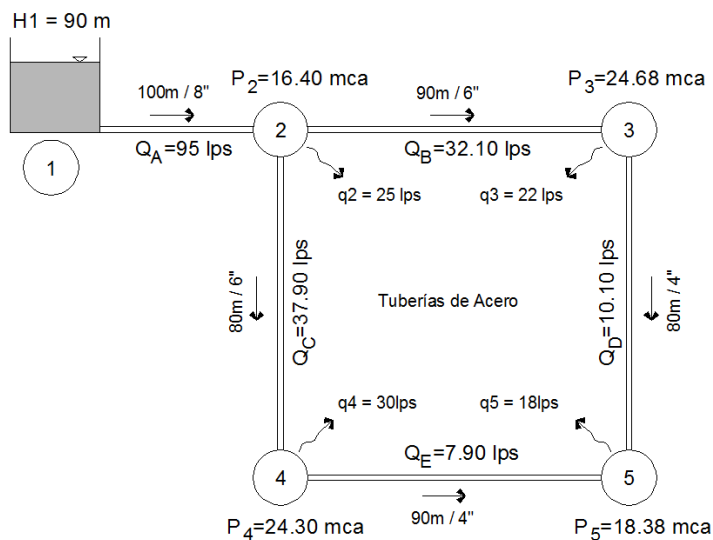
P_i = Presión en el nodo i .

H_i = Carga en el nodo i .

E_i = Elevación en el nodo i .

	Nodo de la Red			
	2	3	4	5
Carga (m)	86.3991	84.6804	84.2959	83.3790
Elevación (m)	70.0000	60.0000	60.0000	65.0000
Presión (m)	16.3991	24.6804	24.2959	18.3790

De esta manera, la solución al problema es la siguiente:



Las ventajas del método del gradiente son su rápida convergencia, menor número de iteraciones para obtener el resultado, no requiere la definición de circuitos y el uso de las matrices en el proceso reduce el tiempo de cómputo (Saldarriaga, 1998).

CAPÍTULO V

>>> PROCEDIMIENTO DE DETECCIÓN PROPUESTO

*“Sólo comprendemos aquellas preguntas que
podemos responder....”
-Friedrich Nietzsche.*

V.1) Fundamentación.

La ley de conservación de la energía indica que la energía no se pierde, aunque si puede transformarse en otros tipos de esta. La energía de un fluido se compone por 3 partes principales: la energía cinética, la energía potencial y la energía de presión. Al fluir, este tipo de energías irán cambiando con las condiciones imperantes, pero su suma siempre será la misma. Esto se establece en el teorema de Bernoulli:

$$Z + \frac{P}{\gamma} + \frac{V^2}{2g} = Cte.$$

Donde:

Z = Posición o carga de altura.

P = Presión.

γ = Peso específico del fluido.

V = Velocidad del fluido.

g = Constante gravitacional.

Ahora bien, el gasto se puede definir mediante la siguiente fórmula:

$$Q = A(V)$$

Donde:

Q = Gasto

A = Área hidráulica de la tubería

V = Velocidad de Flujo.

Entonces el teorema de Bernoulli se puede expresar de la siguiente manera:

$$Z + \frac{P}{\gamma} + \frac{Q^2}{2gA^2} = Cte.$$

Analizando la expresión anterior se puede observar que, en un tramo de tubería en una red de distribución, Z es un valor fijo, puesto que la tubería no cambia su posición una vez que esta ha sido colocada en la obra. El peso específico del agua γ y la constante gravitacional g siempre tendrán un mismo valor. Obviamente, el área del tubo es un valor constante, pues la tubería no puede variar su diámetro.

Entonces, la única forma de hacer variar al gasto en un tramo de una red se logra mediante la variación de la presión. Se concluye que el gasto varía en función de la presión:

$$P = f(Q)$$

Al escalar esto a una red de distribución se puede decir que los gastos de demanda dependerán de las presiones imperantes en los nodos. Visto de otra manera, para un arreglo de gastos determinados, se tendrá un arreglo determinado de presiones.

Cuando se diseña una red de distribución se hace con base en las demandas de la población a servir, las cuales se abastecen mediante los nodos de dicha red. En una red ideal, el organismo operador suministrará el volumen de demanda y este será recibido cabalmente por los usuarios.

En la fig. V.A. se puede apreciar que el gasto de entrada a la red conducido por la tubería A es igual a la suma total de la demanda de los usuarios en los nodos.

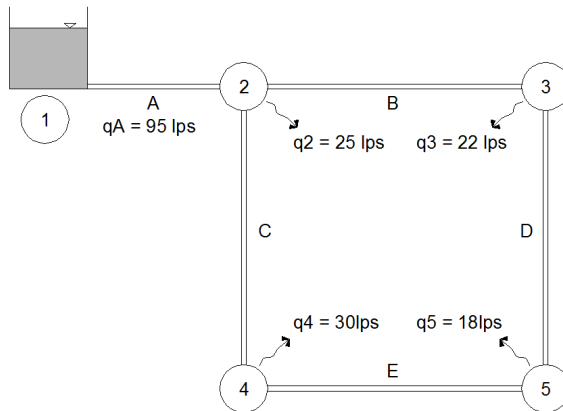


Fig. V.A.- Red con gasto de entrada igual a demanda en los nodos

Con el paso del tiempo, por la mala administración de las presiones en la red y otros factores antes mencionados, se presenta la aparición de fugas. Ahora, el gasto de entrada a la red no será igual a la suma total de las demandas de los usuarios.

Se puede apreciar que la red en la fig.V.B. es la misma que la mostrada en la fig.V.A., con la excepción de contar con dos tomas clandestinas de 2 lps y una fuga de 1 lps. En este último escenario, el organismo operador debe suministrar 100 lps para abastecer a la misma población puesto que, si abastece solamente 95 lps (como en la fig. V.A.), los

usuarios no recibirían suficiente agua por la existencia de la fuga y las tomas clandestinas.

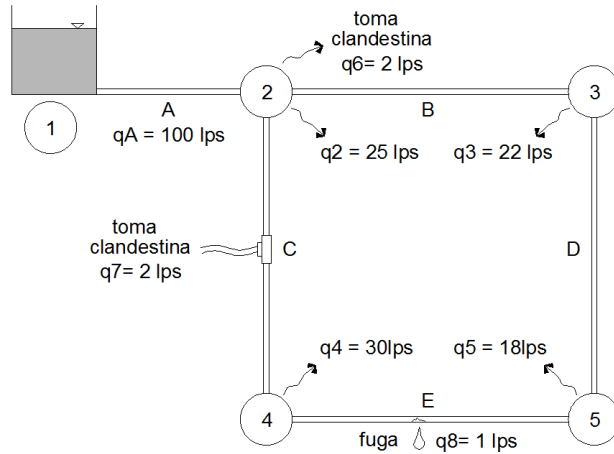


Fig. V.B.- Red con gasto de entrada mayor a la demanda en los nodos

A pesar de que el organismo operador tal vez no sabe la localización ni magnitud de la fuga y/o de las tomas clandestinas, si sabe que tiene que suministrar más agua de la que facturan los usuarios, entonces se puede determinar la existencia de un problema.

Ante un escenario como el presentado en la fig. V.B., los gastos de demanda en los nodos que se tienen registrados son erróneos en unos casos y correctos en otros. Esta incertidumbre hace que los gastos de demanda sean un dato de baja confiabilidad.

El organismo operador debe contar con registradores de presión en los nodos de la red con la finalidad de tener una adecuada administración de presiones y evitar sobrepresiones que aumenten o incentiven la aparición de nuevas fugas.

Como se mencionó anteriormente los gastos dependen de la presión disponible y, la presión registrada en la red está relacionada

directamente a los gastos existentes. Entonces, si se tiene el registro de presiones medidas físicamente en los nodos de la red, se tendrán las presiones para los gastos reales (incluidas las tomas clandestinas y fugas).

Nodo	Gastos (lps)		Presiones (mca)	
	Fig. V.A.	Fig. V.B.	Fig. V.A.	Fig. V.B.
2	25	25+2	16.40	16.02
3	22	22	24.68	24.25
4	30	30	24.29	23.75
5	18	18	18.37	17.82

Tabla V.A.- Variación de presiones en una red al variar los gastos.

En la tabla V.A. se puede apreciar como una red con la misma geometría, elevación, carga y diámetros, varía su configuración de presiones al variar la configuración de gastos (por la presencia de tomas clandestinas y fugas).

Debido a que un arreglo determinado de gastos genera un determinado arreglo de presiones, sería posible determinar si un arreglo de presiones corresponde a una configuración determinada de gastos. Con base en esta observación se puede decir que, a partir de las presiones para la figura V.A, se puede determinar si un arreglo de gastos es el imperante en dicha red.

En la tabla V.A., para las presiones de la figura V.B., no se muestran todos los gastos imperantes en la red. Analizando la figura se puede observar que la tabla nunca menciona la toma clandestina de 2 lps ni la fuga de 1 lps (obsérvese que sí se menciona la toma clandestina del nodo 2). Sin embargo, las presiones imperantes en la red sí toman en cuenta dichos gastos no mencionados. Este sería el caso de un ejemplo práctico, el organismo operador no sabe la existencia, ni ubicación de la

fuga, ni de las tomas clandestinas. A pesar de que sabe que algo anda mal porque suministra más agua de la que factura, no sabe a qué se debe este fenómeno. Mediante la toma de presiones en los nodos de la red se pueden obtener las presiones generadas por los gastos reales. Es decir, se podría proponer un arreglo de gastos y el organismo operador podría cotejar si ese arreglo de gastos pertenece a la configuración de presiones real. La dificultad consiste ahora en cómo determinar el arreglo de gastos para saber si pertenece a la configuración imperante de presiones.

V.2) Descripción.

En este subcapítulo se enuncian las condiciones consideradas en el proceso de detección de fugas. También se hace alusión de la conjunción entre la revisión hidráulica realizada por EPANET, la búsqueda de soluciones de la PSO y la orientación proporcionada por la función objetivo.

Como primer paso, se indican las condiciones consideradas en esta metodología de detección de fugas:

1. La red se trabaja en estado permanente.
2. Se asume que el organismo operador cuenta con registradores de presión en todos los nodos de la red.
3. Una fuga se considera como si se tratara de una demanda adicional.
4. Para la ubicación de fugas, todas las tuberías se dividen por mitad y, en ese punto, se genera un nodo virtual.
5. Las fugas puntuales en los nodos se incluyen en los mismos. Las fugas entre nodos se consideran al centro del tramo de tubería comprendido entre dichos nodos, es decir, en el nodo virtual.
6. Se conocen los gastos demandados y el gasto de entrada a la red.

7. Se conocen a la perfección las características de la red: elevación, geometría, diámetros, material, longitudes, etc.

En el cierre del subcapítulo anterior se enunció que se puede saber si una configuración propuesta de gastos pertenece a una configuración real de presiones. Esas presiones se pueden obtener en campo y comprenden a los gastos no identificados por fugas o tomas clandestinas.

A grosso modo, el método de detección de fugas se puede resumir de la siguiente manera:

- Se tiene definida la red hidráulica por completo, por lo tanto es posible introducir las características de la red en EPANET y, a raíz de la configuración de gastos propuesta, obtener las presiones en los nodos.
- Se propone una configuración de gastos en la red. Obteniendo las presiones en los nodos mediante EPANET, se pueden cotejar las presiones calculadas con las presiones reales y ver que tanto se asemejan. Posteriormente se puede proponer otro arreglo de gastos y observar si este arrojó una configuración de presiones más apegada a las presiones reales.
- Si por algún motivo, un arreglo de gastos arrojara una configuración de presiones idénticas a las presiones reales, la configuración de gastos propuesta serán los gastos reales en la red.
- La diferencia entre la configuración de gastos que dio presiones idénticas a las reales, y los gastos de demanda inicialmente considerados, serán las fugas y tomas clandestinas existentes.

La dificultad de este método consiste en proponer la configuración de gastos que arroje las presiones medidas en campo. En este punto se debe tomar en cuenta que la presión y el gasto en una red no varían proporcionalmente. Esto es, si se varía el gasto de un solo nodo en la red, no implica que solamente varíe la presión en ese nodo.

Nodo	Gastos			Presiones		
	A	B	C	A	B	C
2	25	12.5	25	16.40	19.05	17.60
3	22	11	22	24.68	28.60	26.66
4	30	15	30	24.29	28.50	26.44
5	18	9	0	18.37	23.24	21.53

Tabla V.B. - Variación de gastos y presiones en una misa red.

En la tabla V.B. se muestran 4 diferentes escenarios para la red indicada con anterioridad en la fig. V.A. En dicha tabla se puede apreciar que los gastos en el escenario B corresponden a la mitad de los del escenario A, sin embargo las presiones de B son mayores a las presiones en A. En el escenario C se eliminó la demanda en el nodo 5 y las presiones en C son diferentes en todos los nodos con respecto a A. Con esta tabla se pretende evidenciar que, como se dijo antes, el comportamiento gasto-presión no varía linealmente y la variación en un punto genera variaciones en todo el sistema.

Otro factor a considerar es el número de soluciones posibles, pues la ubicación y determinación de la magnitud de una fuga, es un problema de variables continuas. Para el caso mostrado en la fig. V.B. se tiene un volumen no contabilizado de 4 lps. Si se desea aproximar la fuga hasta centésimas de litros por segundo, tenemos que los valores de fuga pueden adoptar desde 0.01 lps hasta 4.00 lps generando 400 posibles alternativas en la magnitud del gasto en un solo punto. Ahora bien, si se

generan los nodos virtuales para la ubicación de la fuga, se tendrán 9 ubicaciones posibles para dichas fugas.

Se debe recordar que tampoco se sabe el número de fugas, por lo que una solución posible sería que todo el volumen de fuga se encuentre en el nodo 4, pero otra solución posible es que se encuentre 1 lps de fuga en el nodo virtual 7, en el nodo 2, en el nodo virtual 9 y en el nodo virtual 6, etc.; el número de posibles soluciones es inmenso.

Es en este punto donde entra la metodología PSO. La PSO permite la generación de soluciones aleatorias con la finalidad de encontrar un máximo o un mínimo de un problema específico. El nivel de eficiencia de la PSO se medirá al observar la rapidez con la que nos brinda una solución y la exactitud de esta. Por lo tanto, el proceso de detección de fuga propuesto se resume de la siguiente manera:

1. Determinar el volumen no contabilizado de agua (volumen total de fuga). Este volumen se podrá calcular con la diferencia entre el agua suministrada a la red y el agua facturada por el organismo operador.

$$Vf = Qe - \sum_{i=0}^n Qnodo_i$$

Donde:

Vf = Volumen total de fuga.

Qe = Gasto de entrada a la red.

Qnodo= Gasto en el nodo i.

2. Asignar un gasto inicial en cada nodo de la red. Dicho gasto no deberá ser menor al gasto de demanda en dicho nodo, ni podrá ser mayor a la suma del gasto de demanda en dicho nodo más el volumen total de fuga.

$$Qnodo_i = Qnodo_i + [rand() * Vf]$$

Donde:

$Q_{nodo i}$ = Gasto en el nodo i .

$rand()$ = Número aleatorio entre 0 y 1.

V_f = Volumen total de fuga.

3. Revisar hidráulicamente la red con la configuración de gastos propuesta con EPANET y obtener las presiones en los nodos.
4. Mediante la función objetivo se determina que tan bueno o malo es dicho resultado. Un ejemplo de una función objetivo, considerando que la PSO buscará un mínimo para este problema, puede ser la siguiente:

$$FA = \sum_{i=0}^n (P_{EPANET} - P_{REALES})^2$$

Donde:

FA = Factor de aptitud.

P_{REALES} = Presiones tomadas en campo.

P_{EPANET} = Presión obtenida por EPANET a partir de la configuración de gastos propuesta.

5. Si el factor de aptitud no es cero y, si no se han alcanzado el número de iteraciones deseadas, se vuelve al paso 2.

En el proceso de detección de fugas se tendrán un número de dimensiones equivalentes al número de nodos en la red (incluyendo a los nodos virtuales). Por no saber la magnitud de las fugas, en este caso se tendrán variables continuas, pues el volumen total de fugas puede encontrarse compuesto por fugas pequeñas en 1, 2, 3 o en todos los nodos y cada una de diferente magnitud.

V.3) Ejemplo Práctico.

Para su completa comprensión, se tomará la red mostrada en la figura V.C. En dicho caso se tiene una red de 4 nodos y un tanque con una carga de 90 m, el volumen de agua facturada es de 95 lps, sin embargo se tienen que suministrar 100 lps para satisfacer la demanda en los nodos. Es importante recordar que los gastos por la fuga y tomas clandestinas no se saben, aquí se muestran para facilitar su comprensión.

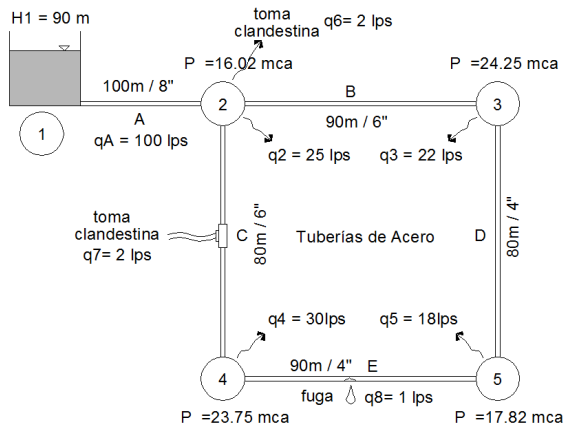


Fig. V.C.- Presiones y fugas existentes en una red

El organismo operador sabe que existe un problema pues tienen que suministrar 100 lps para abastecer 95 lps a los usuarios. Entonces, el volumen de fugas es:

$$V_f = Q_e - \sum_{i=0}^n Q_{nodo_i}$$

$$V_f = 100 - [25 + 22 + 30 + 18]$$

$$V_f = 100 - 95$$

$$V_f = 5 \text{ lps}$$

Habiendo determinado el volumen total de fuga se generan los nodos virtuales a la mitad de los tramos de tuberías. Posteriormente, se registran las presiones en campo (Ver fig.V.D).

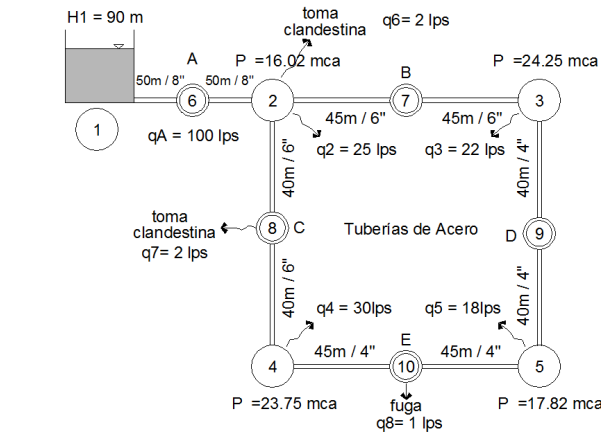


Fig. V.D.- Nodos virtuales en una red

Teniendo las presiones objetivo (presiones medidas en campo) y toda la información física de la red en cuestión, se procede a obtener la primera configuración de gastos.

Nodo	2	3	4	5
Presiones	16.02	24.25	23.75	17.82

Tabla V.C.- Presiones medidas en campo (objetivo).

A partir de este punto inicia la PSO. La primera configuración de gastos la determina la PSO en el paso llamado inicialización de posiciones. Antes de inicializar las posiciones se debe determinar el número de partículas que atacarán el problema, la dimensión del espacio de búsqueda y las dimensiones del espacio de búsqueda.

Por practicidad se elegirán 3 partículas para este problema. La dimensión del espacio de búsqueda equivale al volumen total de fuga calculado con anterioridad. Las dimensiones del problema son 9 (equivalentes al número de nodos).

El primer nodo (tanque) no se toma en cuenta puesto que se considera que la medición del caudal de inyección a la red debe estar después del mismo. Ahora bien, si se tuviera una fuga o toma clandestina después del tanque, esta se considerará al centro del tramo de la tubería entre los nodos 1 y 2, es decir, en el nodo virtual 6.

Se acordó que el rango del espacio de búsqueda sería igual al volumen total de fugas, sin embargo, no se puede limitar a las partículas a buscar hasta el valor de 5 lps puesto que todos los nodos reales tienen demandas mayores a este valor. Por tal motivo, se deberá definir un X_{min} y un X_{max} para cada dimensión del problema. Esto es:

$$X_{min_i} = QDEM_i$$
$$X_{max_i} = QDEM_i + Vf$$

Donde:

X_{min_i} = Mínimo valor del espacio de búsqueda para la dimensión i.

X_{max_i} = Máximo valor del espacio de búsqueda para la dimensión i.

Vf = Volumen total de fugas.

Entonces, para la dimensión 2 (nodo 2):

$$X_{min_2} = 25 \text{ lps.}$$

$$X_{max_2} = 25 \text{ lps} + 5 \text{ lps} = 30 \text{ lps.}$$

Realizando este paso para todos los nodos se obtiene lo siguiente:

Parámetro	Nodo			
	2	3	4	5
Xmin	25	22	30	18
Xmax	30	27	35	23

Tabla V.D.- Posiciones Máximas y Mínimas que pueden adoptar las partículas.

El valor de Xmin y Xmax permite el limitar la búsqueda de la PSO a un rango de valores viables, de nada sirve que la PSO busque soluciones desapegadas a la realidad. Los parámetros de Xmin y Xmax se pueden usar para modalidades de la PSO como Clamping PSO y para la implementación de una función objetivo con penalización (ver siguiente capítulo).

A continuación se definen las velocidades máximas y mínimas de vuelo para las partículas. Para este ejemplo se tomará una velocidad máxima de vuelo de 1.65 y mínima de -1.65.

Hasta el momento se han definido los siguientes parámetros.

Partículas = 3

Vmin = -1.65

Vmax = 1.65

Xmax y Xmin (Ver tabla V.D.)

Iniciando las posiciones X_{ij} y las velocidades V_{ij} de todas las partículas.

$$X_{ij} = rand() * (X_{\max} - X_{\min}) + X_{\min}$$

$$V_{ij} = rand() * (V_{\max} - V_{\min}) + V_{\min}$$

Se reitera que, como se vio en el capítulo III, $rand()$ es un número aleatorio con distribución normal entre cero y uno. Para la dimensión 1 (nodo 2) de la partícula 1 (primera propuesta de gastos para la red):

$$X_{11} = \text{rand}() * (30-25) + 25; \quad \text{Si } \text{rand}() = 0.15; \quad X_{11} = 25.74$$

Poniendo un ejemplo en un nodo virtual, para la dimensión 7 (nodo 8) de la partícula 1 se tiene:

$$X_{17} = \text{rand}() * (5-0) + 0; \quad \text{Si } \text{rand}() = 0.48; \quad X_{17} = 2.39$$

Ahora bien, la velocidad se inicializa de manera similar. Para determinar el signo de la velocidad se asignó la siguiente condición:

$$\text{Si } \text{rand}() < 0.50 \rightarrow V(-)$$

$$\text{Si } \text{rand}() > 0.50 \rightarrow V(+)$$

Entonces, para la dimensión 1 (nodo2) de la partícula 1 se tiene que:

$$V_{11} = \text{rand}() * [4 - (-4)] + (-4); \quad \text{Si } \text{rand}() = 0.64; \quad V_{11} = 1.12$$

Para la dimensión 7 (nodo 8) de la partícula 1 se tiene que:

$$V_{17} = \text{rand}() * [4 - (-4)] + (-4); \quad \text{Si } \text{rand}() = 0.43; \quad V_{17} = -0.56$$

Este proceso se repite para todas las dimensiones de todas las partículas.

n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	X_{n6}	X_{n7}	X_{n8}	X_{n9}
Part. 1	25.74	24.05	31.94	19.84	1.05	3.67	2.39	0.25	2.01
Part. 2	29.88	23.26	34.00	22.07	2.31	4.19	1.13	1.98	0.28
Part. 3	29.37	24.21	31.45	22.29	0.75	2.10	4.64	3.23	1.11

Tabla V.E.- Posiciones en todas las dimensiones de las partículas.

Las velocidades son las siguientes:

n	V _{n1}	V _{n2}	V _{n3}	V _{n4}	V _{n5}	V _{n6}
Partícula 1	1.12	0.50	-1.55	0.95	-1.25	1.30
Partícula 2	-0.79	0.67	0.14	1.37	1.40	0.92
Partícula 3	-0.22	-0.66	0.80	-0.76	0.31	-0.90

n	V _{n7}	V _{n8}	V _{n9}
Partícula 1	-0.56	-0.69	-0.84
Partícula 2	-0.59	-0.84	1.34
Partícula 3	0.33	1.06	-1.12

Tabla V.F. - Velocidades en todas las dimensiones de las partículas.

Como se puede apreciar, ninguna posición de las partículas se encuentra fuera del rango [Xmin, Xmax]. También se aprecia que ninguna velocidad se sale de los límites [Vmin, Vmax].

Habiendo determinado la posición inicial para cada dimensión de cada partícula, se obtienen las presiones en los nodos mediante el programa EPANET. Para saber los detalles en la evaluación, el formato que se utiliza y demás información, refiérase al siguiente subcapítulo.

Evaluando las 3 redes con los gastos propuestos mediante el software EPANET, se obtienen las siguientes presiones:

	Nodo de la Red (Presiones) – mca								
	2	3	4	5	6	7	8	9	10
Part.1	15.17	22.83	22.42	16.13	17.56	18.89	18.72	19.47	19.15
Part.2	14.51	21.99	21.54	14.96	17.20	18.12	17.99	18.33	18.23
Part.3	14.43	21.79	21.41	14.60	17.20	18.05	17.77	17.94	17.92

Tabla V.G. - Presiones en los nodos de la red propuesta por cada partícula.

Los nodos 6 al 10 son nodos virtuales creados para ubicar una fuga en caso de que exista. Esto es, si el nodo virtual existiera, la tabla V.G. muestra las presiones que se tendrían en dichos nodos.

Se procede a evaluar el factor de aptitud para cada partícula. En este ejercicio se buscará un mínimo, esto es, cuando el factor de aptitud sea igual a cero, se habrá encontrado el resultado exacto. Para este ejemplo se usará la siguiente función objetivo con penalización:

$$FA = \frac{1}{n} \sum_{i=1}^n (P_{TEOi} - P_{PSOi})^2 + 9 \times 10^5 \sum_{i=1}^n (PENVOLi + PENDEMi)$$

Donde:

FA = Factor de aptitud.

n = Número de nodos evaluados.

P_{TEOi} = Presión teórica (Objetivo) en el nodo i.

P_{PSOi} = Presión obtenida con EPANET con base a los gastos propuestos por la PSO.

$PENVOLi$ = Penalización por volumen para la dimensión i.

$PENDEMi$ = Penalización por demanda para la dimensión i.

$$Si \quad X_{ij} > (Q_{DEMj} + Vf) \rightarrow PENVOL = X_{ij} - Q_{DEMj} - Vf$$

$$Si \quad X_{ij} < Q_{DEMj} \rightarrow PENDEM = Q_{DEMj} - X_{ij}$$

Donde:

X_{ij} = Posición (gasto) en la dimensión j de la partícula i.

Q_{DEMj} = Gasto de demanda en la dimensión j (nodo j).

Vf = Volumen total de fugas.

La variable PENVOL sanciona a la partícula si alguna de sus dimensiones propone un gasto mayor a la suma de la demanda en el nodo y el volumen total de fugas. Esto se hace porque, si se concentran todas las

fugas de la red en un solo nodo, el valor de gasto será igual a la suma del gasto de demanda y el volumen total de fugas. Por tal motivo, no es posible encontrar una solución mayor a esta. Entonces, si la PSO propone un gasto mayor a este valor, la función objetivo la sanciona severamente para indicarle que ese resultado no es deseado.

La variable PENDEM sanciona a la partícula si alguna de sus dimensiones propone un gasto menor a la demanda en el nodo. No se tiene duda de que la demanda a los usuarios está siendo abastecida, por tal motivo un gasto menor al de demanda no es una solución viable. Si la PSO propone un gasto menor al de demanda, la función objetivo la sanciona fuertemente con la finalidad de indicarle que esa es una solución no deseada.

En esta iteración, los valores de inicialización se asignaron dentro del rango aceptable de gastos [Q_{DEM} , ($Q_{DEM}+V_f$)]. Por tal motivo, PENVOL y PENDEM serán cero en todas las dimensiones de las partículas.

Solamente se toman las primeras 4 dimensiones (nodo 2 al 5) de cada partícula para el cálculo del factor de aptitud, puesto que no se tienen las presiones objetivo para los nodos virtuales.

Partícula 1	X_{12}	X_{13}	X_{14}	X_{15}
(A) PTEO	16.02	24.25	23.75	17.82
(B) PPSO	15.17	22.83	22.42	16.13
(C) [PTEO - PPSO] ²	0.72	2.02	1.77	2.86
(D) PENVOL	0.00	0.00	0.00	0.00
(E) PENDEM	0.00	0.00	0.00	0.00
(F) SUMA (C+D+E)	0.72	2.02	1.77	2.86
FA (SUMA FILA F/4)	1.84			

Tabla V.G.- Cálculo del Factor de Aptitud.

En la tabla V.G., la primera fila hace alusión a las presiones medidas en campo (presiones objetivo), la segunda fila contiene las presiones

obtenidas mediante EPANET para la configuración de gastos propuesta por la partícula 1 de la PSO. La tercera fila se obtiene calculando el cuadrado de la diferencia entre las presiones objetivo y las obtenidas por EPANET. La fila PENVOL contiene las penalizaciones adicionales si el gasto (posición) fue mayor a la suma de la demanda y el volumen total de fuga. La fila PENDEM contiene la penalización atribuida si el gasto (posición) es menor al gasto de demanda en el nodo. La penúltima fila muestra la suma del cuadrado de la diferencia de presiones más las penalizaciones por volumen y por demanda. Si se obtiene el promedio de dichas sumas, se obtiene el factor de aptitud.

Si se realiza el mismo procedimiento para las demás partículas se obtiene lo siguiente:

Partícula	FA
Part. 1	1.84
Part. 2	5.11
Part. 3	6.11

Tabla V.H.- Factor de Aptitud para todas las partículas (iteración 1).

El siguiente paso consiste en determinar cuál ha sido el mejor resultado encontrado en esta iteración para utilizarlo como influencia social en la siguiente iteración. En este caso, la partícula con el mejor resultado ha sido la partícula 1 puesto que su factor de aptitud ha sido el de menor valor obtenido.

Entonces, la mejor posición (Pg) en esta iteración es:

Part.1	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈	X ₁₉
Pg	25.74	24.05	31.94	19.84	1.05	3.67	2.39	0.25	2.01

Aquí inicia la segunda iteración. El primer paso consiste en obtener las nuevas velocidades de las partículas mediante la fórmula:

$$V_{ij(t+1)} = V_{ij(t)} + \varphi_1[Pi - X_{ij(t)}] + \varphi_2[Pg - X_{ij(t)}]$$

Donde:

$V_{ij(t+1)}$ = Velocidad calculada para la siguiente iteración.

$V_{ij(t)}$ = Velocidad actual.

φ_1 = Confianza individual.

Pi = Mejor posición obtenida por el individuo hasta el momento.

$X_{ij(t)}$ = Posición actual.

φ_2 = Influencia social.

Pg = Mejor posición del grupo obtenida hasta el momento.

$$\varphi_1 = 2 * rand()$$

$$\varphi_2 = 2 * rand()$$

Donde:

φ_1 = Confianza individual.

φ_2 = Influencia social.

rand() = Número aleatorio con distribución normal entre cero y uno.

Se procede a calcular los valores de la influencia individual (φ_1) e influencia social (φ_2) para cada dimensión de cada partícula. Para la dimensión 1 de la partícula 1 se tiene que:

$$\varphi_1 = 2 * rand(); \text{ Si } rand() = 0.16, \text{ entonces } \varphi_1 = 0.32$$

$$\varphi_2 = 2 * rand(); \text{ Si } rand() = 0.64, \text{ entonces } \varphi_2 = 1.29$$

Siguiendo este mismo procedimiento, se obtienen los factores de influencia social y personal para todas las dimensiones de todas las partículas:

	Confianza individual por dimensión (φ_1)								
	2	3	4	5	6	7	8	9	10
Part. 1	0.32	0.43	1.51	1.26	0.02	0.47	1.43	1.39	1.90
Part. 2	0.09	0.40	1.36	0.37	0.52	0.88	1.10	0.40	1.19
Part. 3	0.67	1.23	0.84	0.33	0.17	1.23	0.76	1.77	1.20

Tabla V.I.- Factor de influencia individual para todas las partículas.

	Influencia social por dimensión (φ_2)								
	2	3	4	5	6	7	8	9	10
Part. 1	1.29	1.25	1.42	1.19	0.70	0.50	0.65	0.31	0.30
Part. 2	1.84	1.63	0.21	0.26	0.91	0.85	1.06	1.67	1.95
Part. 3	1.90	0.63	0.89	1.64	0.76	0.18	0.16	0.74	0.59

Tabla V.J.- Factor de influencia social para todas las partículas.

Enseguida se procede a calcular la velocidad. La mejor posición P_g es la obtenida por la partícula 1, la mejor posición individual es la obtenida en la iteración anterior por no tenerse más iteraciones. Para la determinación del mejor resultado P_g , en este ejercicio se tomará el mejor encontrado en cualquier iteración por cualquier partícula.

Calculando la velocidad de vuelo para la dimensión 1 de la partícula 2 se tiene que:

$$V_{ij(t+1)} = V_{ij(t)} + \varphi_1 [P_i - X_{ij(t)}] + \varphi_2 [P_g - X_{ij(t)}]$$

$$V_{21} = -0.79 + 0.09 [29.88 - 29.88] + 1.84 [25.74 - 29.88]$$

$$V_{21} = -8.41$$

Una vez calculada la velocidad, se procede a revisar si esta se encuentra entre los límites $[V_{min}, V_{max}]$:

$$V \max < Vij_{(t+1)} < V \min$$

$$Si \quad Vij_{(t+1)} > V \max \rightarrow Vij_{(t+1)} = V \max$$

$$Si \quad Vij_{(t+1)} < V \min \rightarrow Vij_{(t+1)} = V \min$$

Como $V_{21} = -8.41 < V_{\max} = 1.65$, entonces $V_{21} = -8.41$

Como $V_{21} = -8.41 < V_{\min} = -1.65$, entonces $V_{21} = -1.65$

Para su perfecta comprensión, se explicará el cálculo de la velocidad para otra dimensión de otra partícula.

Calculando la velocidad de la dimensión 5 de la partícula 3 se tiene que:

$$Vij_{(t+1)} = Vij_{(t)} + \phi_1 [Pi - Xij_{(t)}] + \phi_2 [Pg - Xij_{(t)}]$$

$$V_{35} = 0.31 + 0.17 [0.75 - 0.75] + 0.76 [1.05 - 0.75]$$

$$V_{35} = 0.54$$

Revisando si se encuentra entre los límites $[V_{\min}, V_{\max}]$:

Como $V_{35} = 0.54 < V_{\max} = 1.65$, entonces $V_{35} = 0.54$

Como $V_{35} = 0.54 > V_{\min} = -1.65$, entonces $V_{35} = 0.54$

De esta manera, se determinan todas las velocidades para todas las dimensiones de las 3 partículas en cuestión:

Partículas	n	V _{n1}	V _{n2}	V _{n3}	V _{n4}	V _{n5}	V _{n6}	V _{n7}	V _{n8}	V _{n9}
1		1.11	0.50	-1.55	0.95	-1.25	1.30	-0.56	-0.69	-0.84
2		-1.65	1.65	-0.29	0.80	0.25	0.49	0.74	-1.65	1.65
3		-1.65	-0.76	1.24	-1.65	0.54	-0.61	-0.04	-1.14	-0.59

Tabla V.K.- Velocidades calculadas para todas las partículas (iteración 2).

Una vez definidas las velocidades, se calculan las nuevas posiciones de las partículas mediante la fórmula:

$$X_{ij(t+1)} = X_{ij(t)} + V_{ij(t+1)}$$

Donde:

$X_{ij(t+1)}$ = Posición de la partícula en la siguiente iteración.

$X_{ij(t)}$ = Posición actual de la partícula.

$V_{ij(t+1)}$ = Velocidad de la partícula en la siguiente iteración.

Calculando la posición de la dimensión 3 de la partícula 1 se tiene que:

$$X_{13} = 31.94 + (-1.55) = 30.39$$

En este punto se puede apreciar que la PSO trata de dirigirse al óptimo global gracias a la orientación que le brinda la función objetivo. Si se observa nuevamente la figura V.D, se aprecia que el gasto solución para la dimensión 3 (nodo 4) es de 30 lps. En la iteración 1, el gasto en este punto era 31.94 lps, la función objetivo le indica a la PSO que el gasto que se está buscando es inferior a este, por tal motivo, el valor en la iteración 2 se acerca un poco más a la solución.

Las posiciones en todas las dimensiones de todas las partículas se muestran en la siguiente tabla:

Gastos (lps).									
n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	X_{n6}	X_{n7}	X_{n8}	X_{n9}
1	26.85	24.54	30.39	20.80	-0.20	4.98	1.83	-0.44	1.17
2	28.23	24.91	33.70	22.87	2.56	4.68	1.87	0.33	1.93
3	27.72	23.45	32.68	20.64	1.29	1.48	4.60	2.09	0.52

Tabla V.L.- Gastos calculados para todas las partículas (iteración 2).

En la tabla V.L. se aprecian valores de gasto negativos. Obviamente, las partículas con estos datos proporcionarán una solución fuera de la

realidad. Para evitar que la PSO le otorgue soluciones no viables a EPANET, se pueden fijar algunas restricciones en el vuelo como el indicado por la técnica Clamping PSO (Ver siguiente capítulo).

Se revisa que los valores de gasto estén en los límites $[X_{maxij}, X_{minij}]$. En caso de que el gasto X_{ij} rebase X_{maxij} se aplica una penalización por volumen PENVOL.

PENVOL									
N	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	X_{n6}	X_{n7}	X_{n8}	X_{n9}
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla V.M.- PENVOL (iteración 2).

Si el gasto X_{ij} es menor que X_{minij} se aplica una penalización por demanda PENDEM.

PENDEM									
n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	X_{n6}	X_{n7}	X_{n8}	X_{n9}
1	0.00	0.00	0.00	0.00	183375.7	0.00	0.00	398008.1	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla V.N.- PENDEM (iteración 2).

Como se puede apreciar en la tabla V.N., el valor de PENDEM para la dimensión 5 de la partícula 1 es 183,375.70. Este valor se obtuvo de la siguiente manera:

$$PENDEM_{15} = 9 \times 10^5 * [0 - (-0.20)] \approx 183,375.70$$

En este caso, el cálculo de PENDEM no genera un valor exacto por el redondeo de -0.20. En realidad el valor de X_{15} es -0.203750075

Ahora se evalúan dichas configuraciones de gasto con EPANET. Solamente se muestran las presiones en los nodos reales por ser los únicos sujetos a evaluación por la función objetivo. Las presiones obtenidas por EPANET son las siguientes:

Presiones (mca)				
Part.	X_{n1}	X_{n2}	X_{n3}	X_{n4}
1	15.20	22.81	22.65	16.22
2	14.34	21.63	21.21	14.50
3	14.87	22.54	21.95	15.62

Tabla V.O.- PENDEM (iteración 2).

Se obtiene el factor de aptitud para cada partícula. Se detalla el cálculo de esta para la partícula 1, por tener penalizaciones por PENDEM.

Partícula 1	X_{12}	X_{13}	X_{14}	X_{15}	X_{18}
(A) PTEO	16.02	24.25	23.75	17.82	0.00
(B) PPSO	15.20	22.81	22.65	16.22	0.00
(C) [PTEO-PPSO] ²	0.67	2.07	1.21	2.56	0.00
(D) PENVOL	0.00	0.00	0.00	0.00	0.00
(E) PENDEM	0.00	0.00	0.00	183,375.68	398,008.10
(F) SUMA (C+D+E)	0.67	2.07	1.21	183,378.24	398,008.10
FA (SUMA FILA F/4)	145,347.57				

El factor de aptitud calculado para cada partícula se indica en la siguiente tabla.

Partícula	FA
Part. 1	145,347.57
Part. 2	6.79
Part. 3	3.08

Tabla V.H.- Factor de Aptitud para todas las partículas (iteración 2).

Como ya se tienen dos iteraciones, se revisa si en esta iteración obtuvo un mejor factor de aptitud que la anterior.

Partícula	FA	
	Iteración 1	Iteración 2
Part. 1	1.84	145,347.57
Part. 2	5.11	6.79
Part. 3	6.11	3.08

Tabla V.I.- Factor de Aptitud para todas las partículas (iteración 1 & 2).

Como se puede apreciar en la tabla anterior, el factor de aptitud en la iteración 2 aumentó dramáticamente en la partícula 1 por haber propuesto soluciones no viables al problema. La partícula 2 también aumentó su factor de aptitud y, a pesar de que la partícula 3 obtuvo el mejor factor de aptitud de la iteración, no se obtuvo un mejor valor que el de la iteración 1 para la primera partícula. Por lo tanto, la mejor configuración de gastos encontrada hasta el momento es la siguiente:

Part.1	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}
Pg	25.74	24.05	31.94	19.84	1.05	3.67	2.39	0.25	2.01

Tabla V.J.- Mejor global (iteración 2).

Habiendo terminado esta iteración, el procedimiento se repite una y otra vez hasta encontrar un factor de aptitud igual a cero o hasta haber alcanzado un número determinado de iteraciones sin haber encontrado una mejor solución global.

En este ejemplo, se simuló una rutina hasta obtener 800 iteraciones sin mejora. Esta simulación alcanzó las 1,152 iteraciones en 171.97 segundos; esto es, se hizo la revisión hidráulica de 3,456 redes diferentes en menos de 3 minutos. El mejor resultado obtenido fue el siguiente:

Part.1	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈	X ₁₉
Pg	25.30	22.02	30.77	18.32	1.65	0.83	0.95	0.69	0.10

Tabla V.K.- Mejor global de toda la rutina.

Al momento de evaluar dicha configuración de gastos con EPANET se obtienen las siguientes presiones.

Presiones (mca)				
Part.	X _{n1}	X _{n2}	X _{n3}	X _{n4}
1	16.03	24.19	23.75	17.77

Tabla V.L.- Presiones de la mejor configuración de gastos encontrada.

La configuración de presiones arriba descrita arroja un factor de aptitud de 0.001550, el cual es bastante cercano a cero. Cotejando el resultado obtenido contra el resultado real, tenemos lo siguiente:

Nodo	Presiones		Gastos	
	Real	PSO	Real	PSO
2	16.02	16.03	27.00	25.30
3	24.25	24.19	22.00	22.02
4	23.75	23.75	30.00	30.77
5	17.82	17.77	18.00	18.32
6	-----	-----	0.00	1.65
7	-----	-----	0.00	0.83
8	-----	-----	2.00	0.95
9	-----	-----	0.00	0.69
10	-----	-----	1.00	0.10

Tabla V.M.- Resultados de la PSO vs. Solución.

Se concluye este subcapítulo enunciando que, el factor más importante a cuidar en el vuelo de la PSO es la diversidad en la exploración/explotación de las partículas. En este ejemplo, la PSO realizó 1,152 iteraciones, sin embargo no se obtuvieron mejores resultados que los obtenidos en la iteración 352. La PSO convergió muy rápido en un óptimo local y no le fue posible encontrar un mejor resultado a este, debido a la falta de diversidad en sus partículas.

V.4) Uso de la rutina propuesta.

Este subcapítulo explica de manera general los requisitos de la rutina indicada en el anexo A. Se agregan algunas capturas de pantalla de la rutina para facilitar su comprensión.

Para la ejecución de la rutina que permite la detección de fugas en redes de agua potable es imprescindible tener instalado el programa MATLAB® de la empresa MathWorks®. Para el desarrollo de esta tesis se utilizó la versión 7.12.0.635 (R2011a) para 64 bits. El software se puede comprar en el siguiente sitio: <http://www.mathworks.com/products/matlab/>

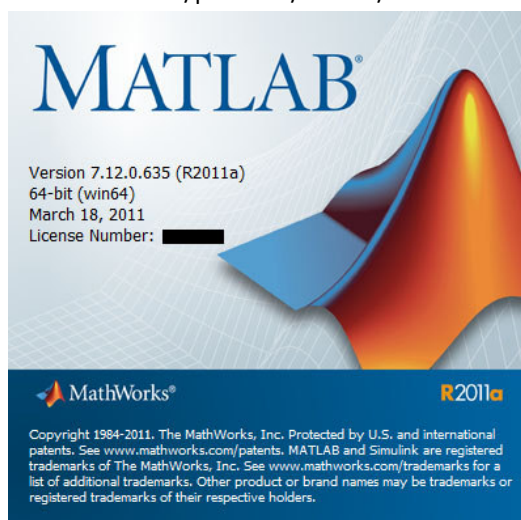


Fig. V.E.- Pantalla arranque del MATLAB.

Para la revisión hidráulica se requiere el software EPANET. En el desarrollo de esta tesis se utilizó la versión 2.0. Este software es gratuito y se puede descargar del siguiente enlace: <http://www.epa.gov/nrmrl/wswrd/dw/epanet.html>

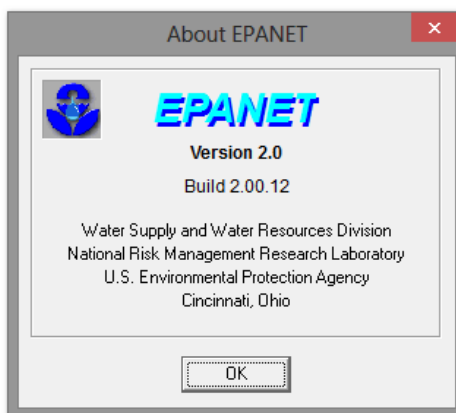


Fig. V.F.- Características del EPANET.

De los archivos que se incluyen en la carpeta de EPANET, solamente se utilizará el archivo ejecutable “epanet2d.exe” de 160 kb. Este archivo deberá copiarse y pegarse a la carpeta C:\Matlab\PSO_JPRG\. En caso de no tenerse estas carpetas, se deberán crear para copiar ahí dicho programa.

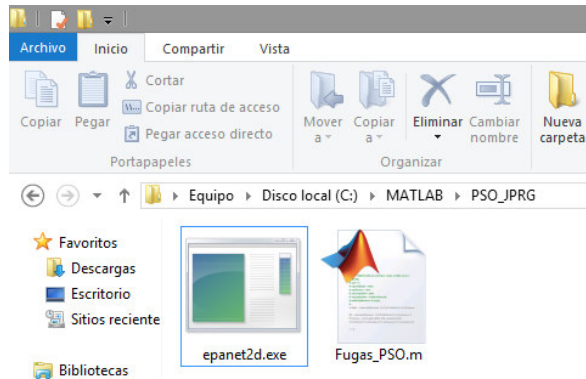


Fig. V.G.- Directorio con rutina y archivo EPANET.

En esta misma carpeta se recomienda tener la rutina para la detección de fugas como se muestra en la figura V.G. (Fugas_PSO para este ejemplo). El nombre de la rutina no tiene importancia y puede ser cambiado en cualquier momento, pero para que esta se efectúe correctamente, el nombre de la rutina no debe contener espacios ni acentos. En caso de que el nombre de la rutina contenga algún carácter no válido, se desplegará un error al correr la rutina como se muestra en la siguiente figura.

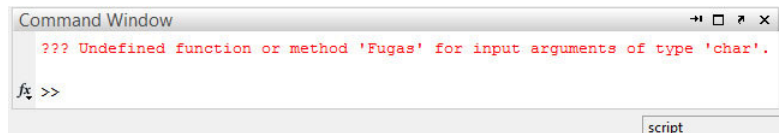


Fig. V.H.- Error en MATLAB por nombre de archivo no apropiado.

Para la obtención de la rutina de MATLAB, basta con copiar y pegar el código incluido en el anexo de esta tesis. El procedimiento es:

- Abrir el programa MATLAB.
- Ir a “File/New/Script” (Fig V.I.)
- En la ventana del Editor pegar el código tal y como se presenta en el anexo A; como se haría con cualquier editor de texto.

- Ir a “File/Save as...” y elegir un nombre con base en las indicaciones antes mencionadas.

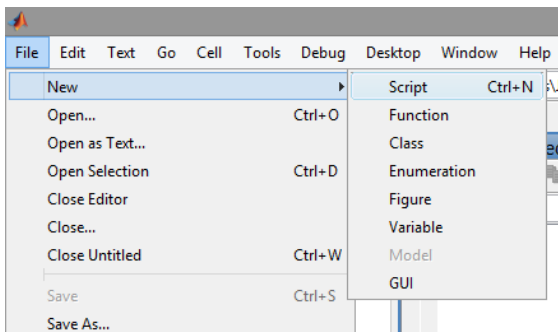


Fig. V.I.-Creando un archivo para grabar la rutina.

Se recomienda ampliamente que la carpeta establecida como actual (current folder) sea C:\Matlab\PSO_JPRG\. La recomendación anterior es con base en la publicación de resultados; la rutina genera resultados de la simulación en formato TXT y los graba en la carpeta actual. De esta forma, se tiene toda la información en un solo lugar.

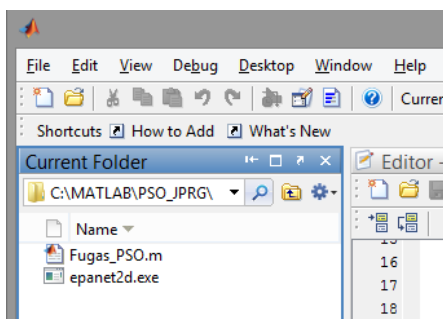


Fig. V.J.-Definiendo carpeta actual (Current Folder).

La rutina en MATLAB contiene las características de vuelo de la PSO, la interacción de esta con EPANET y la función objetivo en cuestión, así como los parámetros de control necesarios para la PSO. Es importante

hacer hincapié en que la rutina no contiene información alguna de la red a analizar; la rutina importa las características hidráulicas y geométricas de la red mediante información contenida en el folder actual.

Se deben generar archivos txt con la información de la red para que la rutina pueda importar las características de la red a analizar. El formato en el que se debe presentar la información es el mismo al usado en la importación de datos en EPANET (Refiérase al manual de usuario de EPANET).

- La información del tanque de abastecimiento (número de nodo, carga) se debe guardar en el archivo “reservoir.txt”
- La información relativa a los nodos (número de nodo, elevación, demanda) se debe incluir en el archivo “junctions.txt”.
- El archivo “pipes.txt” contiene la información de la tubería (número de tubo, nodo inicial, nodo final, longitud, diámetro, rugosidad).
- Las coordenadas de los nodos de la red se especifican en el archivo “coordinates.txt” (número de nodo, coordenada X, coordenada Y). Esta información es de carácter meramente informativo y no influye en el modelo hidráulico, aunque sí es requisito el proporcionar dicha información para el correcto funcionamiento de la rutina, aunque esta sea errónea.
- Las presiones objetivo o presiones físicas medidas en la red se introducen en el archivo “PRESIONES.txt”. Se reitera que estas presiones son las imperantes en la red de distribución cuando se presenta el escenario de fuga (presiones).

Para la completa comprensión de este punto, se muestra el contenido de los archivos txt antes mencionados para el ejemplo expuesto en el subcapítulo anterior (Fig. V.D.).

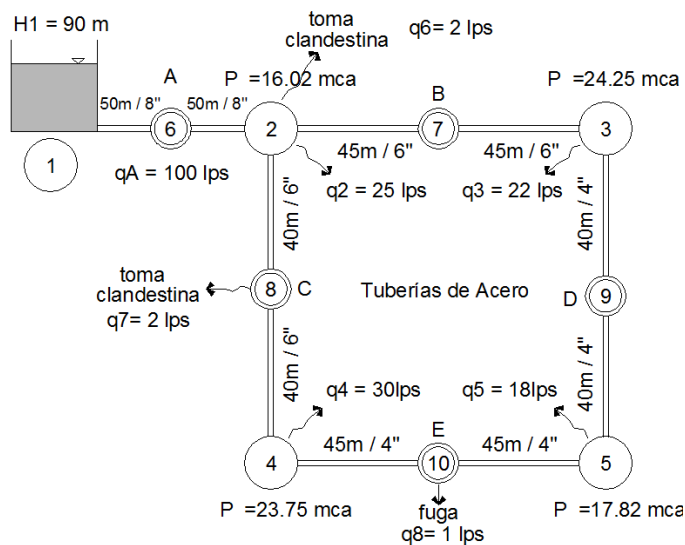


Fig. V.D.- Nodos virtuales en una red

El archivo `reservoir.txt` (Fig. V.K) contiene la información del número de nodo en el que se encuentra el tanque de distribución y la carga del tanque.

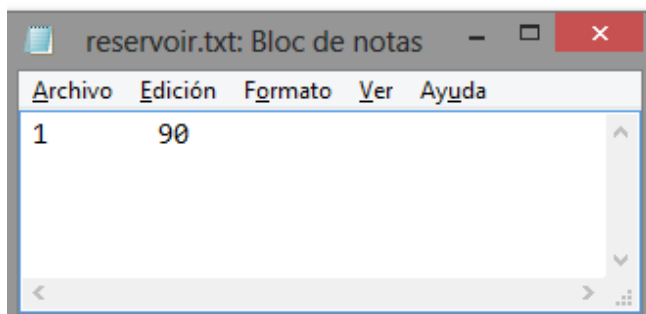
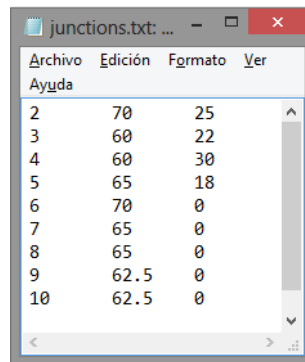


Fig. V.K.- Contenido del archivo `reservoir.txt`

La información contenida en el archivo “junctions.txt” es la siguiente:

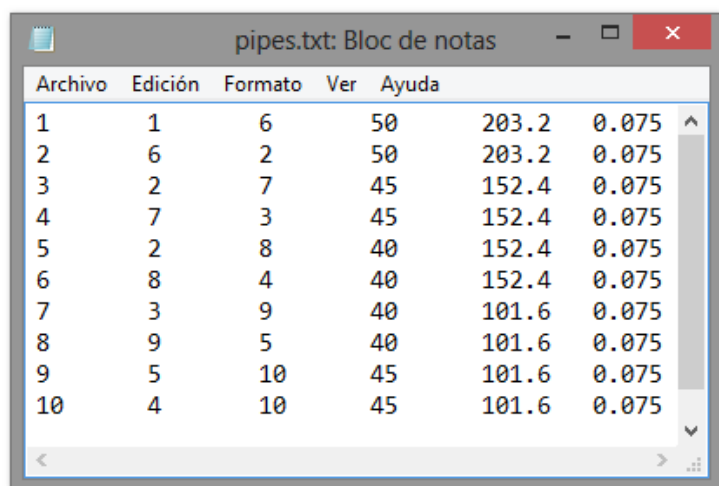


Archivo	Edición	Formato	Ver
Ayuda			
2	70	25	^
3	60	22	
4	60	30	
5	65	18	
6	70	0	
7	65	0	
8	65	0	
9	62.5	0	
10	62.5	0	v

Fig. V.L.-Contenido del archivo junctions.txt

La primera columna contiene el número de nodo (del nodo en cuestión). La segunda columna hace alusión a la elevación que tiene el nodo. La demanda se establece en la tercera columna. Es importante notar que los archivos txt deben contener las características de los nodos virtuales en la red.

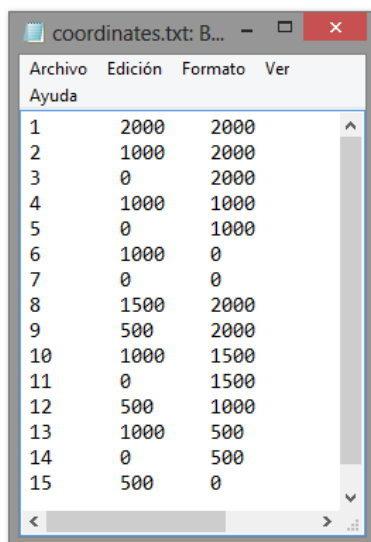
El archivo “pipes.txt” contiene 6 columnas (fig. V.M.). El número del tubo se especifica en la primera. La segunda indica el nodo en el que empieza el tramo de tubería. El nodo en el que termina el tramo de tubo se indica en la tercera. La longitud del tubo se indica en la cuarta columna. Posteriormente, en la quinta, se indica el diámetro del tubo. Finalmente se expresa el factor de fricción en la última columna.



Archivo	Edición	Formato	Ver	Ayuda
1	1	6	50	203.2 0.075
2	6	2	50	203.2 0.075
3	2	7	45	152.4 0.075
4	7	3	45	152.4 0.075
5	2	8	40	152.4 0.075
6	8	4	40	152.4 0.075
7	3	9	40	101.6 0.075
8	9	5	40	101.6 0.075
9	5	10	45	101.6 0.075
10	4	10	45	101.6 0.075

Fig. V.M.-Contenido del archivo pipes.txt

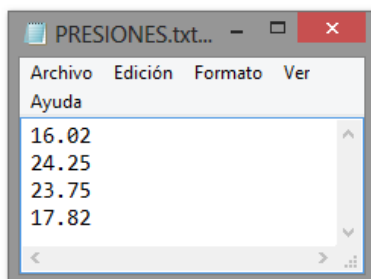
Las coordenadas de los nodos se incluyen en el archivo “coordinates.txt”. Este archivo consta de 3 columnas: la primera indica el nodo en cuestión, la segunda incluye la coordenada X, la tercera indica la coordenada Y. Como se mencionó con anterioridad, los datos de las coordenadas sirven para poder visualizar la red en cuestión. Estos carecen de relevancia alguna en el cálculo hidráulico. De cualquier forma, la captura de estos datos (aunque sea erróneos) es imperante para el funcionamiento de la rutina.



	Edición	Formato	Ver
1	2000	2000	
2	1000	2000	
3	0	2000	
4	1000	1000	
5	0	1000	
6	1000	0	
7	0	0	
8	1500	2000	
9	500	2000	
10	1000	1500	
11	0	1500	
12	500	1000	
13	1000	500	
14	0	500	
15	500	0	

Fig. V.N.-Contenido del archivo coordinates.txt

Las presiones imperantes en la red se almacenan en el archivo “PRESIONES.txt”. Es de vital importancia el reiterar que en este archivo se deben indicar las presiones de la red cuando se tenga la existencia de fugas.



Archivo	Edición	Formato	Ver
16.02			
24.25			
23.75			
17.82			

Fig. V.O.-Contenido del archivo PRESIONES.txt

Si se observa con atención se puede ver que los archivos “coordinates.txt” y “junctions.txt” hacen alusión a todos los nodos de la

red (físicos y virtuales). Para la realización de este ejemplo, el archivo “PRESIONES.txt” hace alusión únicamente a los nodos físicos. La rutina indicada en el apéndice A hace alusión a los nodos físicos y virtuales; esto es, la rutina solicita y evalúa la función objetivo con la presión en los nodos físicos y virtuales (Ver siguiente capítulo).

Otro asunto a tratar son las unidades. A pesar de que EPANET puede trabajar en diferentes unidades, la rutina está programada para trabajar únicamente en las siguientes unidades:

- Carga del tanque: metros (m).
- Elevación del nodo: metros (m).
- Demanda: Litros por segundo (lps).
- Longitud del tubo: metros (m).
- Diámetro del tubo: milímetros (mm).
- Presiones en los nodos: metros columna de agua (mca).

Es importante recalcar que la fórmula utilizada para el cálculo de pérdidas de energía en esta rutina es la de Hazen-Williams; por tal motivo, se deberán indicar los factores de fricción aceptados por esta.

La rutina no realiza conversiones de ningún tipo y asume que las cantidades ingresadas están en las unidades arriba descritas. El no respetar dichas unidades generará un error en la interpretación de los datos proporcionados por la PSO al EPANET. La rutina le indica específicamente al EPANET que los datos están con base a la fórmula de Hazen-Williams (m, mca, lps, mm).

Otro detalle a recalcar, es el ingreso de las características de los nodos virtuales a los archivos txt. El no asignar las características de estos nodos le impedirá al programa el ubicar una fuga en el tubo y, solamente se limitará a ubicar fugas en los nodos (si las hubiera).

Para su fácil manejo, los datos pueden ser capturados en Excel y luego ser guardados como (texto delimitado por tabulaciones).

Habiendo generado los 5 archivos de texto antes mencionados, estos se deben almacenar en la carpeta donde se tiene la rutina y el archivo ejecutable de EPANET (C:\Matlab\PSO_JPRG\).

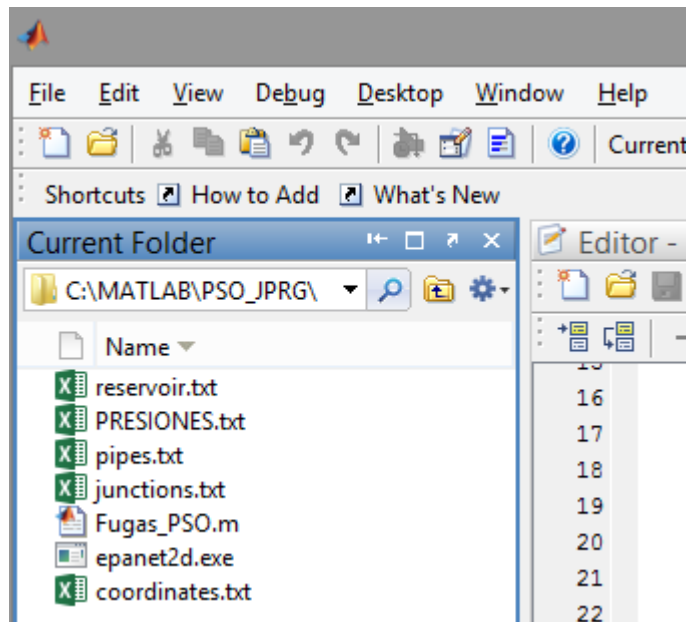


Fig. V.P.-Contenido necesario para correr la rutina.

Para simular la rutina se requiere que esta esté abierta (ver el código en el apartado “Editor”) y presionar el botón “Run” o la tecla F5. Si por algún motivo se desea cancelar la rutina, se puede realizar presionando Ctrl + Z.

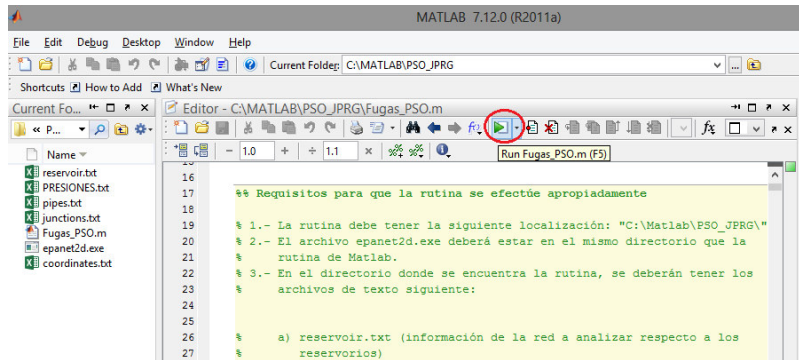


Fig. V.Q.-Puesta en marcha de la rutina.

Al ponerse en marcha, en la ventana command window se muestra la pantalla de bienvenida y se le solicita al usuario el número de partículas que la PSO utilizará.

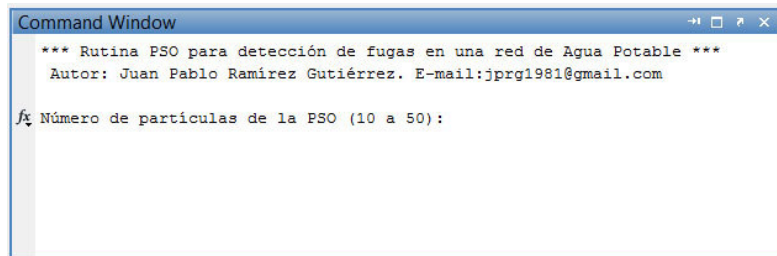


Fig. V.R.-Pantalla de bienvenida.

Después de haber capturado el número de partículas, la rutina muestra el gasto demandado por la red y pide el gasto de entrada a la red. El gasto demandado por la red hace alusión al gasto facturado a los usuarios, es el gasto que el organismo operador tiene plenamente identificado. El gasto de entrada es el gasto que se suministra en realidad para poder abastecer cabalmente el gasto demandado, este comprende a los volúmenes perdidos por fugas y tomas clandestinas.

```
Command Window
*** Rutina PSO para detección de fugas en una red de Agua Potable ***
Autor: Juan Pablo Ramírez Gutiérrez. E-mail:jprg1981@gmail.com

Número de partículas de la PSO (10 a 50): 3

Gasto demandado por la red: 95.00 lps
fx Gasto de entrada a la red (lps)?:
```

Fig. V.S.-Gasto de entrada a la red.

Obviamente, el gasto de entrada a la red deberá ser mayor al gasto demandado por esta. En caso de capturarse un gasto menor, la rutina manda una notificación y vuelve a solicitar el gasto de entrada. Si el gasto de entrada a la red es idéntico al gasto demandado, la rutina manda un aviso de que no se tienen fugas y finaliza.

Habiendo indicado un gasto de entrada mayor al gasto de demanda, se le pregunta al usuario si desea guardar los datos de la rutina en un archivo. Si el usuario indica “S” o “s”, la rutina genera un archivo de texto en el que se almacenan todos los cálculos de todas las iteraciones realizadas por esta. Se darán más detalles sobre el archivo generado más adelante.

```
Command Window
*** Rutina PSO para detección de fugas en una red de Agua Potable ***
Autor: Juan Pablo Ramírez Gutiérrez. E-mail:jprg1981@gmail.com

Número de partículas de la PSO (10 a 50): 3

Gasto demandado por la red: 95.00 lps
Gasto de entrada a la red (lps)?: 100
fx Desea guardar los datos de la rutina en un archivo (S/N)?:
```

Fig. V.T.-Guardar datos de la rutina.

Después, la rutina le pregunta al usuario si desea graficar la exploración (vuelo) de las partículas. Si el usuario indica “S” o “s”, se despliega una ventana que muestra las zonas exploradas del espacio de búsqueda (Fig. V.O.).

La gráfica muestra la posición explorada por cada dimensión de las partículas en una determinada iteración. Como la gráfica desplegada almacena el historial de búsqueda y además muestra todas las dimensiones del problema, el vuelo de la PSO no asemeja al de un enjambre. El graficar la exploración le permite al usuario el analizar problemas de convergencia y de exploración en el espacio de búsqueda.

El realizar la gráfica del vuelo de la PSO hace que la rutina cada vez sea más lenta por cuestiones de memoria. Se recomienda el realizar la gráfica del vuelo en las primeras rutinas realizadas para observar el comportamiento del problema. Al tener identificado cómo se comporta la PSO, se recomienda no visualizar la exploración, para que la rutina se efectúe de manera rápida.

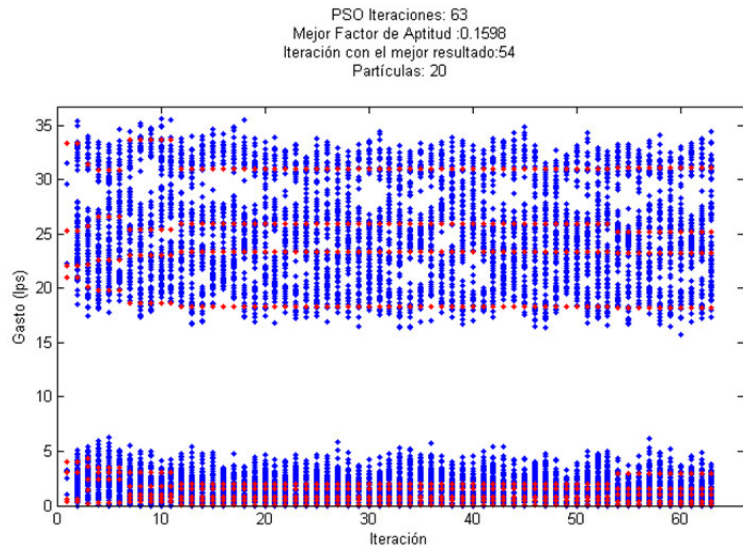


Fig. V.O.-Exploración del espacio de búsqueda.

Finalmente se le pregunta al usuario cuántas veces desea simular la rutina (Fig. V.P.). Una rutina se efectúa por un X número de iteraciones hasta haberse obtenido 800 iteraciones sin encontrar un mejor resultado global o hasta obtener un factor de aptitud igual a cero. Las veces que se

desea simular la rutina hace alusión a que, cuando una rutina concluya, el programa volverá a inicializarla como si se tratara de un problema nuevo, pero respetará los parámetros previamente definidos.

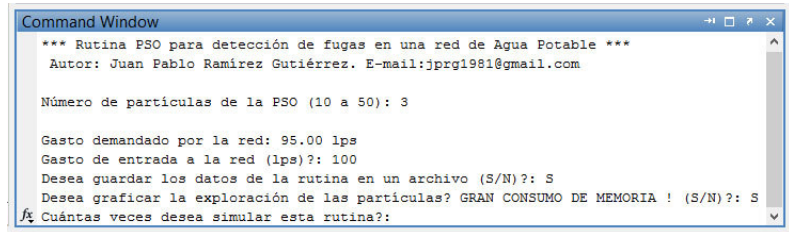


Fig. V.P.-Exploración del espacio de búsqueda.

Para poder determinar el número de redes revisadas, a manera de ejemplo, se expone el siguiente caso: Supóngase que una rutina la simulamos 10 veces con 20 partículas. Suponiendo que la rutina siempre encuentre el mejor global en la iteración 450 tendremos lo siguiente:

- No. de veces que se simulará la rutina: 10
- Partículas: 20
- Iteraciones por rutina: $1250 = 450 + 800$
- Número de redes analizadas: $250,000 = 20 \cdot 1250 \cdot 10$

La intención de simular una rutina X veces consiste en observar un comportamiento. Como el proceso es aleatorio, algunas veces se encontrará rápido un mejor global y, en otras ocasiones, será mucho más tardado.

Después de haber indicado el número de veces que se simulará la rutina, se comienza con la búsqueda de resultados. Esta rutina no requiere de ninguna intervención por parte del usuario, por tal motivo, se puede dejar desatendida.

Durante el cálculo, se despliega en la ventana una serie de líneas que indican que EPANET está evaluando las opciones proporcionadas por la PSO. Las líneas avanzan rápidamente y casi no puede apreciarse su contenido (Fig. V.U.).

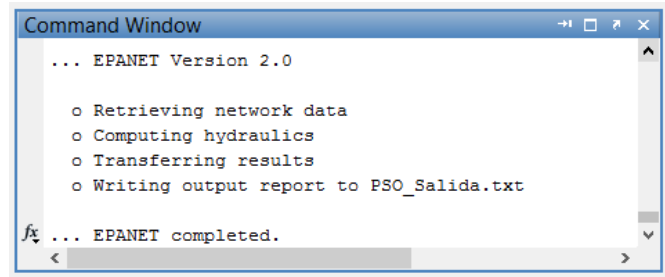


Fig. V.U.-Pantalla durante la simulación.

Durante la ejecución de la rutina se puede observar la generación de otros archivos en la carpeta actual (Fig V.V.).

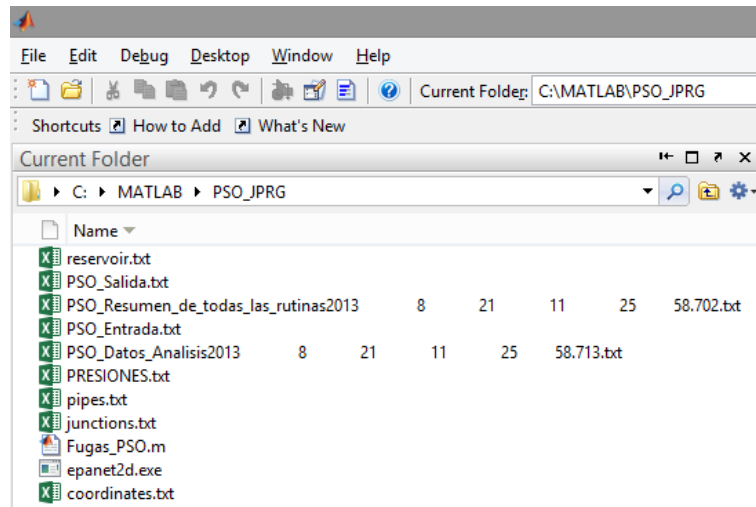
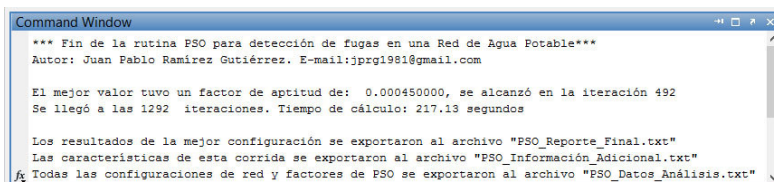


Fig. V.V.-Archivos creados durante rutina.

- PSO_Salida.txt: Archivo temporal que contiene la revisión hecha por EPANET a una propuesta realizada por la PSO.
- PSO_Resumen_de_todas_las_rutinas: Archivo que contiene todos los datos de todas las rutinas corridas durante la simulación.
- PSO_Entrada.txt: Archivo temporal que contiene la configuración de una red propuesta por la PSO. Si este archivo se abre con EPANET, se muestra la geometría y características de la red propuesta.
- PSO_Datos_Analisis: Archivo que contiene todos los cálculos realizados por la PSO en la búsqueda de la solución.

Los archivos “PSO_Resumen_de_todas_las_rutinas”, “PSO_Datos_Analisis” contienen en su nombre la fecha y hora en que arrancó la rutina, de esta forma es imposible confundir una rutina con otra, pues se registra una precisión en el tiempo de arranque con centésimas de segundo.

Al finalizar la rutina, el programa despliega una pantalla de resultados (fig. V.W.) en la que se muestra el mejor factor de aptitud, el tiempo de cálculo en segundos, el número de iteraciones alcanzadas y los nombres de los archivos donde se guardó la información. Si la rutina se simula varias veces, esta pantalla no se aprecia hasta finalizar todas las simulaciones.



```

Command Window
*** Fin de la rutina PSO para detección de fugas en una Red de Agua Potable***
Autor: Juan Pablo Ramírez Gutiérrez. E-mail:jprg1981@gmail.com

El mejor valor tuvo un factor de aptitud de: 0.000450000, se alcanzó en la iteración 492
Se llegó a las 1292 iteraciones. Tiempo de cálculo: 217.13 segundos

Los resultados de la mejor configuración se exportaron al archivo "PSO_Reporte_Final.txt"
Las características de esta corrida se exportaron al archivo "PSO_Información_Adicional.txt"
f Todas las configuraciones de red y factores de PSO se exportaron al archivo "PSO_Datos_Análisis.txt"

```

Fig. V.W.- Pantalla de resultados.

Cada vez que se concluye una rutina, se genera un archivo llamado “PSO_Información_Adicional” (Fig. V.X.). Este archivo contiene datos generales de la rutina efectuada, cada uno de estos archivos contiene fecha y hora para su fácil identificación.

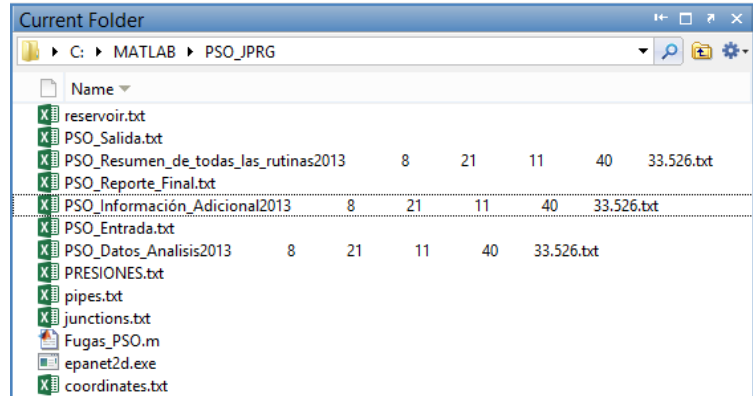


Fig. V.X.- Información Adicional.

La información de las rutinas realizadas se almacena en los siguientes archivos:

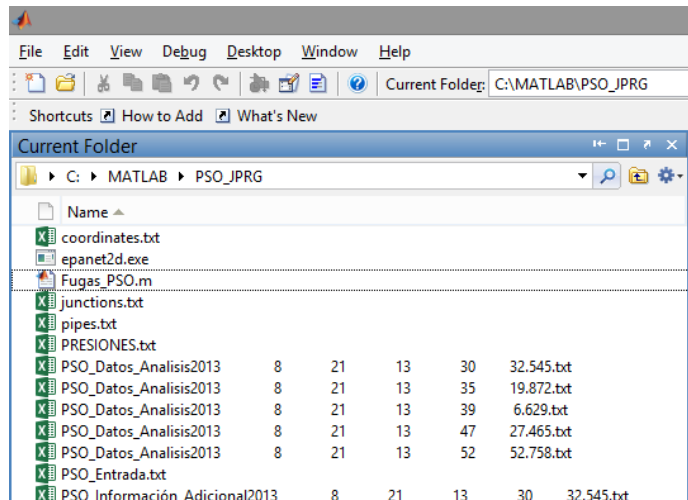


Fig. V.Y.- Archivos con Información de las rutinas simuladas.

- PSO_Resumen_de_todas_las_rutinas

En este archivo se guarda la información de cada rutina concluida. Es decir, si se programa repetir la rutina 5 veces, el resumen contendrá 5 líneas de información (1 por rutina). Entre rutina y rutina programada, el programa actualiza este archivo con la información más relevante de esta.

	A	B	C	D	E	F	G
1	Fecha	Hora	Partículas	Tiempo	FA	Iteraciones	
2	21-Aug-2013	13:30:32	3	287.21	0.01265	1652	
3	21-Aug-2013	13:35:19	3	226.69	0.009075	1580	
4	21-Aug-2013	13:39:06	3	500.74	0.005025	3057	
5	21-Aug-2013	13:47:27	3	325.21	0.0218	2112	
6	21-Aug-2013	13:52:52	3	7635.16	0.00555	2580	

Fig. V.Z.- Archivo Resumen de todas las rutinas (5 en total).

Como se muestra en la figura V.Z., el archivo contiene la información de la fecha y hora de la rutina, las partículas con las que se simuló, el tiempo de ejecución en segundos, el mejor factor de aptitud encontrado y el número de iteraciones alcanzadas.

- PSO_Informacion_Adicional

Este archivo se genera al finalizar cada rutina corrida. Si se programan 5 rutinas, se generarán 5 archivos “PSO_Información_Adicional” con diferentes fechas u horas cada una (Ver fig. V.Y.).

El archivo “PSO_Información_Adicional” contiene información sobre las velocidades permitidas, las velocidades alcanzadas, el volumen de pérdidas, las partículas usadas por la PSO, el tiempo de cálculo, etc. (Ver fig V.AA.).

- PSO_Datos_Análisis

Todos los cálculos realizados por la PSO se almacenan en este archivo. Para su correcta presentación, se recomienda abrir el archivo en Excel y convertir el texto en columnas (delimitado por espacios y tabulaciones). Posteriormente se deberán reemplazar todos los guiones bajos por espacios. Después, se deberá configurar a todas las celdas como número. Habiendo hecho lo anterior, se tendrán los cálculos de manera clara y ordenada.

Es importante mencionar que este archivo se actualiza en tiempo real, esto es, en el momento que se desee se puede interrumpir la rutina y observar en este archivo si los cálculos realizados son los deseados o correctos.

Ejercicio: 21-Aug-2013 13:52:52

Rutina PSO para detección de fugas en una Red de Agua Potable
Autor: Juan Pablo Ramírez Gutiérrez. E-mail:jparg1981@gmail.com

**** Información Adicional de la Rutina ****

**** PSO ****

Número de Partículas: 3
Iteraciones alcanzadas: 2580
Iteraciones sin mejora necesarias para terminar rutina: 801
Mejor valor de aptitud alcanzado: 0.005550000
Iteración en la que se alcanzó el mejor FA: 1780
Velocidad máxima de vuelo permitida: 1.65
Velocidad mínima de vuelo permitida: -1.65
Velocidad máxima de vuelo alcanzada: 1.65
Velocidad mínima de vuelo alcanzada: -1.47
Tiempo de cálculo: 7635.16 segundos

**** RED ****

Número de Nodos en la red: 9
Gasto de entrada: 100.00 lps
Demanda en los nodos: 95.00 lps
Pérdidas: 5.00 lps
Número de redes analizadas: 7740

Fig. V.AA.- Archivo información adicional.

Al inicio de cada archivo se incluyen las fórmulas de la PSO, las demandas en los nodos, las presiones a alcanzar y demás fórmulas para aclarar los cálculos expuestos.

La intención de este archivo es el observar a detalle el cálculo de propuestas obtenidas por la PSO y comprender su procedimiento por completo.

En este archivo se enuncian todas las posiciones, velocidades, factores de aptitud, presiones calculadas, componentes phi, penalizaciones, mejores resultados globales, mejores resultados individuales, etc.

Dependiendo del número de partículas, del número de iteraciones alcanzadas y de la complejidad de la PSO en cuestión, etc., estos archivos pueden alcanzar grandes tamaños por su basta cantidad de información.

B	C	D	E	F	G	H	I	J	K
***** Fórmulas PSO *****									
componente propio= phiuno*(OWNBEST-Xanterior)									
componente social= phidos*(GROUPBEST-Xanterior)									
velocidad= velocidad anterior+componente propio+componente social									
si velocidad<velmin, entonces velocidad=velmin									
si velocidad>velmax, entonces velocidad=velmax									
X= Xanterior+velocidad									
FA= 1/n * SUMATORIA([(Preal-Ppso)^2+PENDEM+PENVOL]									
Si Xicalc<Demanda en nodo, entonces:PENDEM=9x10^5*(Demanda en nodo-Xicalc]									
Si Xicalc>(Demanda en nodo+Vol.Fugas), entonces:PENVOL=9x10^5(Xicalc-Demanda en nodo-Vol.Fugas)									
***** Datos red *****									
1.-Demanda de la red:									
2.50E+01	2.20E+01	3.00E+01	1.80E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
2.-Presiones a alcanzar:									
1.60E+01	2.43E+01	2.38E+01	1.78E+01						
3.-Volumen de fugas: 5.0000									
4.-Velocidad máxima: 1.6500									
5.-Velocidad mínima: -1.6500									
***** Iteración número 1 *****									
X (Valores de gasto)									
2.69E+01	2.56E+01	3.19E+01	1.91E+01	2.46E+00	4.74E+00	1.82E+00	1.57E+00	1.47E+00	

Fig. V.AB.- Archivo Datos Análisis.

En algunas ocasiones, Excel no puede abrir estos archivos de forma completa por su gran tamaño. Para esto se tiene que utilizar un editor de texto de gran tamaño como Large Edit[®] (<http://www.largeedit.com/>) o The V-File Viewer[®] (<http://www.fileviewer.com/>).

CAPÍTULO VI

>>> ANÁLISIS Y MODELADO

“Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa....”

-Mahatma Gandhi

VI.1) Consideraciones.-

Para la obtención de una rutina que fuera apta con esta metodología, se investigaron diferentes adaptaciones hechas a la PSO con la finalidad de la obtención de mejores resultados.

Las modalidades investigadas fueron las siguientes:

- Mejor partícula en la iteración y diferente número de partículas.
- Mejor partícula histórica y diferente número de partículas.
- PSO con formación anillo.
- PSO con inicialización de posiciones mediante Hammersley.
- PSO con inicialización de velocidades mediante Half-Diff.
- PSO con modificación de velocidad máxima de vuelo.
- Clamping PSO
- Inercia dinámica
- Operadores de reinicialización y mutación en PSO con división de trabajo.

El procedimiento para la obtención de la mejor rutina en la detección de fugas fue por medio de prueba y error. Se elige una metodología y se observa su desempeño, posteriormente se modifica con otra técnica; si el desempeño de la rutina modificada es mejor que el de la original, se deja la modificación, en caso contrario, se continúa con la rutina inicial.

El elemento discriminante en la selección entre una y otra metodología fue el factor de aptitud. Una vez que se obtuvo la mejor metodología en la detección de fugas, se utilizan 2 formas adicionales para revisar su eficiencia en la resolución del problema:

a. Similitud con la solución en porcentaje (SSP).

Se compara cada uno de los nodos de los resultados obtenidos contra la solución. En esta modalidad no se revisa la precisión en el pronóstico de fugas, sino el hecho de que se haya indicado una fuga donde en realidad está.

Cuando se proporciona un dato correcto se asigna un punto al nodo, ya sea por el correcto pronóstico de una fuga o por el correcto pronóstico de un nodo sin fuga. Los errores en pronóstico se evalúan con un cero.

Al final del cotejo de todos los nodos, se obtiene el promedio y el porcentaje de similitud de la respuesta con la solución se expresa en forma de porcentaje.

Habiendo determinado la SSP de todos los resultados, se indica cuál fue el mejor SSP, el peor SSP y el promedio del universo de soluciones.

Este método no se obtuvo de alguna bibliografía, fue ideado por el autor para determinar la confiabilidad de los resultados. El objetivo de esta investigación es el determinar la existencia de una fuga incluso sobre la estimación de su volumen, el cual cae a segundo término.

Si la metodología le permite al usuario el identificar las zonas de fuga, aunque no se determine exactamente el volumen, el objetivo de esta tesis se habrá alcanzado.

b. Eficiencia de Nash-Sutcliffe (NSE).

El método Nash-Sutcliffe es usado para la validación de modelos hidrológicos. Este determina que tanto se ajusta un dato obtenido a un resultado determinado, se calcula mediante (Krause et al., 2005):

$$NSE = 1 - \frac{\sum_{i=1}^n (O_i - P_i)^2}{\sum_{i=1}^n (O_i - \bar{O})^2}$$

Donde:

NSE = Eficiencia de Nash-Sutcliffe.

n = Número de valores.

O_i = Valor i del resultado.

P_i = Dato i obtenido.

$\bar{O}$ = Media de los valores que comprenden al resultado

NSE varía de $-\infty$ a 1.00. Si NSE = 1 se tiene un ajuste perfecto entre el dato obtenido y el resultado determinado.

Los valores comprendidos entre cero y uno son considerados comúnmente como niveles aceptables de semejanza (Moriassi et al., 2007).

A pesar de que este problema se formuló como la obtención de un mínimo, los resultados se expresan como 1/FA con la finalidad de hacer más visible un mejor resultado.

Para cada metodología (excepto cuando se evaluaron velocidades o número de partículas), se realizaron 102 simulaciones. Se descartó la rutina con el mejor y peor factor de aptitud encontrado y los otros 100 resultados fueron sujetos de análisis.

El factor de aptitud de todas las rutinas se evaluó con base en la presión obtenida en todos los nodos de la red (tanto virtuales como físicos). En la práctica, esto no es posible puesto que los nodos virtuales no existen.

El procedimiento para obtener las presiones a alcanzar (presiones objetivo) fue el siguiente:

- Se crea un modelo sin fugas; este deberá contener los nodos virtuales también. De dicho modelo se extrae la información para los archivos `reservoir.txt`, `pipes.txt`, `junctions.txt` y `coordinates.txt`.
- Se le agregan las fugas al modelo modificando las demandas de este.
- Se evalúa el modelo mediante EPANET y se obtienen las presiones.
- Las presiones que arroja EPANET serán las presiones objetivo (`PRESIONES.txt`).

Para todas las rutinas simuladas, la condición de paro fue un factor de aptitud igual a cero o la ejecución de 800 iteraciones sin haber obtenido un mejor resultado global.

Cada simulación se realizó de manera paralela en 18 computadoras del laboratorio de Sistemas de Información Geográfica en la Sede Belén de la Universidad de Guanajuato, Campus Guanajuato.

VI.2) Experimentación con diversas metodologías.

Para la experimentación de las diferentes modalidades de la PSO, se creó una red inspirada en la propuesta por Alperovitz & Shamir (1977). El factor de aptitud utilizado en todas las simulaciones fue el anteriormente descrito en el subcapítulo V.3.

$$FA = \frac{1}{n} \sum_{i=1}^n (P_{TEOi} - P_{PSOi})^2 + 9 \times 10^5 \sum_{i=1}^n (PENVOLi + PENDEMi)$$

A pesar de que el factor de aptitud ideado propone esta investigación como la búsqueda de un mínimo, el autor presenta los resultados de manera inversa (1/FA) para que el lector pueda distinguir más fácilmente un buen resultado encontrado.

A no ser que se exprese lo contrario, las fórmulas usadas y las condiciones para el vuelo y posición de la PSO son las expresadas en el subcapítulo V.3.

$$Vij_{(t+1)} = Vij_{(t)} + \varphi_1 [Pi - Xij_{(t)}] + \varphi_2 [Pg - Xij_{(t)}]$$

$$Xij_{(t+1)} = Xij_{(t)} + Vij_{(t+1)}$$

El espacio de búsqueda se estableció para cada dimensión (nodo de la red). El valor mínimo comprendido por el espacio de búsqueda será la demanda por los usuarios en el nodo (sin considerar el volumen de fuga). El máximo valor del espacio de búsqueda equivaldrá a la suma del volumen total de fugas en la red más el volumen de demanda en cada

nodo. De esta manera, el espacio de búsqueda será diferente para cada dimensión del problema.

- Red 1.

Esta red consta de 1 tanque con carga de 210 m y 7 nodos con diferentes elevaciones (fig. VI.A.)

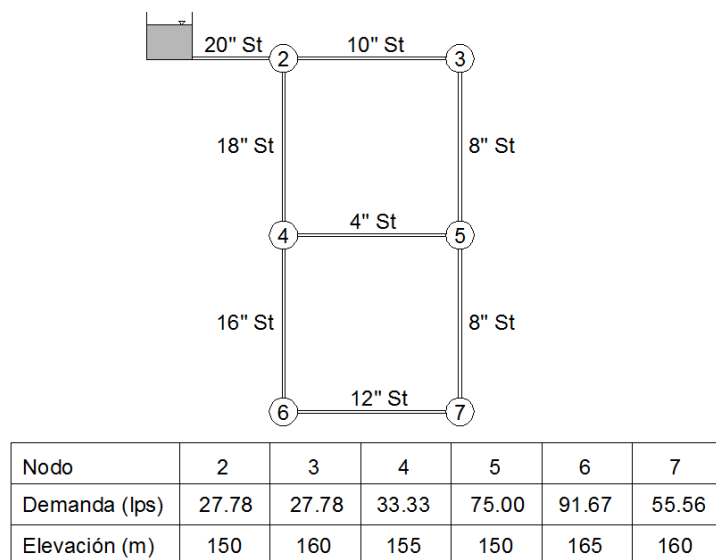
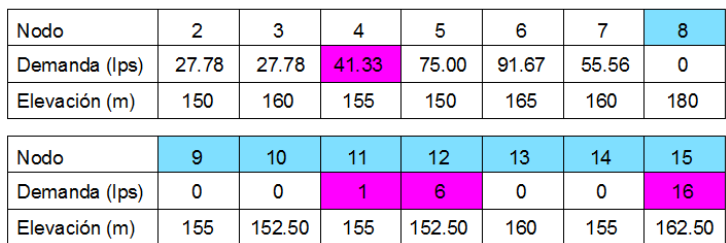


Fig. VI.A.- Red 1 sin fugas.

La longitud de cada uno de los tramos de la tubería es de 1,000 m, la fórmula usada para el cálculo hidráulico fue la de Hazen-Williams, tiene una demanda total de 311.12 lps. Toda la tubería de la red es de acero ($C=130$).

Para la ubicación de fugas, se dividió cada tramo de tubería en dos segmentos, generando 16 tramos de tubería y 15 nodos (7 físicos y 8 virtuales). El gasto de entrada a la red es de 342.12 lps y el volumen de fuga es de 31 lps (Fig. VI.B.).



El espacio de búsqueda definido para cada dimensión en esta red es el siguiente:

Nodo	2	3	4	5	6	7	8
Esp. Busq. Min	27.78	27.78	33.33	75.00	91.67	55.56	0
Esp. Busq. Max	58.78	58.78	64.33	106.00	122.67	86.56	31.00

Nodo	9	10	11	12	13	14	15
Esp. Busq. Min	0	0	0	0	0	0	0
Esp. Busq. Max	31.00	31.00	31.00	31.00	31.00	31.00	31.00

VI.2.1.) Mejor partícula en la iteración (MPI).

Las partículas de la PSO son influenciadas por la experiencia social y personal. La experiencia social es aquella influencia de un grupo sobre un individuo con base en las experiencias obtenidas por algún elemento de ese conjunto. En la PSO, el mejor global (definido por el grupo) influye a las demás partículas.

En esta metodología, se establece cuál partícula de esa iteración obtuvo el mejor resultado del grupo (Yuan et al., 2008). Dicha partícula será la que ejerza la influencia social sobre las demás. En esta modalidad no importa la mejor partícula histórica; es decir, en la iteración actual, la partícula con mejor resultado puede tener un peor factor de aptitud que la mejor partícula en la iteración anterior y esto carecerá de importancia (Fig. VI.C.).

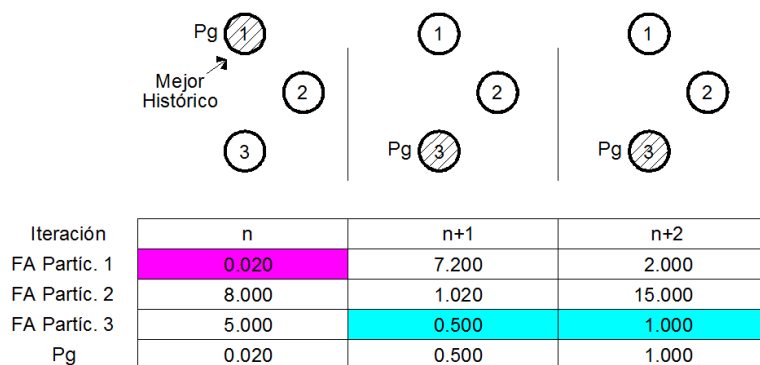


Fig. VI.C.- Mejor partícula por iteración.

Además de determinar el mejor global por iteración, se simuló para varios números de partículas con la finalidad de observar su comportamiento. Los números de partículas simulados fueron: 2, 17,25, 50, 75, 80, 100 y 150 partículas. Se corrieron 52 rutinas para cada número de partículas indicado; de estas, la mejor y peor rutina fueron descartadas y las otras 50 fueron sujetas de estudio.

La elección de los números de partículas simulados se hizo con la intención de identificar el comportamiento de la PSO, siendo 2 el menor número de partículas con interacción social, y 150 partículas el mayor número de partículas simulado.

No se consideró viable la simulación de más de 150 partículas por el tiempo requerido en cada iteración y los recursos computacionales necesarios. El valor de 80 partículas se seleccionó con la finalidad de cotejar su desempeño contra el de 75 partículas; esto se hizo puesto que en otras simulaciones previamente hechas, el mejor desempeño se obtuvo entre 75 y 80 partículas.

Según Maurice Clerc y su Standard PSO (SPSO), se puede determinar el número de partículas a utilizar mediante la siguiente fórmula (Clerc et al., 2010):

$$S = \text{Int}(10 + 2\sqrt{D})$$

Donde:

S= Tamaño del enjambre.

Int= Función que devuelve un número entero.

D= Dimensiones del problema.

Utilizando la fórmula de Clerc en este problema, se obtienen 17 partículas a utilizar para la PSO.

Los parámetros para la PSO en este caso (MPI) fueron los siguientes:

- Inicialización de posiciones y velocidades aleatorias dentro del espacio de búsqueda definido y rango de velocidades establecido.
- Topología de estrella (gbest).

- Pg se asigna a la mejor partícula en la iteración sin recordar el mejor resultado en la rutina (histórico).
- $V_{max} = 0.33 * Vol. \text{ Fugas} = 10.23$
- $V_{min} = -V_{max} = -10.23$
- $\varphi_1 = 2 * rand() , \varphi_2 = 2 * rand()$

El mejor resultado obtenido tuvo un factor de aptitud de 0.4479000 ($1/FA = 2.2326$), este se presentó cuando se usaron 150 partículas. El peor factor de aptitud fue de 29.9411286 ($1/FA = 0.0334$) y se obtuvo cuando se utilizaron 2 partículas. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	39.11	28.16	34.67	77.02	97.94	60.50	1.63
Peor FA	28.39	30.67	46.27	77.28	92.84	57.29	0.29

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	4.40	5.34	0.87	2.54	1.25	0.31	2.34
Peor FA	24.03	11.53	1.17	12.72	11.99	8.56	11.42

Para cada grupo de partículas se obtuvo el factor de aptitud promedio. Los resultados fueron los siguientes:

Partículas	1/FA	Partículas	1/FA
2	0.093721	75	0.453495
17	0.232603	80	0.433805
25	0.260129	100	0.469018
50	0.324296	150	0.665356
PROMEDIO 1/FA		0.366553	

Como se puede apreciar, para 150 partículas se obtuvo el mejor factor de aptitud. Con la finalidad de visualizar mejor los resultados, se muestran en la gráfica VI.D.

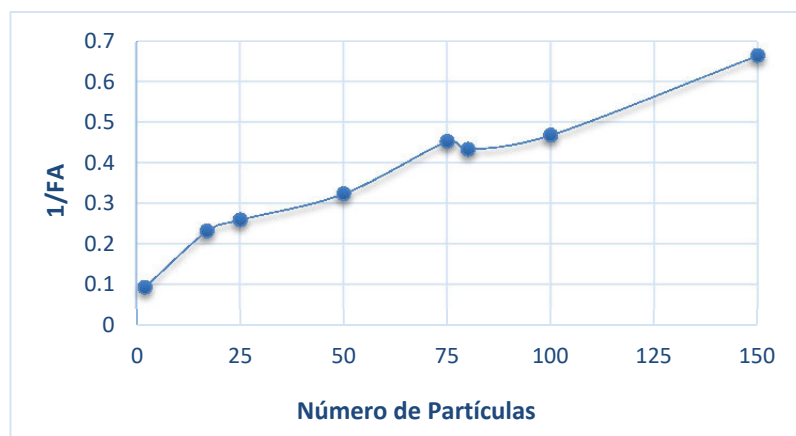


Fig. VI.D.- Eficiencia Vs No Partículas (MPI)

La similitud con la solución en porcentaje (SSP) y la eficiencia de Nash-Sutcliffe (NSE) promedio obtenidas se detallan en la siguiente tabla:

Partículas	2	17	25	50	75	80
NSE	0.907	0.924	0.932	0.930	0.946	0.933
SSP	28.57%	28.71%	28.57%	28.57%	28.43%	28.57%

Partículas	100	150
NSE	0.94838	0.95623
SSP	28.57%	28.57%

Analizando los resultados encontrados por esta rutina se observó que se asignó un volumen de fuga en todos los nodos y por tal motivo se acertó en 4 ocasiones (donde sí se tenía fuga), generando que el SSP arrojara un nivel de similitud del 28.57% en todos los casos.

Los valores máximos obtenidos de NSE y SSP se indican en la siguiente tabla:

Partículas	Valores Máximos	
	NSE	SSP
2	0.9625	28.57%
17	0.9651	35.71%
25	0.9778	28.57%
50	0.9717	28.57%
75	0.9737	28.57%
80	0.9708	28.57%
100	0.9696	28.57%
150	0.9754	28.57%

Según Moriasi et al. (2007), un resultado en el método Nash-Sutcliffe entre cero y uno se considera como aceptable. En esta rutina siempre se obtuvieron resultados en este rango a pesar de carecer una similitud significativa. Los valores promedio de SSP y NSE fueron 29.46% y 0.9708275 respectivamente.

El factor que más impactó el desempeño de esta rutina es la función objetivo pues esta restringe el rango de resultados posibles a la PSO mediante la penalización (Fig. VI.E).

El tiempo promedio de ejecución y el número promedio de rutinas alcanzadas para cada grupo de partículas se detallan en la siguiente tabla:

Partículas	Tiempo (s)	Iteraciones
2	388.70	2,054.16
17	1,357.64	1,590.06

Partículas	Tiempo (s)	Iteraciones
25	2,009.60	1,914.48
50	3,372.77	1,828.64
75	4,915.28	1,569.02
80	4,959.73	1,466.42
100	6,191.74	1,561.02
150	9,295.46	1,644.82

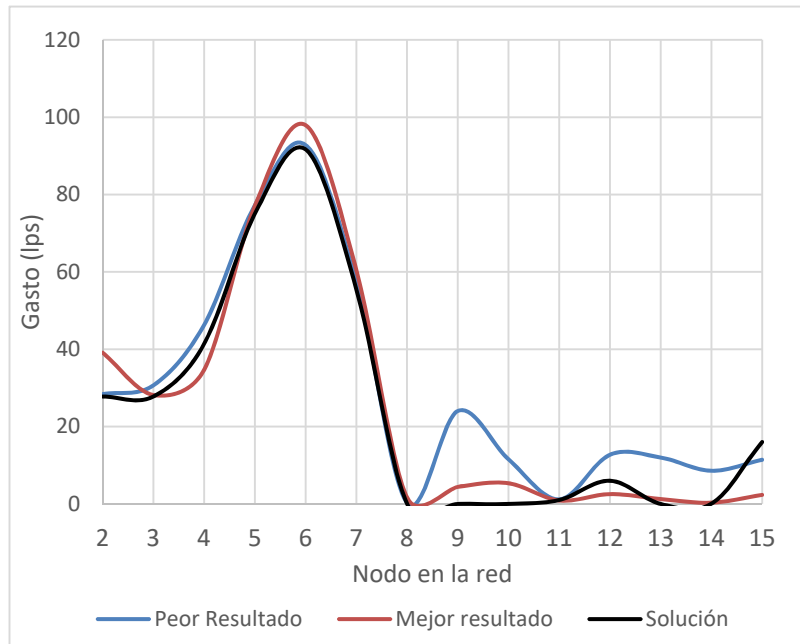


Fig. VI.E.-Resultados Vs solución (Mejor partícula en iteración).

Para la obtención de estos resultados fue necesaria la propuesta y revisión de más de 40.4 millones de redes hidráulicas.

- En resumen, el pseudo código de MPI es el siguiente:

Inicializar las posiciones $X(i,j)$ y las velocidades $V(i,j)$ de todas las partículas

for $i = 1$ until “número de partículas”

for $j=1$ until “número de nodos en la red”

*$X(i,j) = (X_{max}-X_{min})*rand()+X_{min}$*

*$V(i,j) = (V_{max}-V_{min})*rand()+V_{min}$*

end

end

While iteraciones < 800

for $i = 1$, until “número de partículas”

Ejecutar la función objetivo y obtener $FA(i,1)$

if iteraciones ==1

$FAOWNBEST(1,1) = FA(i,1)$; Por no tenerse más datos

$OWNBEST(1,:)=X(i,:)$; Por no tenerse más datos

else

if $FA(i,1) < FAOWNBEST(i,1)$

$FAOWNBEST(i,1)=FA(i,1)$

$OWNBEST(i,:)=X(i,:)$

$Pi(i,:)=OWNBEST(i,:)$; Mejor encontrado por partícula

if $FA(i,1) < \min(\min(FAOWNBEST))$

$GLOBALBEST(1,:)=X(:,1)$; Mejor histórico

iteraciones=1 ; Se resetean las iteraciones

end

end

end

for $i=1$ until “número de partículas”

if $FA(i,1) < \min(\min(FA))$

$GROUPBEST(1,:)= X(i,:)$

```

Pg(1,:) = GROUPBEST(1,:) ; Mejor en la iteración
end
end

if min(min(FAOWNBEST)) == 0
    then "fin de ciclo"
else
    Calcular velocidades  $V_{ij}$  y las posiciones  $X_{ij}$  de todas las
    partículas
    for i = 1 until "número de partículas"
        for j=1 until "número de dimensiones"
             $C_1 = \text{rand}() * 2$  ;  $C_2 = \text{rand}() * 2$  ;
             $\Phi_1(i,j) = C_1 * \text{rand}()$  ;  $\Phi_2(i,j) = C_2 * \text{rand}()$  ;
             $V(i,j) = V(i,j) + \Phi_1(i,j) * [P_i(i,j) - X(i,j)]$ 
                 $+ \Phi_2(i,j) [Pg(1,j) - X(i,j)]$ 

            if  $V(i,j) > V_{max}$ 
                 $V(i,j) = V_{max}$ 
            end

            if  $V(i,j) < V_{min}$  then
                 $V(i,j) = V_{min}$ 
            end

        end
    end
    iteraciones = iteraciones + 1
loop

```

VI.2.2.) Mejor partícula histórica (MPH).

Se realizó una modificación a la rutina anterior (MPI) con la finalidad de que el grupo fuera influenciado por la mejor partícula durante toda la rutina (MPH). A diferencia de la anterior, esta rutina va analizando el resultado de todas las partículas y, cuando se encuentra un mejor resultado histórico, se almacena e influencia socialmente a las demás partículas hasta encontrar una partícula con mejor resultado (Cai et al., 2009).

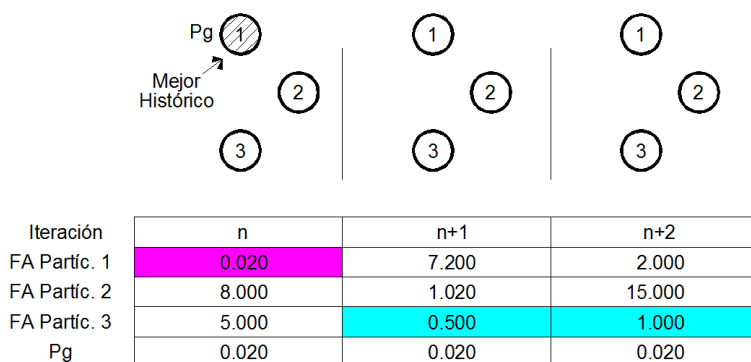


Fig. VI.F.- Mejor partícula histórica.

Al igual que en la MPI, se simularon 52 rutinas para diferentes números de partículas (2, 17, 25, 50, 75, 80, 100 y 150 partículas). En el subcapítulo anterior se especificó el por qué de la selección de dichos números.

Los parámetros de la PSO para MPH fueron los siguientes:

- Inicialización de posiciones y velocidades aleatorias dentro del espacio de búsqueda definido y rango de velocidades establecido.
- Topología de estrella (gbest).

- Pg se asigna a la mejor partícula en la rutina.
- $V_{max} = 0.33 * Vol. \text{ Fugas} = 10.23$
- $V_{min} = -V_{max} = -10.23$
- $\varphi_1 = 2 * rand()$, $\varphi_2 = 2 * rand()$

El mejor resultado tuvo un factor de aptitud de 0.21812143 ($1/FA = 4.58460226$), este se presentó cuando se usaron 150 partículas. El peor factor de aptitud fue de 16.6573429 ($1/FA = 0.06003358$) y se obtuvo cuando se utilizaron 2 partículas. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	31.04	28.11	33.88	76.86	93.98	58.40	16.27
Peor FA	35.71	38.09	37.92	77.27	95.40	57.09	27.41

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	6.47	1.09	0.74	4.42	9.06	0.82	0.16
Peor FA	1.11	21.46	3.39	3.16	14.68	1.63	19.55

Para cada grupo de partículas se obtuvo el factor de aptitud promedio. Los resultados fueron los siguientes:

Partículas	1/FA	Partículas	1/FA
2	0.138639	75	0.946742
17	0.538767	80	0.985268
25	0.572058	100	1.066395
50	0.816473	150	1.097067
PROMEDIO 1/FA		0.770176	

A diferencia de la MPI, el mejor resultado significativo no se obtuvo con un mayor número de partículas. Se puede observar que, por un estrecho margen, los enjambres con 150 partículas obtuvieron un mejor desempeño que los constituidos por 100.

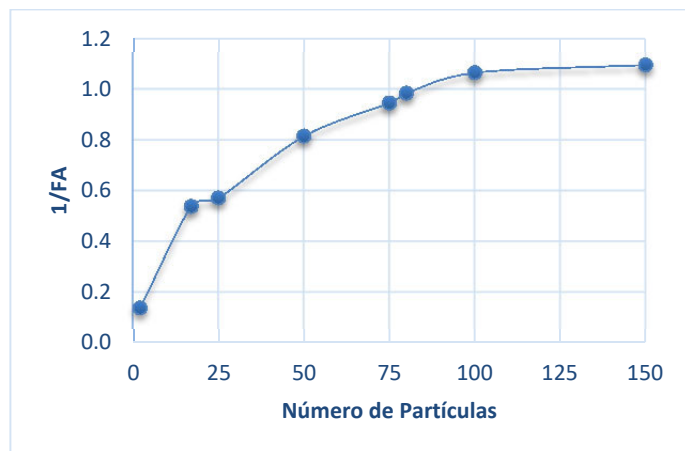


Fig. VI.G.- Eficiencia Vs No Partículas (MPH).

La similitud con la solución en porcentaje (SSP) y la eficiencia de Nash-Sutcliffe (NSE) promedio obtenidas se detallan en la siguiente tabla:

Partículas	2	17	25	50	75	80
NSE	0.91455	0.93571	0.94017	0.94045	0.94210	0.94292
SSP	28.57%	28.57%	28.43%	28.57%	28.43%	28.57%

Partículas	100	150
NSE	0.95183	0.93532
SSP	28.43%	29.43%

Las simulaciones realizadas para un enjambre de 75 y 100 partículas mostraron un cambio en el SSP. Esto demostró una mejora pues, a pesar de que a simple inspección se observa un decremento en la similitud, analizando los datos se notó que el código comenzó a proponer redes con nodos que no presentaban fuga, lo cual se asemeja a la realidad; el código ha dejado de divagar. Los valores promedio obtenidos para el SSP y NSE fueron de 29.46% y 0.977196 respectivamente.

Los valores máximos obtenidos de NSE y SSP se indican en la siguiente tabla:

Partículas	Valores Máximos	
	NSE	SSP
2	0.9686201	28.57%
17	0.9756062	28.57%
25	0.9794524	28.57%
50	0.9823853	28.57%
75	0.9790704	28.57%
80	0.9742649	28.57%
100	0.9790927	28.57%
150	0.9790785	35.71%

Las simulaciones realizadas con MPH presentaron un mejor desempeño que las de MPI. Por este motivo, el código a optimizar mediante otras técnicas será el de la mejor partícula histórica (MPH).

Se observó que el uso de 80 partículas provee una buena relación entre eficiencia y tiempo-computacional. Por este motivo, todas las demás simulaciones realizadas en esta tesis se realizarán con enjambres de 80 partículas.

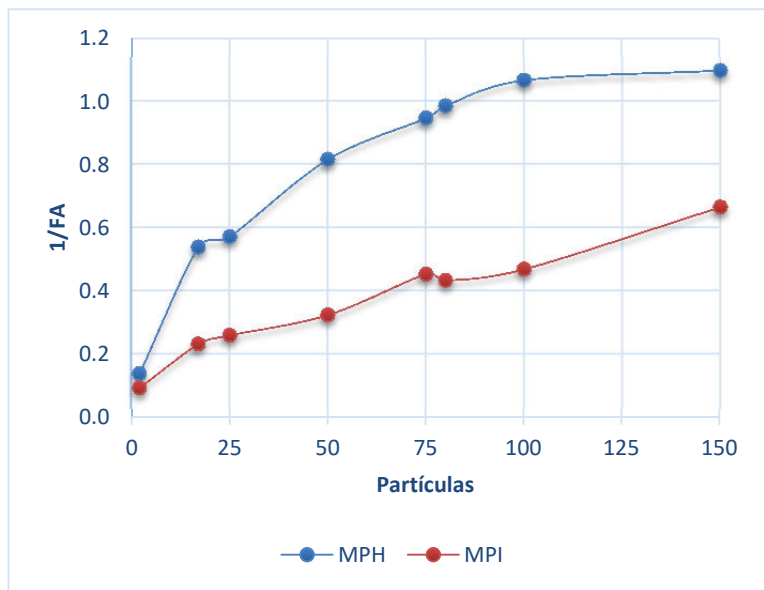


Fig. VI.H.- MPH Vs MPI.

Los tiempos de ejecución por partícula y las iteraciones se detallan a continuación:

Partículas	Tiempo (s)	Iteraciones
2	228.66	1,707.52
17	1,633.43	1,812.66
25	1,698.04	1,644.96
50	3,341.51	1,686.26
75	5,119.77	1,689.58
80	5,467.35	1,633.76
100	6,916.15	1,676.08
150	7,995.32	1,371.88

Se observa que la MPH se realizó en un menor tiempo que la MPI y obtuvo un mejor FA. Para la obtención de estos resultados se requirió la revisión de 39.5 millones de redes hidráulicas.

- En resumen, el pseudo código de MPH es el siguiente:

Inicializar las posiciones $X(i,j)$ y las velocidades $V(i,j)$ de todas las partículas

for i = 1 until “número de partículas”

for j=1 until “número de nodos en la red”

*$X(i,j) = (X_{max}-X_{min})*rand()+X_{min}$*

*$V(i,j) = (V_{max}-V_{min})*rand()+V_{min}$*

end

end

While iteraciones < 800

for i = 1, until “número de partículas”

Ejecutar la función objetivo y obtener $FA(i,1)$

if iteraciones ==1

$FAOWNBEST(1,1) = FA(i,1)$; Por no tenerse más datos

$OWNBEST(1,:)=X(i,:)$; Por no tenerse más datos

else

if $FA(i,1) < FAOWNBEST(i,1)$

$FAOWNBEST(i,1)=FA(i,1)$

$OWNBEST(i,:)=X(i,:)$

$Pi(i,:)=OWNBEST(i,:)$; Mejor encontrado por partícula

if $FA(i,1) < \min(\min(FAOWNBEST))$

$GLOBALBEST(1,:)=X(:,1)$

$Pg(1,:) = GLOBALBEST(1,:)$; Mejor histórico

iteraciones=1 ; Se resetean las iteraciones

end

end

```

end

if min(min(FAOWNBEST)) == 0
    then "fin de ciclo"
else
    Calcular velocidades  $V_{ij}$  y las posiciones  $X_{ij}$  de todas las
    partículas
    for i = 1 until "número de partículas"
        for j=1 until "número de dimensiones"
             $C_1 = \text{rand}() * 2$  ;  $C_2 = \text{rand}() * 2$  ;
             $\Phi_1(i,j) = C_1 * \text{rand}()$  ;  $\Phi_2(i,j) = C_2 * \text{rand}()$  ;
             $V(i,j) = V(i,j) + \Phi_1(i,j) * [P_i(i,j) - X(i,j)]$ 
                 $+ \Phi_2(i,j) [Pg(1,j) - X(i,j)]$ 

            if  $V(i,j) > V_{\max}$ 
                 $V(i,j) = V_{\max}$ 
            end

            if  $V(i,j) < V_{\min}$  then
                 $V(i,j) = V_{\min}$ 
            end

        end
    end
    iteraciones = iteraciones + 1
loop

```

VI.2.3.) Formación Anillo (PSOFA).

Habiéndose elegido a la rutina MPH como la que mejor desempeño tuvo, se procedió a modificarla con otra técnica. Según varios autores (Bratton & Kennedy, 2007; Parsopoulos & Vrahatis, 2008), esta formación evita una convergencia prematura de la PSO y favorece la exploración.

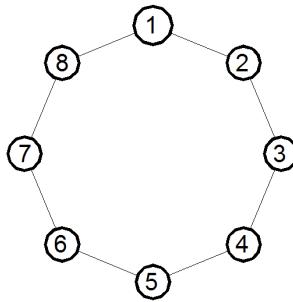


Fig. VI.I.- Formación Anillo.

Para la implementación de la formación anillo se realizaron pequeños subenjambres con 3 partículas cada uno (Fig. VI.J.), en los que cada uno tiene una influencia social ejercida por el mejor elemento comprendido dentro del subenjambre.

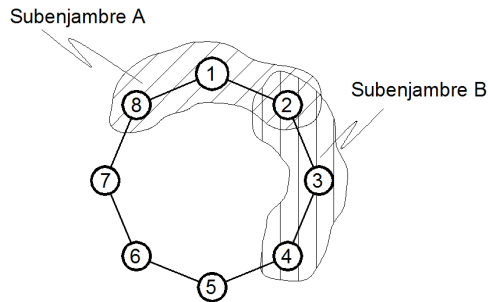


Fig. VI.J.- Influencia social en subenjambre.

Los vecinos para cada partícula considerados en esta rutina son los inmediatos superior e inferior. De esta manera, la partícula 2 tendrá dos vecinos: la partícula 1 y 3; la partícula 1 tendrá de igual forma 2 vecinos: la 8 y la 2.

Se realizaron 102 simulaciones con 80 partículas cada una. Como ya se expresó con anterioridad, la simulación con el mejor y peor resultado fueron descartadas y las otras 100 fueron sujetas de análisis.

Los parámetros de la PSO para PSOFA fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones y velocidades aleatorias dentro del espacio de búsqueda definido y rango de velocidades establecido.
- Topología de anillo: 3 integrantes con 2 vecinos por partícula (inmediato superior e inferior).
- Se considera un P_g para cada subenjambre. El mejor resultado histórico obtenido por algún elemento del subenjambre, será el que ejerza la influencia social sobre los demás elementos de esta agrupación.
- $V_{max} = 0.33 * Vol.$ Fugas = 10.23
- $V_{min} = -V_{max} = -10.23$
- $\varphi_1 = 2 * rand()$, $\varphi_2 = 2 * rand()$

El mejor resultado tuvo un factor de aptitud de 0.969264 ($1/FA = 1.03171$) y se obtuvo con 2361 iteraciones. El peor factor de aptitud fue de 2.237914 ($1/FA = 0.446845$) y se obtuvo con 1819 iteraciones. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	30.09	28.64	45.74	77.55	104.39	60.42	3.33
Peor FA	49.46	33.81	47.84	75.13	95.84	55.64	11.63

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	1.55	4.46	1.77	2.15	1.52	1.23	0.86
Peor FA	5.92	9.91	0.64	2.89	3.53	0.42	3.61

El factor de aptitud promedio de la PSOFA fue de 1.545142 ($1/FA = 0.6471899$). Se obtuvo una similitud con la solución en porcentaje promedio (SSP) de 28.57%. La eficiencia de Nash-Sutcliffe promedio

(NSE) obtenida fue de 0.944208. Los valores máximos para NSE y SSP fueron 0.973290 y 28.57% respectivamente.

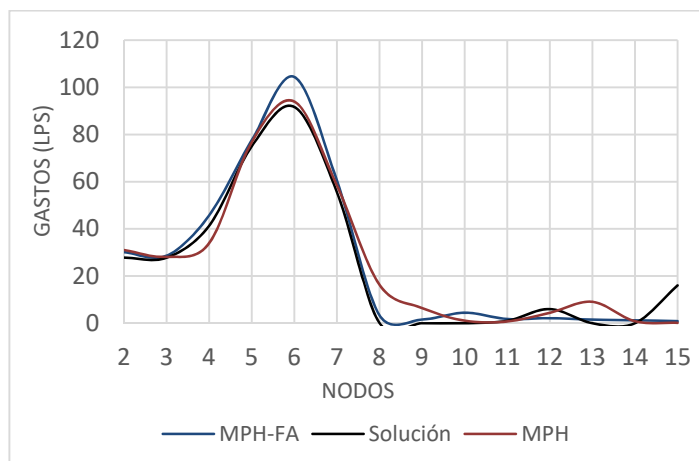


Fig. VI.K.- MPH Vs PSOFA.

La modalidad PSOFA obtuvo un mejor resultado en el NSE y en el SSP. A pesar de lo anterior, la PSOFA obtuvo un factor de aptitud considerablemente menor a la MPH.

	MPH (Para 80 partículas)	PSOFA
SSP	28.57%	28.57%
NSE	0.94292	0.94421
1/FA	0.985268	0.647190
Iteraciones	1,633.76	2,007.23
Tiempo	5,467.35	6,713.04

Como PSOFA no obtuvo un menor factor de aptitud (FA) y además requiere más iteraciones y más tiempo, se elige descartar dicha metodología y continuar modificando a MPH.

Para la obtención de estos resultados se requirió la revisión de más de 16 millones de redes hidráulicas.

- En resumen, el pseudo código de PSOFA es el siguiente:

; Inicializar las posiciones $X(i,j)$ y las velocidades $V(i,j)$ de todas las partículas

for i = 1 until “número de partículas”

for j=1 until “número de nodos en la red”

*$X(i,j) = (X_{max}-X_{min})*rand()+X_{min}$*

*$V(i,j) = (V_{max}-V_{min})*rand()+V_{min}$*

end

end

While iteraciones < 800

for i = 1, until “número de partículas”

Ejecutar la función objetivo y obtener $FA(i,1)$

if iteraciones ==1

$FAOWNBEST(1,1) = FA(i,1)$; Por no tenerse más datos

$OWNBEST(1,:)=X(i,:)$; Por no tenerse más datos

else

if $FA(i,1) < FAOWNBEST(i,1)$

$FAOWNBEST(i,1)=FA(i,1)$

$OWNBEST(i,:)=X(i,:)$

$Pi(i,:)=OWNBEST(i,:)$; Mejor encontrado por partícula

if $FA(i,1) < \min(\min(FAOWNBEST))$

$GLOBALBEST(1,:)=X(:,1)$

iteraciones=1 ; Se resetean las iteraciones

end

end

end

; Determinación de vecinos para cada partícula

for i=1:1:partículas


```

    if i==1
        partant=particulas
    else
        partant=i-1
    end

    if i==particulas
        partpost=1
    else
        partpost=i+1
    end

; Determinando cuál partícula es la mejor del subenjambre
    if FAOWNBEST(i,1)<FAOWNBEST(partant,1)
        if FAOWNBEST(i,1)<FAOWNBEST(partpost,1)
            GROUPBEST(i,:)=OWNBEST(i,:);
        else
            GROUPBEST(i,:)=OWNBEST(partpost,:);
        end
    else
        if FAOWNBEST(partant,1) <FAOWNBEST(partpost,1)
            GROUPBEST(i,:)=OWNBEST(partant,:);
        else
            GROUPBEST(i,:)=OWNBEST(partpost,:);
        end
    end

Pg(i,1)=GROUPBEST(i,:); Mejor del grupo

end

if min(min(FAOWNBEST))==0
    then "fin de ciclo"
else
    Calcular velocidades  $V_{ij}$  y las posiciones  $X_{ij}$  de todas las
    partículas

```

```

for i = 1 until "número de partículas"
  for j=1 until "número de dimensiones"
    C1 = rand()*2 ; C2 = rand()*2 ;
    Φ1(i,j) = C1*rand() ;      Φ2(i,j) = C2*rand() ;
    V(i,j) = V(i,j) + Φ1(i,j) * [ Pi(i,j) - X(i,j) ]
              + Φ2(i,j) [ Pg(1,j) - X(i,j) ]
    if V(i,j) > Vmax then
      V(i,j) = Vmax
    end
    if V(i,j) < Vmin then
      V(i,j) = Vmin
    end
  end
end
iteraciones = iteraciones + 1
loop

```

VI.2.4.) Inicialización Hammersley (IH).

La implementación de la formación anillo (PSOFA) no brindó los resultados esperados y por tal motivo fue descartada. Se buscó mejorar la eficiencia de la PSO modificando la inicialización de partículas.

En la fig.VI.L. se observa que, al momento de inicializar las posiciones de las partículas de manera aleatoria, se obtienen datos con cierta tendencia la cual requiere de varias iteraciones para ser eliminada (Wong et al., 1997).

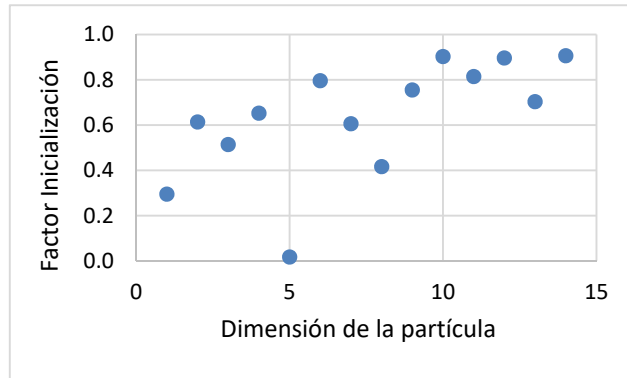


Fig. VI.L.- Inicialización Aleatoria.

La secuencia Hammersley & Halton proporciona vectores direccionales uniformemente distribuidos (Cui & Freedman, 1997), los cuales mejoran la posibilidad de ubicar a las partículas de la PSO en un punto cercano a la solución óptima y poder obtener un mejor global en un menor número de iteraciones (Fig. VI.M.).

El análisis matemático de la secuencia Hammersley & Halton excede los límites de esta tesis, para una mayor información el lector se puede referir al artículo “Sampling with Hammersley and Halton Points” de Wong et al. (1997).

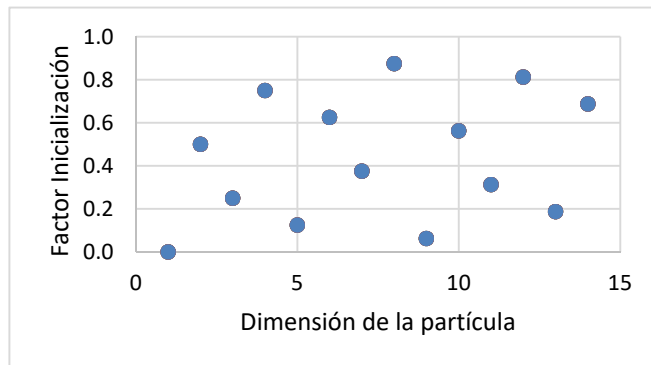


Fig. VI.M.- Inicialización por Hammersley.

Es importante hacer notar que la inicialización con Hammersley fue la única técnica implementada en la PSO que no se programó por el autor. Para la inicialización de partículas por este método, se descargó la rutina complementaria “Hammersley_Dataset.m” desarrollada por John Burkardt del sitio web: http://people.sc.fsu.edu/~jburkardt/m_src/hammersley_dataset/hammersley_dataset.m

Los parámetros utilizados para la configuración de la rutina “Hammersley_Dataset.m” y su implementación en la inicialización de partículas en la detección de fugas por PSO fueron los siguientes:

NDIM=1.

N= Número de nodos.

SEED, LEAP, STEP = Número primo aleatorio menor a 109.

BASE =2.

Los valores anteriormente mencionados se determinaron mediante la puesta en marcha de la rutina y la observación directa de las posiciones asignadas por esta.

Se realizaron 102 simulaciones con 80 partículas cada una. La simulación con el mejor y peor resultado fueron descartadas y las otras 100 fueron sujetas de análisis.

Los parámetros de la PSO para IH fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones definidas mediante la secuencia Hammersley dentro del espacio de búsqueda (NDIM=1, N=14, BASE=2; SEED, LEAP y STEP = Número primo aleatorio menor a 109).
- Inicialización de velocidades aleatorias dentro del rango de velocidades establecido.
- Topología de estrella (gbest).

- Pg se asigna a la mejor partícula en la rutina.
- $V_{max} = 0.33 * Vol. \text{ Fugas} = 10.23$
- $V_{min} = -V_{max} = -10.23$
- $\varphi_1 = 2 * rand()$, $\varphi_2 = 2 * rand()$

El mejor resultado tuvo un factor de aptitud de 0.349021 ($1/FA = 2.865154$) y se obtuvo con 1998 iteraciones. El peor factor de aptitud fue de 2.117479 ($1/FA = 0.472260$) y se obtuvo con 1059 iteraciones. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	44.36	28.11	35.88	75.24	92.03	61.01	8.67
Peor FA	37.90	30.99	41.22	80.53	91.84	63.08	6.71

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	1.79	5.36	1.37	6.59	5.06	1.94	0.13
Peor FA	6.05	12.68	1.89	0.56	0.41	0.61	1.41

El factor de aptitud promedio de la IH fue de 1.067404 ($1/FA = 0.936852$).

Se obtuvo una similitud con la solución en porcentaje promedio (SSP) de 28.57%. La eficiencia de Nash-Sutcliffe promedio (NSE) obtenida fue de 0.945031. Los valores máximos obtenidos para NSE y SSP fueron 0.985193 y 35.71 % respectivamente.

Como se muestra en la siguiente tabla, la diferencia entre la IH y la MPH fue mínima a favor de esta última.

	MPH	IH
FA	1.014952	1.067404
1/FA	0.985268	0.936852
SSP	28.57%	28.57%
NSE	0.942921	0.945031
Tiempo (s)	5,467.35	5,325.58
Iteraciones	1,633.76	1,626.49

Por no haber obtenido un mejor factor de aptitud, se decide descartar la inicialización por Hammersley (IH) y continuar con la búsqueda de una técnica para mejorar la MPH.

Para la obtención de estos resultados, se requirió la revisión y evaluación de más de 13 millones de redes hidráulicas.

- En resumen, el pseudo código de IH es el siguiente:

Inicializar las posiciones $X(i,j)$ mediante Hammersley

for $i = 1$ until “número de partículas”

for $j=1$ until “número de nodos en la red”

NDIM=1; N= 14 ; BASE =2

SEED = random(Número primo < 109)

LEAP = random(Número primo < 109)

STEP = random(Número primo < 109)

$X(i,j) = \text{run}(\text{“Hammersley_Dataset.m”})$

end

end

Inicializar las velocidades $V(i,j)$ de todas las partículas de manera aleatoria

for $i = 1$ until “número de partículas”

for $j=1$ until “número de nodos en la red”

*$V(i,j) = (V_{\max} - V_{\min}) * \text{rand}() + V_{\min}$*

```

        end
    end

    While iteraciones < 800
        for i = 1, until "número de partículas"
            Ejecutar la función objetivo y obtener FA(i,1)
            if iteraciones == 1
                FAOWNBEST(1,1) = FA(i,1) ; Por no tenerse más datos
                OWNBEST(1,:)=X(i,:) ; Por no tenerse más datos
            else
                if FA(i,1) < FAOWNBEST(i,1)
                    FAOWNBEST(i,1)=FA(i,1)
                    OWNBEST(i,:)=X(i,:)
                    Pi(i,:)=OWNBEST(i,:) ; Mejor encontrado por partícula

                    if FA(i,1) < min(min(FAOWNBEST))
                        GLOBALBEST(1,:)=X(:,1)
                        Pg(1,:) = GLOBALBEST(1,:) ; Mejor histórico
                        iteraciones=1 ; Se resetean las iteraciones
                    end
                end
            end
        end

        if min(min(FAOWNBEST)) == 0
            then "fin de ciclo"
        else
            Calcular velocidades Vij y las posiciones Xij de todas las
            partículas de manera aleatoria.
            for i = 1 until "número de partículas"
                for j=1 until "número de dimensiones"
                    C1= rand()*2 ; C2= rand()*2 ;
                     $\Phi_1(i,j) = C_1 * rand()$  ;       $\Phi_2(i,j) = C_2 * rand()$  ;
                end
            end
        end
    end

```

```


$$V(i,j) = V(i,j) + \phi_1(i,j) * [P_i(i,j) - X(i,j)]$$


$$+ \phi_2(i,j) [P_g(1,j) - X(i,j)]$$


if  $V(i,j) > V_{max}$ 
     $V(i,j) = V_{max}$ 
end

if  $V(i,j) < V_{min}$  then
     $V(i,j) = V_{min}$ 
end

end
end
iteraciones = iteraciones + 1
loop

```

VI.2.5.) Inicialización velocidades por Half-Diff (IHD).

Maurice Clerc (2010) propone en su artículo “From Theory to Practice in Particle Swarm Optimization” una manera de inicializar las velocidades de las partículas llamada “Half-Diff”.

Si la partícula comienza con una velocidad muy baja, le tomará varias iteraciones el acercarse a un óptimo local. De la misma manera, una velocidad muy alta hace que la partícula dé “saltos muy grandes” en el espacio de búsqueda y, su desaceleración para la obtención de un buen resultado, requiera de varias iteraciones. Por otra parte, una velocidad muy alta puede hacer que las partículas se “estrellen” en los límites del espacio de búsqueda y les sea muy difícil adquirir valores no marginales debido a los grandes valores que la componente velocidad le proporciona a la PSO en la asignación de nuevas posiciones.

La inicialización de velocidades por el método “Half-Diff” se obtiene con la siguiente fórmula (Clerc, 2010):

$$V(0)_{i,j} = 0.50 * \{rand(X_{\max_{i,j}}, X_{\min_{i,j}}) - X(0)_{i,j}\}$$

Donde:

$V(0)_{i,j}$ = Velocidad inicial para la partícula “i” en la dimensión “j”

$rand(X_{\max_{i,j}}, X_{\min_{i,j}})$ = Número aleatorio con distribución normal comprendido en el espacio de búsqueda de la partícula “i” en la dimensión “j”

$X(0)_{i,j}$ = Posición inicial de la partícula “i” en la dimensión “j”

Se realizaron 102 simulaciones con 80 partículas cada una. La simulación con el mejor y peor resultado fueron descartadas y las otras 100 fueron sujetas de análisis.

Los parámetros de la PSO para IHD fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones de manera aleatoria dentro del espacio de búsqueda.
- Inicialización de velocidades mediante el método Half-Diff.
- Topología de estrella (gbest).
- Pg se asigna a la mejor partícula en la rutina.
- $V_{\max} = 0.33 * Vol. \text{ Fugas} = 10.23$
- $V_{\min} = -V_{\max} = -10.23$
- $\varphi_1 = 2 * rand()$, $\varphi_2 = 2 * rand()$

El mejor resultado tuvo un factor de aptitud de 0.349536 ($1/FA = 2.860938$) y se obtuvo con 2311 iteraciones. El peor factor de aptitud fue de 2.154936 ($1/FA = 0.464051$) y se obtuvo con 1043 iteraciones. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	28.35	30.74	37.91	77.37	93.73	57.71	4.00
Peor FA	43.79	34.84	39.44	76.48	92.88	58.61	3.04

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	2.74	1.59	2.66	5.84	4.67	0.92	2.27
Peor FA	2.45	2.87	0.47	4.56	16.28	1.17	2.65

El factor de aptitud promedio de la IHD fue de 1.0448728 ($1/FA = 0.95705431$).

Se obtuvo una similitud con la solución en porcentaje promedio (SSP) de 28.50%. La eficiencia de Nash-Sutcliffe promedio (NSE) obtenida fue de 0.944960. Los valores máximos para NSE y SSP fueron 0.979482 y 28.57% respectivamente.

La comparativa entre la IHD y la MPH se muestra en la siguiente tabla:

	MPH	IHD
FA	1.014952	1.044873
1/FA	0.985268	0.957054
SSP	28.57%	28.50%
NSE	0.942921	0.944960
Tiempo	5,467.35	5,396.13
Iteraciones	1,633.76	1,643.11

Pese a que no se obtuvo mejora alguna en el factor de aptitud, NSE y SSP, se optó por dejar la inicialización de velocidades por Half-Diff porque provee de múltiples velocidades a las partículas; esto aporta una diversidad de exploración al código.

Para la obtención de estos resultados, se requirió la revisión y evaluación de más de 13.1 millones de redes hidráulicas.

- En resumen, el pseudo código de IHD es el siguiente:

Inicializar las posiciones $X(i,j)$ aleatoriamente

for $i = 1$ until “número de partículas”

for $j=1$ until “número de nodos en la red”

*$X(i,j) = (X_{max}-X_{min})*rand()+X_{min}$*

end

end

Inicializar las velocidades $V(i,j)$ de todas las partículas por el método “Half-Diff”

for $i = 1$ until “número de partículas”

for $j=1$ until “número de nodos en la red”

*$V(i,j) = 0.50*rand(X_{max},X_{min})-X_{ij}$*

end

end

While iteraciones < 800

for $i = 1$, until “número de partículas”

Ejecutar la función objetivo y obtener $FA(i,1)$

if iteraciones ==1

$FAOWNBEST(1,1) = FA(i,1)$; Por no tenerse más datos

$OWNBEST(1,:)=X(i,:)$; Por no tenerse más datos

else

if $FA(i,1) < FAOWNBEST(i,1)$

$FAOWNBEST(i,1)=FA(i,1)$

$OWNBEST(i,:)=X(i,:)$

$Pi(i,:)=OWNBEST(i,:)$; Mejor encontrado por partícula

if $FA(i,1) < \min(\min(FAOWNBEST))$

$GLOBALBEST(1,:)=X(:,1)$

$Pg(1,:) = GLOBALBEST(1,:)$; Mejor histórico

```

                                iteraciones=1 ; Se resetean las iteraciones
                                end
                            end
                        end

                    if min(min(FAOWNBEST)) ==0
                        then "fin de ciclo"
                    else
                        Calcular velocidades  $V_{ij}$  y las posiciones  $X_{ij}$  de todas las
                        partículas de manera aleatoria.
                        for i = 1 until "número de partículas"
                            for j=1 until "número de dimensiones"
                                 $C_1 = rand() * 2$  ;  $C_2 = rand() * 2$  ;
                                 $\Phi_1(i,j) = C_1 * rand()$  ;  $\Phi_2(i,j) = C_2 * rand()$  ;
                                 $V(i,j) = V(i,j) + \Phi_1(i,j) * [P_i(i,j) - X(i,j)]$ 
                                    +  $\Phi_2(i,j) [Pg(1,j) - X(i,j)]$ 

                                if  $V(i,j) > V_{max}$ 
                                     $V(i,j) = V_{max}$ 
                                end

                                if  $V(i,j) < V_{min}$  then
                                     $V(i,j) = V_{min}$ 
                                end
                            end
                        end
                    iteraciones = iteraciones +1
                loop

```

VI.2.6.) Variación de Velocidad Máxima (VVM).

La velocidad máxima indica que tan rápido se le permite volar a las partículas de una posición a otra. Una velocidad máxima de gran

magnitud le brinda a las partículas una mejor capacidad de exploración en el espacio de búsqueda, mientras que una menor velocidad máxima otorga una mayor explotación.

Al hacer alusión a la velocidad máxima se debe tener presente que, para esta investigación, la velocidad máxima es igual a la velocidad mínima en todos los casos, esto es:

$$V_{\max} = - V_{\min}$$

A manera de ejemplo, en un caso en el que la solución global se encuentre en 20, si la posición inicial de una partícula tiene el valor de 5, a mayor velocidad máxima, se podrá llegar más rápido a dicha solución. Pero, si la posición inicial es 19.50, una velocidad máxima de gran magnitud puede hacer que la partícula vuele de un lugar a otro y sea más difícil refinar el resultado.

Se debe recordar que la velocidad es el componente que indica el cambio de posición de la partícula y no la velocidad máxima, pero la naturaleza de la velocidad de las partículas de la PSO, es el incrementar con las iteraciones hasta explotar si no se esta no se delimita (Kennedy & Eberhart, 2001).

Muy pocos autores tratan el tema de la velocidad máxima en sus publicaciones y, a excepción de Kennedy & Eberhart (2001), no se encontró recomendación alguna para este factor en la bibliografía consultada. Kennedy & Eberhart (2001), indican que un buen valor para la velocidad máxima es ± 4 .

Con la finalidad de analizar si la variación de la velocidad máxima produce un cambio en la obtención de resultados de la PSO, se decidió realizar diversas simulaciones con diferentes valores para la velocidad máxima.

Se dividió el espacio de búsqueda en 4 partes 25%, 50%, 75% y 100% para asignar estos valores a la velocidad máxima. El volumen de fugas (espacio de búsqueda) para la red en cuestión es de 31 lps, por este motivo, se simuló para velocidades máximas de ± 4 (recomendación de Kennedy & Eberhart), ± 7.75 (25%), ± 15.50 (50%), ± 23.25 (75%) y ± 31 (100%).

Para una velocidad máxima de 31, la partícula podrá volar del límite inferior al límite superior del espacio de búsqueda en una iteración (si su velocidad de vuelo es muy alta).

Se realizaron 52 simulaciones con 80 partículas para cada valor propuesto en la velocidad máxima (± 4 , ± 7.75 , ± 15.50 , ± 23.25 , ± 31). La simulación con el mejor y peor resultado fueron descartadas y las restantes fueron sujetas de análisis.

Los parámetros de la PSO para VVM fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones de manera aleatoria dentro del espacio de búsqueda.
- Inicialización de velocidades mediante el método Half-Diff.
- Topología de estrella (gbest).
- Pg se asigna a la mejor partícula en la rutina.
- $V_{max} = [\pm 4, \pm 7.75, \pm 15.50, \pm 23.25, \pm 31]$
- $V_{min} = -V_{max}$
- $\varphi_1 = 2 * rand() , \varphi_2 = 2 * rand()$

El mejor resultado se generó con una restricción en la velocidad máxima igual a 4, se tuvo un factor de aptitud de 0.023650 ($1/FA = 42.283292$), se requirieron 2031 iteraciones para su obtención. El peor factor de aptitud fue de 78.811600 ($1/FA = 0.012688$) y se obtuvo con 844 iteraciones y una velocidad máxima permitida de 31. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	29.04	28.02	38.44	76.57	97.55	58.30	2.50
Peor FA	32.07	50.48	39.13	77.10	122.04	55.79	9.56

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	0.37	3.67	0.11	5.57	1.95	1.64	3.48
Peor FA	5.19	13.86	4.58	3.95	10.76	22.42	12.61

Los factores de aptitud promedio, tiempo e iteraciones se muestran en la siguiente tabla:

Vmax	FA	1/FA	Tiempo (s)	Iteraciones
4.00	0.055683	17.958895	6,114.57	1,738.70
7.75	0.342541	2.919360	5,901.28	1,759.40
15.50	5.530342	0.180821	4,883.91	1,578.88
23.25	24.713865	0.040463	3,788.23	1,294.00
31.00	50.113315	0.019955	3,143.30	898.68

Como se puede observar, el factor de aptitud se eleva considerablemente a un menor valor absoluto en la velocidad máxima (Fig. VI.N).

Los valores promedios de SSP y NSE para VVM son los siguientes:

Vmax	NSE	SSP
4.00	0.964045	28.43%
7.75	0.954614	28.57%
15.50	0.923974	28.57%
23.25	0.877764	28.57%
31.00	0.846656	28.57%

Los valores máximos obtenidos para NSE y SSP fueron los siguientes:

Partículas	Valores Máximos	
	NSE	SSP
4.00	0.986809	28.57%
7.75	0.985297	28.57%
15.50	0.968833	28.57%
23.25	0.967721	28.57%
31.00	0.926100	28.57%

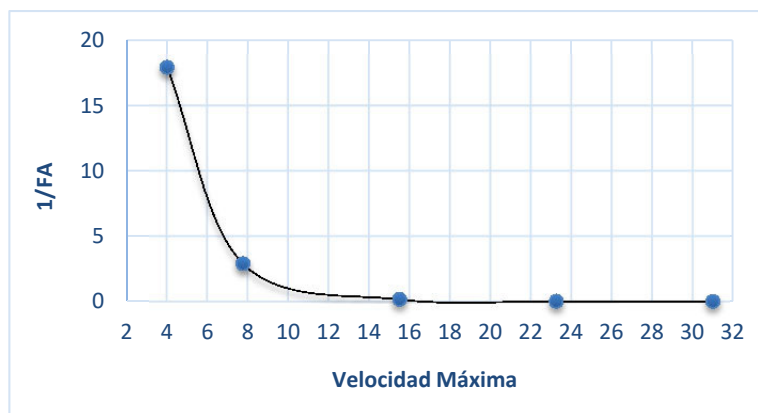


Fig. VI.N.- Resultados VVM.

En el desarrollo de esta tesis se simuló para velocidades más bajas que la recomendada por Kennedy & Eberhart (2001). A pesar de que no se muestran estos resultados, se observó que, para este escenario, el FA mejora mientras menor es el valor de la velocidad máxima. Se observó que, a velocidades menores, la PSO deja de explorar y “barre” el espacio de búsqueda.

La intención de esta investigación es la de proveer una herramienta para múltiples escenarios. El “barrido” del espacio de búsqueda

proporcionará muy buenos resultados solo si la posición de las partículas se encuentra cercana al óptimo global, además de que se corre el riesgo de quedar atrapado en un óptimo local (dependiendo de la forma del espacio de búsqueda).

Una desventaja muy grande en el barrido del espacio de búsqueda es el tiempo. A un menor valor en la velocidad máxima permitida, el tiempo de ejecución de la rutina aumenta considerablemente.

A criterio del autor, se decide descartar cualquier velocidad máxima menor a cuatro y continuar con este parámetro recomendado por Kennedy & Eberhart (2001). La finalidad de la PSO es la exploración de un amplio espacio de búsqueda, el “barrido” de este se aleja de la intención de la metodología.

La comparativa entre la VVM e IHD se muestra en la siguiente tabla:

	VVM	IHD
FA	0.055683	1.044873
1/FA	17.958895	0.957054
SSP	28.43%	28.50%
NSE	0.964045	0.944960
Tiempo (s)	6,114.57	5,396.13
Iteraciones	1,738.70	1,643.11

Para la obtención de estos resultados, se requirió la revisión y evaluación de más de 29 millones de redes hidráulicas.

- En resumen, el pseudo código de VVM es el siguiente:

$V_{max} = 4$; $V_{min} = -V_{max}$

Inicializar las posiciones $X(i,j)$ aleatoriamente

```

for i = 1 until "número de partículas"
    for j=1 until "número de nodos en la red"
         $X(i,j) = (X_{max}-X_{min}) * rand() + X_{min}$ 
    end
end

Inicializar las velocidades  $V(i,j)$  de todas las partículas por el método
"Half-Diff"
for i = 1 until "número de partículas"
    for j=1 until "número de nodos en la red"
         $V(i,j) = 0.50 * rand(X_{max}, X_{min}) - X_{ij}$ 
    end
end

While iteraciones < 800
    for i = 1, until "número de partículas"
        Ejecutar la función objetivo y obtener  $FA(i,1)$ 
        if iteraciones == 1
             $FAOWNBEST(1,1) = FA(i,1)$  ; Por no tenerse más datos
             $OWNBEST(1,:)=X(i,:)$  ; Por no tenerse más datos
        else
            if  $FA(i,1) < FAOWNBEST(i,1)$ 
                 $FAOWNBEST(i,1)=FA(i,1)$ 
                 $OWNBEST(i,:)=X(i,:)$ 
                 $Pi(i,:)=OWNBEST(i,:)$  ; Mejor encontrado por partícula

                if  $FA(i,1) < \min(\min(FAOWNBEST))$ 
                     $GLOBALBEST(1,:)=X(:,1)$ 
                     $Pg(1,:) = GLOBALBEST(1,:)$  ; Mejor histórico
                    iteraciones=1 ; Se resetean las iteraciones
                end
            end
        end
    end
end

```

```

if min(min(FAOWNBEST)) == 0
    then "fin de ciclo"
else
    Calcular velocidades  $V_{ij}$  y las posiciones  $X_{ij}$  de todas las
    partículas de manera aleatoria.
    for i = 1 until "número de partículas"
        for j=1 until "número de dimensiones"
             $C_1 = \text{rand}() * 2$  ;  $C_2 = \text{rand}() * 2$  ;
             $\Phi_1(i,j) = C_1 * \text{rand}()$  ;  $\Phi_2(i,j) = C_2 * \text{rand}()$  ;
             $V(i,j) = V(i,j) + \Phi_1(i,j) * [P_i(i,j) - X(i,j)]$ 
                +  $\Phi_2(i,j) [Pg(1,j) - X(i,j)]$ 

            if  $V(i,j) > V_{\max}$ 
                 $V(i,j) = V_{\max}$ 
            end

            if  $V(i,j) < V_{\min}$  then
                 $V(i,j) = V_{\min}$ 
            end

        end
    end
    iteraciones = iteraciones + 1
loop

```

VI.2.7.) Clamping PSO (CMP).

La PSO propone soluciones a un problema dado. Mediante $V_{\max}$ y $V_{\min}$ se puede indicar que tan rápido cambiará de una posición a otra. Gracias a la función objetivo se le puede indicar a la PSO que tan cerca o alejada está de la solución; que tan bien se ha desempeñado.

A pesar de que las partículas de la PSO se ubican dentro de un espacio de búsqueda definido por el usuario, la PSO puede salirse de este y empezar

a proponer soluciones alejadas de la realidad; esto genera un desperdicio de tiempo y recursos computacionales por la revisión de soluciones no viables.

Ahora bien, ¿cómo se le puede indicar a la PSO que solamente elija opciones de un espacio de búsqueda válido?. Maurice Clerc en su artículo “From Theory to Practice in Particle Swarm Optimization” menciona el método Clamping para este fin (Clerc, 2010):

$$\text{Clamping PSO} \left\{ \begin{array}{ll} X_{(t+1)} < X_{\min} & \left\{ \begin{array}{l} X_{(t+1)} = X_{\min} \\ V_{(t+1)} = 0 \end{array} \right. \\ X_{(t+1)} > X_{\max} & \left\{ \begin{array}{l} X_{(t+1)} = X_{\max} \\ V_{(t+1)} = 0 \end{array} \right. \end{array} \right.$$

Adecuando el método a esta investigación, se tendrá que:

$$\left\{ \begin{array}{ll} X_{ij} < QDEM_{ij} & \left\{ \begin{array}{l} X_{ij} = QDEM_{ij} \\ V_{ij} = 0 \end{array} \right. \\ X_{ij} > QDEM_{ij} + qdif & \left\{ \begin{array}{l} X_{ij} = QDEM_{ij} + qdif \\ V_{ij} = 0 \end{array} \right. \end{array} \right.$$

El procedimiento de la Clamping PSO es el siguiente:

1. Calcular las nuevas posiciones para la siguiente iteración.
2. Revisar si las posiciones calculadas están dentro del rango deseado [Qdem, Qdem+qdif].
3. Si una posición es menor a Qdem, la posición y velocidad para la siguiente iteración serán Qdem, y cero respectivamente.
4. Si una posición es mayor a (Qdem+qdif), la posición y velocidad para la siguiente iteración serán (Qdem+qdif), y cero respectivamente.
5. Si las posiciones calculadas están dentro del espacio de búsqueda deseado [Qdem , Qdem+qdif], estas no sufrirán cambio alguno.

Se realizaron 102 simulaciones con 80 partículas. La simulación con el mejor y peor resultado fueron descartadas y las 100 restantes fueron sujetas de análisis.

Los parámetros de la PSO para CMP fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones de manera aleatoria dentro del espacio de búsqueda.
- Inicialización de velocidades mediante el método Half-Diff.
- Topología de estrella (gbest).
- P_g se asigna a la mejor partícula en la rutina.
- $V_{max} = 4$
- $V_{min} = -V_{max}$
- $\varphi_1 = 2 * rand()$, $\varphi_2 = 2 * rand()$
- Clamping PSO
 - ✓ Si $X < Q_{dem} \rightarrow X = Q_{dem}$, $V = 0$;
 - ✓ Si $X > Q_{dem} + q_{dif} \rightarrow X = Q_{dem} + q_{dif}$, $V = 0$

El mejor resultado tuvo un factor de aptitud de 0.000221429 ($1/FA = 4516.12$), se requirieron 3261 iteraciones para su obtención. El peor factor de aptitud fue de 2.577557 ($1/FA = 0.39$) y se obtuvo con 848 iteraciones. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	27.78	27.78	62.26	83.94	91.67	55.56	0.00
Peor FA	27.78	27.78	33.33	75.00	106.35	55.56	0.00

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	0.00	0.02	0.00	0.00	0.00	0.00	3.48
Peor FA	31.00	0.00	0.00	0.00	0.00	0.00	0.00

El factor de aptitud promedio de la CMP fue de 0.377483 ($1/FA = 2.649127$).

Se obtuvo una similitud con la solución en porcentaje promedio (SSP) de 56.64%. La eficiencia de Nash-Sutcliffe promedio (NSE) obtenida fue de 0.958684.

La comparativa entre la CMP y VVM se muestra en la siguiente tabla:

	CMP	VVM
FA	0.377483	0.055683
1/FA	2.649127	17.958895
SSP	56.64%	28.43%
NSE	0.958684	0.964045
Tiempo (s)	5850.14	6114.57
Iteraciones	1776.07	1738.70

En la tabla anterior se aprecia que el FA obtenido por la metodología VVM es casi 7 veces superior al CMP. Sin embargo, analizando los resultados obtenidos por CMP se observó una mejor consistencia. Para ejemplificar esto, en la figura VI.O se muestran los peores resultados obtenidos por ambas metodologías.

Haciendo a un lado el FA, se observa que el NSE para el mejor resultado obtenido fue de 0.999971 y el SSP fue de 78.57%. Esto es, el mejor resultado proporcionado por la metodología CMP pronosticó fugas con menos de un 25% de error.

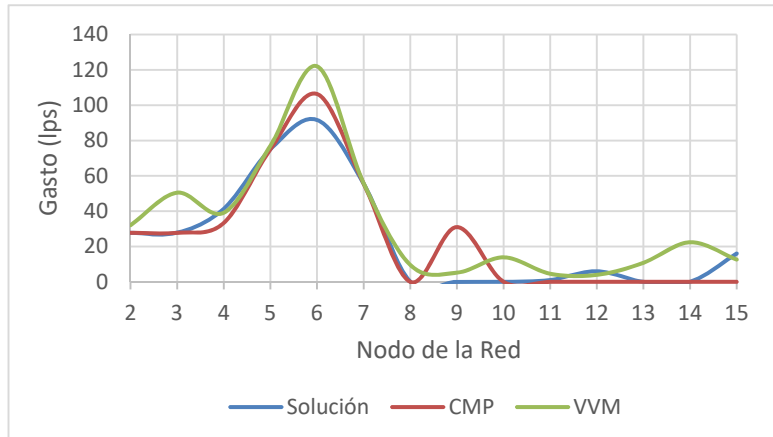


Fig. VI.O.- Peores resultados VVM contra CMP.

Haciendo un paréntesis, la metodología CMP restringe la proposición de resultados no viables por parte de la PSO. Es por tal motivo que, la CMP hace que las penalizaciones por volumen (PENVOL) y por demanda (PENDEM), incluidas en la función objetivo, carezcan de relevancia de ahora en adelante, pues siempre serán cero.

Por estas razones se decide que la metodología Clamping PSO (CMP) se desempeñó mejor y servirá de base para la próxima mejora.

Para la obtención de estos resultados se requirió la revisión de más de 14 millones de redes hidráulicas.

- En resumen, el pseudo código de CMP es el siguiente:

$V_{max} = 4$; $V_{min} = -V_{max}$

Inicializar las posiciones $X(i,j)$ aleatoriamente

for $i = 1$ until “número de partículas”

for $j=1$ until “número de nodos en la red”

$X(i,j) = (X_{max}-X_{min})*rand()+X_{min}$

end

```

end
Inicializar las velocidades  $V(i,j)$  de todas las partículas por el método
“Half-Diff”
for i = 1 until “número de partículas”
    for j=1 until “número de nodos en la red”
         $V(i,j) = 0.50 * rand(Xmax,Xmin) - X_{ij}$ 
    end
end

While iteraciones < 800
    for i = 1, until “número de partículas”
        Ejecutar la función objetivo y obtener  $FA(i,1)$ 
        if iteraciones ==1
             $FAOWNBEST(1,1) = FA(i,1)$  ; Por no tenerse más datos
             $OWNBEST(1,:)=X(i,:)$  ; Por no tenerse más datos
        else
            if  $FA(i,1) < FAOWNBEST(i,1)$ 
                 $FAOWNBEST(i,1)=FA(i,1)$ 
                 $OWNBEST(i,:)=X(i,:)$ 
                 $Pi(i,:)=OWNBEST(i,:)$  ; Mejor encontrado por partícula

                if  $FA(i,1) < \min(\min(FAOWNBEST))$ 
                     $GLOBALBEST(1,:)=X(:,1)$ 
                     $Pg(1,:) = GLOBALBEST(1,:)$  ; Mejor histórico
                    iteraciones=1 ; Se resetean las iteraciones
                end
            end
        end
    end

    if  $\min(\min(FAOWNBEST)) == 0$ 
        then “fin de ciclo”
    else

```


Calcular velocidades V_{ij} y las posiciones X_{ij} de todas las partículas de manera aleatoria.

for $i = 1$ until “número de partículas”

for $j=1$ until “número de dimensiones”

$C_1 = \text{rand}() * 2$; $C_2 = \text{rand}() * 2$;

$\Phi_1(i,j) = C_1 * \text{rand}()$; $\Phi_2(i,j) = C_2 * \text{rand}()$;

$V(i,j) = V(i,j) + \Phi_1(i,j) * [P_i(i,j) - X(i,j)]$
 $+ \Phi_2(i,j) [Pg(1,j) - X(i,j)]$

if $V(i,j) > V_{\max}$

$V(i,j) = V_{\max}$

end

if $V(i,j) < V_{\min}$ then

$V(i,j) = V_{\min}$

end

if $X(i,j) < QDEM_{ij}$

$X(i,j) = QDEM_{ij}$

$V(i,j) = 0$

else

if $X(i,j) > [QDEM_{ij} + qdif]$

$X(i,j) = QDEM_{ij} + qdif$

$V(i,j) = 0$

end

end

end

end

iteraciones = iteraciones + 1

loop

VI.2.8.) Inercia Dinámica (ID).

En la búsqueda de proveerle a la PSO una razón ideal entre exploración y explotación, se buscó la manera de limitar la velocidad máxima. Una velocidad máxima de mayor valor brinda una mejor exploración, mientras que una velocidad máxima de menor valor otorga una mejor explotación.

Durante la investigación de este tema se buscó la manera de eliminar Vmax de la PSO, fue entonces que surgió la proposición de un factor de inercia (Eberhart & Shi, 2001).

Finalmente, la condición Vmax no fue eliminada de la PSO, pero si se incluyó el uso del factor de inercia con un valor Vmax establecido en un valor dentro del rango dinámico de cada dimensión.

El factor de inercia modifica el cálculo de la velocidad de la PSO como se indica a continuación (Eberhart & Shi, 2001):

$$V_{ij(t+1)} = W_{ij} * V_{ij(t)} + \phi_1 [P_i - X_{ij(t)}] + \phi_2 [P_g - X_{ij(t)}]$$

Donde:

$V_{ij(t+1)}$ = Velocidad calculada para la siguiente iteración.

W_{ij} = Factor de inercia.

$V_{ij(t)}$ = Velocidad actual.

ϕ_1 = Confianza individual.

P_i = Mejor posición obtenida por el individuo hasta el momento.

$X_{ij(t)}$ = Posición actual.

ϕ_2 = Influencia social.

P_g = Mejor posición del grupo obtenida hasta el momento.

Maurice Clerc propuso un factor de constricción para asegurar la convergencia a la PSO (1999):

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}$$

Donde:

K = Factor de constricción

$\varphi = C_1 + C_2$; $\varphi > 4$

El factor de constricción K afecta directamente al cálculo de la velocidad V, esto es:

$$V_{ij(t+1)} = K * \{W_{ij} * V_{ij(t)} + \varphi_1 [P_i - X_{ij(t)}] + \varphi_2 [P_g - X_{ij(t)}]\}$$

Normalmente, Clerc establece que $\varphi = 4.10$, por tal motivo, el factor de constricción $K = 0.729$. Ahora bien, si C_1 y $C_2 = 2.05$, entonces:

$$\varphi_1 = 1.49445 * rand()$$

$$\varphi_2 = 1.49445 * rand()$$

Para un sistema dinámico, Eberhart & Shi (2001) recomiendan la implementación del siguiente factor de inercia:

$$W_{DS} = \left[0.50 + \frac{rand()}{2} \right]$$

Donde:

W_{DS} = Factor de inercia para sistemas dinámicos.

$rand()$ = Número aleatorio entre cero y uno.

Entonces, el factor de inercia a implementar en esta rutina será:

$$V_{ij(t+1)} = W_{DS}ij * V_{ij(t)} + \varphi_1[Pi - X_{ij(t)}] + \varphi_2[Pg - X_{ij(t)}]$$

Se realizaron 102 simulaciones con 80 partículas. La simulación con el mejor y peor resultado fueron descartadas y las 100 restantes fueron sujetas de análisis.

Los parámetros de la PSO para ID fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones de manera aleatoria dentro del espacio de búsqueda.
- Inicialización de velocidades mediante el método Half-Diff.
- Topología de estrella (gbest).
- Pg se asigna a la mejor partícula en la rutina.
- Vmax= 4
- Vmin = -Vmax
- $\varphi_1 = 1.494 * \text{rand}()$, $\varphi_2 = 1.494 * \text{rand}()$
- $W = 0.50 + \text{rand}()/2$
- $V = W * V + \varphi_1[Pi - X_{ij}] + \varphi_2[Pg - X_{ij}]$
- Clamping PSO
 - ✓ Si $X < Q_{dem} \rightarrow X = Q_{dem}$, $V = 0$;
 - ✓ Si $X > Q_{dem} + q_{dif} \rightarrow X = Q_{dem} + q_{dif}$, $V = 0$

El mejor resultado tuvo un factor de aptitud de 0.00 ($1/FA = \infty$), se requirieron 840 iteraciones para su obtención. El peor factor de aptitud fue de 1.057586 ($1/FA = 0.945550$) y se obtuvo con 854 iteraciones. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	27.78	27.78	41.34	75.00	91.67	55.56	0.07
Peor FA	27.81	27.78	64.25	84.04	91.67	57.53	0.10

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	0.00	0.01	1.00	6.00	0.00	0.02	15.95
Peor FA	0.01	0.00	0.00	0.00	0.00	0.07	0.00

El factor de aptitud promedio de la ID fue de 0.208434 ($1/FA = 4.797672$).

Se obtuvo una similitud con la solución en porcentaje promedio (SSP) de 50.07%. La eficiencia de Nash-Sutcliffe promedio (NSE) obtenida fue de 0.965073. Los valores máximos para SSP y NSE fueron 71.43% y 1.00 respectivamente.

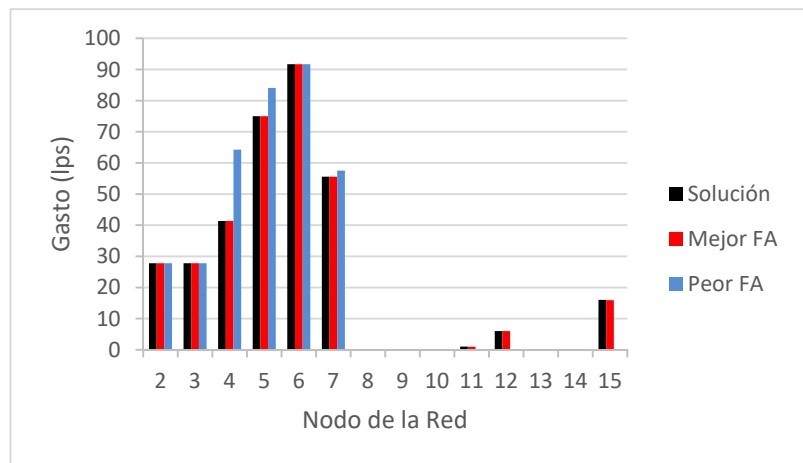


Fig. VI.P.- Resultados marginales ID contra solución.

La comparativa entre ID y CMP se muestra en la siguiente tabla:

	ID	CMP
FA	0.208434	0.377483
1/FA	4.797672	2.649127
SSP	50.07%	56.64%
NSE	0.965073	0.958684
Tiempo (s)	4217.34	5850.14
Iteraciones	1324.97	1776.07

Como se puede observar, la PSO con ID obtuvo la solución buscada. Se aprecian pequeñas diferencias entre la solución y el resultado por errores en aproximación, pero este se considera satisfactorio.

De los 100 resultados analizados, la PSO obtuvo un total de 9 redes con un FA =0, cabe mencionar que todas las redes presentaron la misma solución en un mismo número de iteraciones.

Para la obtención de estos resultados se requirió la formulación y revisión de más de 10.5 millones de redes hidráulicas.

- En resumen, el pseudo código de ID es el siguiente:

Vmax = 4 ; Vmin = -Vmax

Inicializar las posiciones X(i,j) aleatoriamente

for i = 1 until “número de partículas”

for j=1 until “número de nodos en la red”

*X(i,j) = (Xmax-Xmin)*rand()+Xmin*

end

end

Inicializar las velocidades V(i,j) de todas las partículas por el método “Half-Diff”

for i = 1 until “número de partículas”

```

    for j=1 until "número de nodos en la red"
         $V(i,j) = 0.50 * \text{rand}(X_{\max}, X_{\min}) - X_{ij}$ 
    end
end

While iteraciones < 800
    for i = 1, until "número de partículas"
        Ejecutar la función objetivo y obtener FA(i,1)
        if iteraciones == 1
            FAOWNBEST(1,1) = FA(i,1) ; Por no tenerse más datos
            OWNBEST(1,:)=X(i,:) ; Por no tenerse más datos
        else
            if FA(i,1) < FAOWNBEST(i,1)
                FAOWNBEST(i,1)=FA(i,1)
                OWNBEST(i,:)=X(i,:)
                Pi(i,:)=OWNBEST(i,:) ; Mejor encontrado por partícula

                if FA(i,1) < min(min(FAOWNBEST))
                    GLOBALBEST(1,:)=X(:,1)
                    Pg(1,:) = GLOBALBEST(1,:) ; Mejor histórico
                    iteraciones=1 ; Se resetean las iteraciones
                end
            end
        end
    end

    if min(min(FAOWNBEST)) == 0
        then "fin de ciclo"
    else
        Calcular velocidades Vij y las posiciones Xij de todas las
        partículas de manera aleatoria.
        for i = 1 until "número de partículas"
            for j=1 until "número de dimensiones"
                 $C_1 = 1.494$  ;  $C_2 = 1.494$  ;
            end
        end
    end
end

```

```

 $\Phi_1(i,j) = C_1 * rand() ; \quad \Phi_2(i,j) = C_2 * rand() ;$ 
 $W = 0.50 + rand()/2$ 
 $V(i,j) = W * V(i,j) + \Phi_1(i,j) * [ P_i(i,j) - X(i,j) ]$ 
 $\quad + \Phi_2(i,j) * [ P_g(1,j) - X(i,j) ]$ 

    if  $V(i,j) > V_{max}$ 
         $V(i,j) = V_{max}$ 
    end

    if  $V(i,j) < V_{min}$  then
         $V(i,j) = V_{min}$ 
    end

    if  $X(i,j) < QDEMIj$ 
         $X(i,j) = QDEMIj$ 
         $V(i,j) = 0$ 
    else
        if  $X(i,j) > [QDEMIj + qdif]$ 
             $X(i,j) = QDEMIj + qdif$ 
             $V(i,j) = 0$ 
        end
    end

end
end
iteraciones = iteraciones + 1
loop

```

VI.2.9.) Reinicialización y Mutación con división de trabajo (RMDL).

La modalidad RMDL propuesta por Wei & Cao (2011), descompone a la PSO en varias partes y les permite atacar el problema desde enfoques diferentes. Por su versatilidad, nivel de personalización, e incluso la posible interacción por parte del usuario en la búsqueda de soluciones,

se decidió implementar esta técnica en la rutina y observar su desempeño.

La técnica RMDL comienza realizando un cambio en la fórmula de la PSO para el cálculo de la velocidad:

$$V_{ij(t+1)} = G_1 + G_2 + G_3$$

Donde:

$V_{ij(t+1)}$ = Velocidad calculada para la siguiente iteración.

G_1 = Momento de las partículas.

G_2 = Componente Cognitiva.

G_3 = Componente Social.

El momento (G_1), hace alusión al impulso con el que cuenta una partícula gracias a su velocidad anterior. Este se define mediante:

$$G_1 = W * V_{ij}$$

Donde:

G_1 = Momento de las partículas.

W = Factor de inercia.

V_{ij} = Velocidad anterior.

La componente cognitiva hace referencia a la experiencia propia de la partícula, el mejor resultado que ha obtenido en su historia y que tan alejada se encuentra de dicho punto.

$$G_2 = C_1 * rand() * [P_i - X_{ij}]$$

Donde:

G_2 = Componente cognitiva.

C_1 = Constante personal.

P_i = Mejor resultado histórico de la partícula.

X_{ij} = Posición anterior de la partícula.

El último término (G_3) considera la experiencia del grupo durante la búsqueda, esto es:

$$G_3 = C_2 * rand() * [Pg - X_{ij}]$$

Donde:

G_3 = Componente social.

C_2 = Constante social.

P_g = Mejor resultado histórico del grupo (alcanzado por cualquier partícula).

X_{ij} = Posición anterior de la partícula.

Para la implementación de la nueva fórmula de velocidad, el enjambre se divide en 4 sub-enjambres, cada uno tendrá una forma diferente de actualizarla (Wei & Cao, 2011):

- Sub-enjambre 1: Enjambre con un factor de inercia bajo ($W=0.30$). Esto proporcionará una mejor explotación.
- Sub-enjambre 2: Se omite la velocidad anterior ($G1 = 0$), las partículas explorarán el área entre la mejor posición global y propia.
- Sub-enjambre 3: Se elimina la componente cognitiva ($G2=0$), la partícula es altamente influenciada por el grupo y carece de confianza en sus logros propios.
- Sub-enjambre 4: Se establece un alto factor de inercia ($W=0.90$), esto facilita la exploración.

Con la finalidad de evitar una convergencia prematura, se introdujo el término de reinicialización en esta técnica. Después de M iteraciones, se revisa la diversidad del enjambre (S), en caso de no cumplir con un límite

establecido “eps”, el enjambre es reinicializado aunque sus mejores históricos globales y personales son almacenados.

$$S = \frac{1}{n} \sum_{i=1}^n \sqrt{\sum_{j=1}^D (x_{ij} - \overline{x_j})^2}$$

Después de m iteraciones (brecha mutacional) sin haberse encontrado un mejor resultado global, se analiza si se deberá obtener un mejor global mutado. Al haber tenido m iteraciones sin mejora global, se genera un número aleatorio, si este es menor a Pm, se calcula y evalúa un nuevo mejor global mutado, el cual se determina mediante:

$$P'g = Pg * [1 + gaussian(\sigma)]$$

El valor de Pm puede ser del rango de 0.10 de la longitud del espacio de búsqueda, aunque se sugiere que Pm se ubique dentro de [0.10, 0.30] (Wei & Cao, 2011).

A grosso modo, el procedimiento de RMDL es el siguiente (Wei & Cao, 2011):

1. Establecer valor de M; parámetro que detonará la posibilidad de reiniciar el enjambre.
2. Definir valor de eps (Límite mínimo de variación). Determina si se reinicializa el enjambre de acuerdo a su diversidad.
3. Definir valor de m (brecha mutacional). Cada m iteraciones se propone un mejor global por Gauss.
4. Inicializar subenjambres, establecer flag=0.
5. Obtener el factor de aptitud, determinar Pi y Pg.
 - a. Si existe un nuevo Pg, entonces flag=0
 - b. De otra manera flag=flag+1

6. Actualizar las velocidades de las partículas dependiendo del subenjambre al que pertenezcan.
7. Actualizar las nuevas posiciones de las partículas.
8. Si $\text{flag-M} = 0$ y la diversidad $< \text{eps}$, reinicializar las posiciones y velocidades de las partículas
9. Si $\text{flag} > m$ y $\text{ran}() < P_m$, entonces calcular un nuevo mejor global P'_g (Gaussiano).
10. Regresar al punto 5 hasta que se cumpla condición de paro.

Para la adaptación de RMDL en esta investigación se tomaron las siguientes consideraciones:

1. Se creó un quinto subenjambre. En el artículo de Wei & Cao (2011) se mencionan 4 subenjambres, en esta investigación se generó uno adicional para considerar que este se regiría por un factor de inercia para sistemas dinámicos:

$$W_{DS} = \left[0.50 + \frac{\text{rand}()}{2} \right]$$

2. Los parámetros M, m y eps considerados fueron los siguientes:
 - $M = 50$ (Cada 50 iteraciones se buscará la reinicialización del enjambre).
 - $m = 10$ (Cada 10 iteraciones se buscará un mejor global por Gauss).
 - $\text{eps} = 9.99 \times 10^{17}$; cada M iteraciones, se reinicializará el enjambre, puesto que el valor de eps es sumamente alto. Se calcula antes la diversidad, pero esta carece de relevancia por el valor tan alto de eps.
 - $P_m = [0.10, 0.30]$. Probabilidad de que se busque un mejor global por Gauss.

3. La reinicialización de las posiciones de partículas se realizó por el método del caos (Ren & Zhong, 2011)
4. La reinicialización de velocidades se hizo por el método Half-Diff (Clerc, 2010).

La reinicialización de posiciones por el método del caos se determina mediante (Ren & Zhong, 2011):

$$X_{k+1} = L * X_k (1 - X_k)$$

Donde L es un parámetro que, cuando se encuentra entre el rango [3.56, 4.00], produce una variable caótica X. La variable X inicial debe ser un valor aleatorio entre (0,1) pero no igual a 0.25, 0.50 ni 0.75.

Los parámetros de la PSO para RMDL fueron los siguientes:

- Tamaño del enjambre = 80 partículas
- Inicialización de posiciones por el método del Caos.
- Inicialización de velocidades mediante el método Half-Diff.
- Topología de estrella (gbest).
- Pg se asigna a la mejor partícula en la rutina.
- Vmax= 4
- Vmin = -Vmax
- $\varphi_1 = 1.494 * \text{rand}()$, $\varphi_2 = 1.494 * \text{rand}()$
- $V = G1 + G2 + G3$
- División de partículas en 5 subenjambres
 - ✓ 1er subenjambre: W= 0.30
 - ✓ 2do subenjambre: W = 0.00
 - ✓ 3er subenjambre: G2=0.00
 - ✓ 4to subenjambre: W= 0.90
 - ✓ 5to subenjambre: W= 0.50 + rand()/2

- RMDL
 - ✓ $M=50$.
 - ✓ $m=10$.
 - ✓ $\text{eps} = 9.99 \times 10^{17}$.
 - ✓ $P_m = 0.30$
- Clamping PSO
 - ✓ Si $X < Q_{\text{dem}} \rightarrow X = Q_{\text{dem}}, V = 0$;
 - ✓ Si $X > Q_{\text{dem}} + q_{\text{dif}} \rightarrow X = Q_{\text{dem}} + q_{\text{dif}}, V = 0$

Se realizaron 102 simulaciones con 80 partículas. La simulación con el mejor y peor resultado fueron descartadas y las 100 restantes fueron sujetas de análisis.

El mejor resultado tuvo un factor de aptitud de 7.143×10^{-6} ($1/\text{FA} = 139,997.20$), se requirieron 2087 iteraciones para su obtención. El peor factor de aptitud fue de 0.01202143 ($1/\text{FA} = 83.1847861$) y se obtuvo con 1453 iteraciones. Los gastos obtenidos (lps) para estos casos fueron los siguientes:

Nodo	2	3	4	5	6	7	8
Solución	27.78	27.78	41.33	75.00	91.67	55.56	0.00
Mejor FA	27.78	27.78	41.54	75.00	91.67	55.83	0.00
Peor FA	27.78	27.78	33.40	75.02	91.67	57.50	31.00

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	1.00	6.00	0.00	0.00	16.00
Mejor FA	0.00	0.00	1.00	6.00	0.00	0.00	15.57
Peor FA	0.00	0.01	0.00	6.07	0.35	0.00	14.11

El factor de aptitud promedio de la RMDL fue de 0.001551714 ($1/\text{FA} = 644.45$).

Se obtuvo una similitud con la solución en porcentaje promedio (SSP) de 61.64%. La eficiencia de Nash-Sutcliffe promedio (NSE) obtenida fue de 0.986811. Los valores máximos para SSP y NSE fueron 92.86% y 1.00 respectivamente.

La comparativa entre RMDL e ID se muestra en la siguiente tabla:

	RMDL	ID
FA	0.001551714	0.208434
1/FA	644.45	4.797672
SSP	61.64%	50.07%
NSE	0.986811	0.965073
Tiempo (s)	7832.27	4217.34
Iteraciones	2335.89	1324.97

A pesar de que la metodología RMDL no obtuvo ninguna solución exacta ($FA=0$), la obtención de mejores factores de aptitud en el universo de soluciones fue mayor. También se puede observar que, aunque no se obtuvo el volumen exacto de fugas, si se determinó su localización.

Esta metodología permite la inclusión de múltiples subenjambrs en la solución de un problema y cada uno analiza el problema de diferente forma. Esta característica hace pensar que, más que una modalidad de la PSO, se trata de una modalidad de la Agent Swarm Optimization (ASO).

La consistencia en la obtención de resultados y versatilidad de esta metodología son los principales puntos fuertes a considerar. Sería prudente el realizar futuras investigaciones en las que un subenjambre fuera operado por el usuario y que cada n número de iteraciones, el usuario pudiera determinar por donde se debe explorar.

Para la obtención de estos resultados se requirió la formulación y revisión de más de 18.6 millones de redes hidráulicas.

- En resumen, el pseudo código de RMDL es el siguiente:

```

Vmax = 4 ; Vmin = -Vmax
Dividir número de partículas en 5 subenjambres
Inicializar las posiciones X(i,j) por el método del Caos
for i = 1 until "número de partículas"
    for j=1 until "número de nodos en la red"
        if i =1 then Fij=0.25
            while Fij = 0.25 || Fij=0.50 || Fij=0.75
                Fij = rand()
            end
        else
            Lij = rand()*(4-3.56)+3.56
            Fij = Lij*Fij*(1-Fi(j-1))
        end
        Xij =Fij*qdif+QDEM(i,j)
    end
end
Inicializar las velocidades V(i,j) de todas las partículas por el método
"Half-Diff"
for i = 1 until "número de partículas"
    for j=1 until "número de nodos en la red"
        V(i,j) = 0.50*rand(Xmax,Xmin)-Xij
    end
end
Asignando variables RMDL
M= 50
Pm= 0.30
m=10
eps=9.99X1017

```



```

flag = 0
iteraciones=0
While iteraciones < 800
    for i = 1, until "número de partículas"
        Ejecutar la función objetivo y obtener FA(i,1)
        if iteraciones ==1
            FAOWNBEST(1,1) = FA(i,1) ; Por no tenerse más datos
            OWNBEST(1,:)=X(i,:) ; Por no tenerse más datos
        else
            if FA(i,1) < FAOWNBEST(i,1)
                FAOWNBEST(i,1)=FA(i,1)
                OWNBEST(i,:)=X(i,:)
                Pi(i,:)=OWNBEST(i,:) ; Mejor encontrado por partícula

                if FA(i,1) < min(min(FAOWNBEST))
                    GLOBALBEST(1,:)=X(:,1)
                    Pg(1,:) = GLOBALBEST(1,:) ; Mejor histórico
                    iteraciones=1 ; Se resetean las iteraciones
                    flag = 0
                end
            end
        end
    end

    if min(min(FAOWNBEST)) ==0
        then "fin de ciclo"
    else
        Calcular velocidades Vij y las posiciones Xij de todas las
        partículas de manera aleatoria.
        for i = 1 until "número de partículas"
            for j=1 until "número de dimensiones"
                C1 = 1.494 ; C2 = 1.494 ;
                Φ1(i,j) = C1*rand() ;      Φ2(i,j) = C2*rand() ;
                G1 = 0.30*V(i,j)
            end
        end
    end
end

```

$$G2 = \phi_1(i,j) * [Pi(i,j) - X(i,j)]$$

$$G3 = \phi_2(i,j) [Pg(1,j) - X(i,j)]$$

Para el subenjambre 2

$$G1 = 0.00$$

Para el subenjambre 3

$$G2=0.00$$

Para el subenjambre 4

$$G1 = 0.90 * V(i,j)$$

Para el subenjambre 5

$$G1 = 0.50 + \text{rand}()/2$$

$$V(i,j) = G1 + G2 + G3$$

if $V(i,j) > V_{\max}$

$$V(i,j) = V_{\max}$$

end

if $V(i,j) < V_{\min}$ *then*

$$V(i,j) = V_{\min}$$

end

if $X(i,j) < QDEMIj$

$$X(i,j) = QDEMIj$$

$$V(i,j) = 0$$

else

if $X(i,j) > [QDEMIj + qdif]$

$$X(i,j) = QDEMIj + qdif$$

$$V(i,j) = 0$$

end

end

end

end

if flag= M

Calcular la diversidad (S) para los subenjambres

$$S = \frac{1}{n} \sum_{i=1}^n \sqrt{\sum_{j=1}^D (x_{ij} - \overline{x_j})^2}$$

If S<eps then reiniciar posiciones por método del caos y velocidades por half-diff

end

if flag=m

if rand()<Pm

Determinar un mejor global con:

$$P'g = Pg * [1 + \text{gaussian}(\sigma)]$$

if FA(P'g)<FA(Pg), then Pg = P'g

end

end

iteraciones = iteraciones +1

flag = flag + 1

loop

VI.3) Modelado en otras redes.

Habiendo determinado que las metodologías Reinicialización y Mutación con división de trabajo (RMDL) e Inercia Dinámica (ID) fueron las de mejor desempeño en esta investigación, se procedió a ponerlas a prueba en otras redes. No se volverán a mencionar los parámetros ni el funcionamiento de dichas metodologías; en caso de requerirse más información refiérase a los subcapítulos VI.2.8 y VI.2.9.

Las redes en las que se puso en prueba las metodologías RMDL e ID fueron las siguientes:

- Red 2 a Red 5

Todas estas redes son modificaciones de la Red 1 (Ver subcapítulo VI.2). La geometría, distancias, diámetros y características hidráulicas son idénticas a la red 1. Para su modelado, solamente se modificaron el volumen, el número y la ubicación de fugas.

- Red 2.

El gasto de entrada a la red es de 408 lps, el volumen total de fuga es de 96.88 lps, el cual representa el 23.75%. Se crearon 2 puntos de fuga: 3.88 lps en el nodo 2 y 93 lps en el nodo 14.

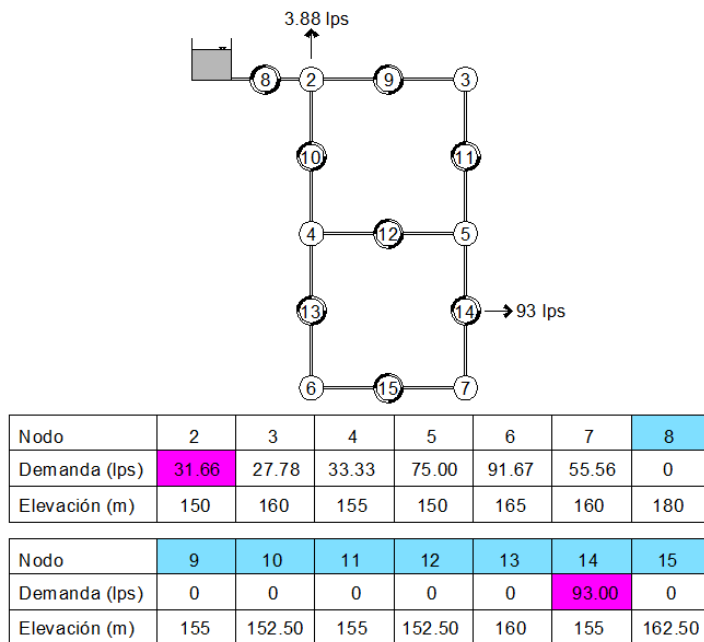


Fig. VI.Q.- Red 2.

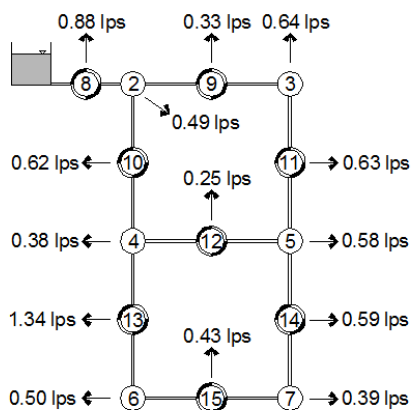
El espacio de búsqueda definido para cada dimensión en esta red es el siguiente:

Nodo	2	3	4	5	6	7	8
Esp. Busq. Min	27.78	27.78	33.33	75.00	91.67	55.56	0
Esp. Busq. Max	124.66	124.66	130.21	171.88	188.55	152.44	96.88

Nodo	9	10	11	12	13	14	15
Esp. Busq. Min	0	0	0	0	0	0	0
Esp. Busq. Max	96.88	96.88	96.88	96.88	96.88	189.88	96.88

- Red 3.

El gasto de entrada a la red es de 319.164 lps, el volumen total de fuga es de 8.044 lps, el cual representa el 2.52%. Se crearon 14 puntos de fuga:



Nodo	2	3	4	5	6	7	8
Demanda (lps)	28.274	28.422	33.712	75.580	92.173	55.949	0.880
Elevación (m)	150	160	155	150	165	160	180

Nodo	9	10	11	12	13	14	15
Demanda (lps)	0.325	0.620	0.625	0.251	1.335	0.588	0.430
Elevación (m)	155	152.50	155	152.50	160	155	162.50

Fig. VI.R.- Red 3.

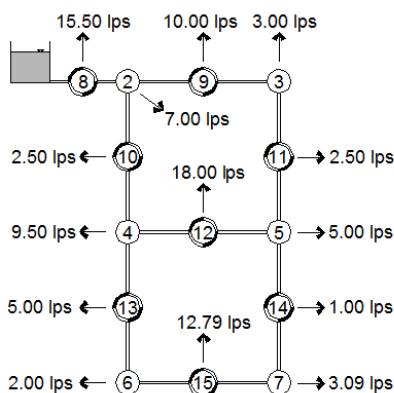
El espacio de búsqueda definido para cada dimensión en esta red es el siguiente:

Nodo	2	3	4	5	6	7	8
Esp. Busq. Min	27.78	27.78	33.33	75.00	91.67	55.56	0
Esp. Busq. Max	35.284	35.284	41.374	83.044	99.714	63.604	8.044

Nodo	9	10	11	12	13	14	15
Esp. Busq. Min	0	0	0	0	0	0	0
Esp. Busq. Max	8.044	8.044	8.044	8.044	8.044	8.044	8.044

- Red 4.

El gasto de entrada a la red es de 408 lps, el volumen total de fuga es de 96.88 lps, el cual representa el 23.75%. Se crearon 14 puntos de fuga:



Nodo	2	3	4	5	6	7	8
Demanda (lps)	34.78	30.78	42.83	80.00	93.67	58.65	15.50
Elevación (m)	150	160	155	150	165	160	180

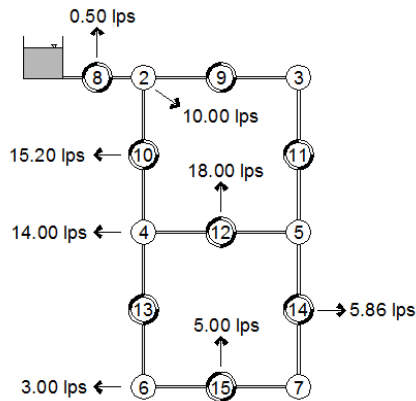
Nodo	9	10	11	12	13	14	15
Demanda (lps)	10.00	2.50	2.50	18.00	5.00	1.00	12.79
Elevación (m)	155	152.50	155	152.50	160	155	162.50

Fig. VI.S.- Red 4.

Nodo	2	3	4	5	6	7	8
Esp. Busq. Min	27.78	27.78	33.33	75.00	91.67	55.56	0
Esp. Busq. Max	124.66	124.66	130.21	171.88	188.55	152.44	96.88

Nodo	9	10	11	12	13	14	15
Esp. Busq. Min	0	0	0	0	0	0	0
Esp. Busq. Max	96.88	96.88	96.88	96.88	96.88	96.88	96.88

El gasto de entrada a la red es de 382.68 lps, el volumen total de fuga es de 71.56 lps, el cual representa el 18.77%. Se crearon 8 puntos de fuga:



Nodo	2	3	4	5	6	7	8
Demanda (lps)	37.78	27.78	47.33	75.00	94.67	55.56	0.50
Elevación (m)	150	160	155	150	165	160	180

Nodo	9	10	11	12	13	14	15
Demanda (lps)	0.00	15.20	0.00	18.00	0.00	5.86	5.00
Elevación (m)	155	152.50	155	152.50	160	155	162.50

[Cap. VI] Análisis y Modelado | 201

El espacio de búsqueda definido para cada dimensión en esta red es el siguiente:

Nodo	2	3	4	5	6	7	8
Esp. Busq. Min	27.78	27.78	33.33	75.00	91.67	55.56	0
Esp. Busq. Max	99.34	99.34	104.89	146.56	163.23	127.12	71.56

Nodo	9	10	11	12	13	14	15
Esp. Busq. Min	0	0	0	0	0	0	0
Esp. Busq. Max	71.56	71.56	71.56	71.56	71.56	71.56	71.56

VI.3.1) Resultados Obtenidos.

Las metodologías ID y RMDL se corrieron en 102 ocasiones, descartando la que obtuvo el mejor y peor factor de aptitud y sometiendo los otros 100 resultados a análisis.

- Red 2

Se detallan los resultados encontrados por ambas rutinas:

	ID	RMDL
Mejor FA	0	0
Peor FA	27.98971	0.812178
FA Promedio	3.60608	0.024414
Mejor SSP	100.00%	100.00%
SSP Promedio	72.57%	70.07%
Mejor NSE	1	1
NSE Promedio	0.639083	0.980904
Tiempo (s)	3004.17	5055.92
Iteraciones	890.13	1592.74
Veces que se alcanzó FA = 0	18	9

Las mejores configuraciones de gastos obtenidas fueron:

Nodo	2	3	4	5	6	7	8
Solución	31.66	27.78	33.33	75.00	91.67	55.56	0.00
ID	31.65	27.78	33.33	75.00	91.67	55.56	0.00
RMDL	31.67	27.78	33.33	75.00	91.67	55.56	0.00

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	0.00	0.00	0.00	93.00	0.00
ID	0.00	0.00	0.00	0.00	0.00	92.99	0.02
RMDL	0.00	0.00	0.00	0.00	0.01	93.00	0.00

Las peores configuraciones de gastos obtenidas son:

Nodo	2	3	4	5	6	7	8
Solución	31.66	27.78	33.33	75.00	91.67	55.56	0.00
ID	27.78	27.78	33.33	75.00	91.67	55.56	0.02
RMDL	27.78	27.78	33.33	75.00	91.67	55.56	96.88

Nodo	9	10	11	12	13	14	15
Solución	0.00	0.00	0.00	0.00	0.00	93.00	0.00
ID	49.32	0.00	0.00	0.00	0.00	0.00	96.88
RMDL	0.00	0.00	0.00	0.00	0.00	86.22	0.00

- Red 3

Se detallan los resultados encontrados por ambas rutinas:

	ID	RMDL
Mejor FA	1.43×10^{-5}	2.14×10^{-5}
Peor FA	0.07180	0.00159
FA Promedio	0.002685	4.67×10^{-4}
Mejor SSP	92.86 %	85.71%
SSP Promedio	62.07 %	63.72%
Mejor NSE	1.00	1.00
NSE Promedio	0.998171	0.998711
Tiempo (s)	3734.92	3955.96
Iteraciones	1185.74	1239.01
Veces que se alcanzó FA = 0	0	0

Las mejores configuraciones de gastos obtenidas fueron:

Nodo	2	3	4	5	6	7	8
Solución	28.274	28.422	33.712	75.58	92.173	55.949	0.88
ID	28.46	28.56	33.33	75.44	91.73	55.62	2.46
RMDL	30.06	28.56	34.09	75.42	93.29	56.12	0.03

Nodo	9	10	11	12	13	14	15
Solución	0.325	0.62	0.625	0.251	1.335	0.588	0.43
ID	0.09	0.00	0.66	0.26	1.97	0.74	0.85
RMDL	0.02	0.00	0.66	0.250	0.00	0.70	0.00

Las peores configuraciones de gastos obtenidas son:

Nodo	2	3	4	5	6	7	8
Solución	28.274	28.422	33.712	75.58	92.173	55.949	0.88
ID	35.82	27.78	33.33	75.00	91.67	55.64	0.02
RMDL	35.82	27.78	33.33	76.69	91.67	55.56	0.00

Nodo	9	10	11	12	13	14	15
Solución	0.325	0.62	0.625	0.251	1.335	0.588	0.43
ID	8.04	0.00	0.00	0.00	0.00	0.00	0.00
RMDL	0.35	0.02	0.00	0.19	0.00	0.00	1.25

- Red 4

Se detallan los resultados encontrados por ambas rutinas:

	ID	RMDL
Mejor FA	0.074557	0.000478
Peor FA	9.780329	0.011443
FA Promedio	6.093368	0.016475
Mejor SSP	64.29%	92.86%
SSP Promedio	38.07%	74.28%
Mejor NSE	0.948450	0.993301
NSE Promedio	0.679626	0.877770
Tiempo (s)	4855.92	8419.42
Iteraciones	1537.92	2597.53
Veces que se alcanzó FA = 0	0	0

Las mejores configuraciones de gastos obtenidas fueron:

Nodo	2	3	4	5	6	7	8
Solución	34.78	30.78	42.83	80.00	93.67	58.65	15.50
ID	39.10	41.94	35.98	75.00	91.67	69.87	11.58
RMDL	38.97	30.62	45.67	80.77	91.67	59.86	9.26

Nodo	9	10	11	12	13	14	15
Solución	10.00	2.50	2.50	18.00	5.00	1.00	12.79
ID	0.00	12.09	0.00	18.96	5.13	0.00	6.75
RMDL	10.05	0.00	2.37	17.91	5.18	0.00	12.97

Las peores configuraciones de gastos obtenidas son:

Nodo	2	3	4	5	6	7	8
Solución	34.78	30.78	42.83	80.00	93.67	58.65	15.50
ID	27.78	27.78	33.33	75.00	91.67	55.56	0.00
RMDL	27.78	31.74	33.33	75.00	91.67	55.56	19.58

Nodo	9	10	11	12	13	14	15
Solución	10.00	2.50	2.50	18.00	5.00	1.00	12.79
ID	29.14	48.44	0.00	0.00	0.00	0.00	40.11
RMDL	7.94	20.17	4.41	18.43	4.03	5.16	16.43

- Red 5

Se detallan los resultados encontrados por ambas rutinas:

	ID	RMDL
Mejor FA	0.000386	0.000150
Peor FA	8.246214	0.039407
FA Promedio	3.484938	0.002904
Mejor SSP	92.86%	92.86%
SSP Promedio	57.29%	70.50%
Mejor NSE	0.992752	0.992646
NSE Promedio	0.839908	0.940758
Tiempo (s)	4241.94	8690.87
Iteraciones	1296.83	2762.91
Veces que se alcanzó FA = 0	0	0

Las mejores configuraciones de gastos obtenidas fueron:

Nodo	2	3	4	5	6	7	8
Solución	37.78	27.78	47.33	75.00	94.67	55.56	0.50
ID	27.78	28.11	41.85	75.00	91.67	55.56	7.78
RMDL	32.19	27.78	44.46	75.00	91.68	55.56	3.20

Nodo	9	10	11	12	13	14	15
Solución	0.00	15.20	0.00	18.00	0.00	5.86	5.00
ID	0.01	22.62	0.00	18.00	5.17	5.89	5.23
RMDL	0.32	19.61	0.00	18.00	3.83	5.86	5.53

Las peores configuraciones de gastos obtenidas son:

Nodo	2	3	4	5	6	7	8
Solución	37.78	27.78	47.33	75.00	94.67	55.56	0.50
ID	27.78	27.78	33.33	75.00	91.71	55.56	19.40
RMDL	27.81	27.78	33.33	75.00	91.67	60.09	71.56

Nodo	9	10	11	12	13	14	15
Solución	0.00	15.20	0.00	18.00	0.00	5.86	5.00
ID	0.00	61.97	0.00	0.00	0.00	0.00	29.84
RMDL	0.00	0.03	0.00	18.08	13.72	4.87	0.00

CAPÍTULO VII

>>> CONCLUSIONES, RECOMENDACIONES Y LIMITACIONES

*“La vida es el arte de sacar conclusiones suficientes
a partir de datos insuficientes”*

-Samuel Butler

El desarrollo de técnicas computacionales inspiradas en procesos biológicos brinda una pauta novedosa y práctica para la solución de problemas aplicados en ingeniería.

Durante el desarrollo de esta investigación se revisaron más de 194 millones de redes hidráulicas, se analizaron diversas modificaciones a la PSO y se pudo observar que, si bien en un principio no se obtuvieron resultados idóneos, con la implementación de nuevas técnicas se logró determinar la ubicación y magnitud de diversas fugas en una red de distribución de agua potable.

A pesar de que se obtuvieron varios resultados exactos, se puede apreciar que la calibración de la PSO depende directamente de las características del problema en sí. Cuando la red tenía gastos de fuga de poca magnitud y en pocos nodos, estas se pudieron detectar con bastante precisión. Sin embargo, al incrementar el número de nodos con fuga no se obtuvieron resultados exactos.

El método propuesto es poderoso, económico y se considera que su potencial no ha sido completamente desarrollado. El código propuesto es amigable, sencillo, escalable y fácil de implementar en cualquier escenario de fuga.

La publicación de todos los cálculos realizados por la rutina en su proceso de búsqueda le facilitan al usuario la comprensión de la metodología, así como la generación de ideas en su proceso de búsqueda.

La visualización del vuelo de la PSO le brinda al usuario la posibilidad de observar cómo se comportan las partículas, la comunicación entre enjambres, los patrones cíclicos que a veces se presentan y qué factores son los que lo generan.

Las principales limitaciones que se tuvieron durante el desarrollo de esta tesis fueron las siguientes:

- La modelación de los puntos de fuga como una demanda: Este es el principal punto a reforzar en esta investigación. Una fuga depende directamente de la presión y, esta no solo genera un mayor gasto, sino hasta una mayor área de fuga. En esta investigación se consideró que la fuga es constante sin importar la variación en la presión.
- La aplicación de esta metodología en un caso real: A pesar de haberse considerado su aplicación, no se tuvieron los elementos necesarios para cotejar la modelación matemática con un problema real de aplicación.
- Simulación en otros escenarios: Por razones de tiempo no se pudo modelar esta rutina en otros escenarios para observar su comportamiento. Se generaron 5 diversos escenarios para observar

el desempeño del algoritmo, hubiera sido deseable la simulación en otras redes con diferente arreglo de fugas.

Dentro de las recomendaciones para un trabajo a futuro se indican las siguientes:

- Se considera ampliamente recomendable la participación del usuario en la búsqueda de un mejor global. Sería ideal el que el usuario pueda indicarle a la PSO un camino a tomar como si se tratara del líder de un subenjambre. El usuario podría, mediante una interfaz gráfica, indicarle al enjambre la exploración de áreas no consideradas anteriormente por la PSO.
- Implementación de la PSO en otro lenguaje de programación. A pesar de que el Matlab es muy amigable y poderoso, se recomienda su migración a bases de programación de mayor alcance como el C++ o C#. El Matlab presenta problemas en su desempeño al graficar el vuelo de la PSO.
- Se observó que el rendimiento de la PSO se incrementó en redes con menor número de fugas. Por tal motivo, se vislumbra la posibilidad de segmentar una gran red en pequeños tramos y analizar cada uno de manera individual.
- Otra posible mejora puede ser el cambiar el FA de la siguiente forma:

$$FA = \frac{1}{n} \sum_{i=1}^n (P_{TEOi} - P_{PSOi})^2 + 9 \times 10^5 \text{ PENFUG}$$

Donde:

FA = Factor de aptitud.

n = Número de nodos evaluados.

P_{TEOi} = Presión teórica (Objetivo) en el nodo i .

P_{PSOi} = Presión obtenida con EPANET con base a los gastos propuestos por la PSO.

DIFUG = Penalización por fuga. Obtiene la diferencia entre el volumen total suministrado menos el volumen total propuesto por la PSO.

$$PENFUG = \frac{1}{n} \sum_{i=1}^n (V_{t_{REAL}} - V_{t_{PSO}})^2$$

Donde:

$V_{t_{REAL}}$ = Volumen total suministrado por el organismo operador.

$V_{t_{PSO}}$ = Suma de todas las demandas propuestas por la PSO.

Al implementar la Clamping PSO, los términos PENVOL y PENDEM del factor de aptitud anterior carecen de relevancia. Se considera que la sustitución de estos por una penalización por obtener un volumen de fuga diferente al suministrado ayudará a refinar el espacio de búsqueda.

>>> REFERENCIAS

“No es preciso tener muchos libros, sino tenerlos buenos...”

-Séneca

**** Capítulo I.- Introducción. ****

Belsito, S., Lombardi, P., Andreussi, P. and Banerjee, S. (1998), **“Leak detection in liquefied gas pipelines by artificial neural networks”**. AIChE J., Vol. 44., No.12., Pp. 2675–2688. doi: 10.1002/aic.690441209

Burn S., De Silva D., Eiswirth M., Hunaindi O., Speers A., Thornton J. (1999) **“Pipe Leakage – future challenges and solutions”** Pipes Conference, Wagga Wagga, N.S.W., 12-14.

CONAPO (2007). **“Proyecciones de la población de México 2005-2050”**. México.

De Silva D., Mashford, Burn S. (2011) **“Computer aided location and sizing in pipe networks”**. Urban Water Security Research Alliance Technical Report No 17. CSIRO. ISSN 1836-5558.

Esquivel J., Tello J., Martínez R., Marchand J., Balderas E. (2010). **“Aspectos Teóricos y Prácticos en la solución del problema de Detección de Fugas en Ductos de Líquidos”**. Congreso Anual de la Asociación de México de Control Automático. Puerto Vallarta, Jalisco, México.

FAO (2011). **“Information System on Water and Agriculture, Aquastat”**.
<http://www.fao.org/nr/water/aquastat/data/query/index.html?lang=es>
(07/04/2011).

Fuentes-Mariles O., Rodríguez-Vázquez K., Palma-Nava A. (2010).
“Estimación y localización de fugas en una red de tuberías de agua potable usando algoritmos genéticos” Ingeniería Investigación y Tecnología. V. XII, Núm. 2, 2011, Pp. 235-242. ISSN 1405-7743 FI-UNAM

Gama L., Sánchez M., Ochoa C (2010). **“Diseño de una interfaz para la detección de fugas de agua”**. Revista Digital Universitaria [en línea]. V.11, No.2 ISSN: 1607-6079.

Geiger, G. (2008). **“Principles of Leak Detection. 1st Edition (September 2008)”**. Krohne Oil & Gas, Breda (The Netherlands).

IMTA (2010). **“El agua se pierde 43.2% en fugas: ONU”**.
<http://www.imta.gob.mx/internacional/index.php/noticias-internacionales/noticias-del-sector-hidrico-internacional/243-el-agua-se-pierde-432-en-fugas-onu.html>

INEGI (2010). **“Volumen y Crecimiento. Población total en zona urbana por países seleccionados, 1950 a 2010”**.
<http://www3.inegi.org.mx/sistemas/sisept/Default.aspx?t=mdemo114&s=est&c=23643>

Izquierdo J., Montalvo I., Pérez-García R., Alonso C. (2010). **“Swarm Intelligence for Optimization in the Urban Water Industry, Modelling Simulation and Optimization”**, Gregorio Romero Rey and Luisa Martinez Muneta (Ed.), ISBN: 978-953-307-048-3.

Kingdom B., Liemberger R., Marin P. (2006) **“The Challenge of reducing non-revenue wáter in developing countries”**. Water Supply and

Sanitation Sector Board Discussion Paper Series. The World Bank Group. No 8.

Montalvo I., Izquierdo J., Pérez-García R., Herrera M. (2010). **“Improved performance of PSO with self-adaptive parameters for computing the optimal design of Water Supply Systems”**. Engineering Applications of Artificial Intelligence, No. 23, Pp. 727–735. doi: 10.1016/j.engappai.2010.01.015.

Sosa F. (2010). **“Exploring the risks of ineffective water supply and sewage disposal: A case of study of Mexico City”**. Environmental Hazards. V.9. Issue 2. Pp. 135-146. doi: 10.3763/ehaz.2010.0016

Zhang (1996). **“Designing a Cost Effective and Reliable Pipeline Leak Detection System”**. Pipeline Reliability Conference, Houston, USA, November 19-22.

****** Capítulo II.- Generalidades del Tema. ******

Billava N., Kulkarni A. (2012) **“Status and Role of Gram Panchayats in Water Supply and Sanitation in Rural Area: An Empirical Study from Karnataka”** International Journal in Multidisciplinary and Academic Research (SSIJMAR). Vol. 1, No. 4, November-December (ISSN 2278 – 5973).

Burn S., De Silva D., Eiswirth M., Hunaindi O., Speers A., Thornton J. (1999) **“Pipe Leakage – future challenges and solutions”** Pipes Conference, Wagga Wagga, N.S.W., 12-14.

CNA - Comisión Nacional del Agua (2007), **“Manual de Agua Potable, Alcantarillado y Saneamiento”** Tomos: Redes de Distribución, Obras de Toma. Edición 2007. ISBN: 978-968-817-880-5.

CNA – Comisión Nacional del Agua (2011), **“Situación del Subsector Agua Potable, Alcantarillado y Saneamiento”**. Edición 2011

Farrer H. (1979). **“Control de fugas en redes de distribución de agua”**. Taller para ingenieros sobre operación y mantenimiento de sistemas de abastecimiento de agua potable y alcantarillado. Lima-Perú; CEPIS; Octubre 1979. 30 p.

Fuentes-Mariles O., Rodríguez-Vázquez K., Palma-Nava A. (2010). **“Estimación y localización de fugas en una red de tuberías de agua potable usando algoritmos genéticos”** Ingeniería Investigación y Tecnología. V. XII, Núm. 2, 2011, Pp. 235-242. ISSN 1405-7743 FI-UNAM

Gama L., Sánchez M., Ochoa C (2010). **“Diseño de una interfaz para la detección de fugas de agua”**. Revista Digital Universitaria [en línea]. V.11, No.2 ISSN: 1607-6079.

Madrigal J., Soriano N., Arista G. (2012), **“Agua potable, disponibilidad en México, estrés hídrico, aportaciones a la vivienda urbana”**, Ide@s CONCYTEG, 7 (86), pp. 1001-1020. ISSN: 2007-2716.

Mays L., Koutsoyiannis D., Angelakis A. (2007). **“A Brief History of Urban Water Supply in the Antiquity”**. Water Science & Technology: Water Supply. Vol.7, No.1, Pp.1–12, IWA Publishing 2007. doi:10.2166/ws.2007.001

Mora-Rodríguez J. (2011). **“Modelación de la intrusión patógena a través de fallos y su implicación en la red de distribución de agua”**. Tesis Doctoral. Universidad Politécnica de Valencia.

Tafuri A., Yezzi J., Watts D., Carlyle J. (1997). **“Leak detection and leak location in underground pipelines”**. International oil spill conference proceedings. April 1997, Vol. 1997, No. 1, pp. 379-381. doi: 10.7901/2169-3358-1997-1-379

Tello L. (2007). **“El acceso al agua potable en México”**. Instituto de Investigaciones Jurídicas de la UNAM. Revista del Centro Nacional de Derechos Humanos, Número 4, Sección de Artículos.

****** Capítulo III.- Particle Swarm Optimization (PSO). ******

Bai Q. (2010) **“Analysis of Particle Swarm Optimization Algorithm”**. Computer and Information Science. V.3., No.1. Pp. 180-184. ISSN 1913-8997 (Online)

Barrera J., Coello C. (2009). **“Limiting the velocity in particle swarm optimization using a geometric series”**. Proceedings of the 11th Annual conference on Genetic and evolutionary computation. Pp. 1739-1740. doi: 10.1145/1569901.1570135.

Bratton D., Kennedy J. (2007). **“Defining a Standard for Particle Swarm Optimization”**. Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007). Pp. 120-127. doi: 10.1109/SIS.2007.368035

Cai X., Cui Z., Zeng J., Tan Y. (2009). **“Individual Parameter Selection Strategy for Particle Swarm Optimization”**, Particle Swarm Optimization. Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0

Clerc M. (2010). **“From Theory to Practice in Particle Swarm Optimization”**. Handbook of Swarm Intelligence. Adaptation, Learning, and Optimization Volume 8, 2010, pp 3-36. doi: 10.1007/978-3-642-17390-5_1

Corcoran K., Cransius J., Mussweiler T. (2011). **“Social comparison: Motives, standards, and mechanisms”**. Theories in social psychology First Edition. Pp. 119-139. Blackwell Publishing.

Cui Z., Zeng J. (2005). **“A Modified Particle Swarm Optimization Predicted by Velocity”**. Proceedings of the 2005 conference on Genetic and evolutionary computation. Pp. 277-278. doi: 10.1145/1068009.1068054

Dörr A., Banz C. (2010). **“Ventanas de oportunidad y capacidad de establecer vínculo afectivo en niños ferales v/s niños abandonados en instituciones”** Gaceta de Psiquiatría Universitaria, Facultad de Medicina, Universidad de Chile. V.6. Pp. 38-45. ISN: 0718-1981 (Versión en línea).

Eberhart R., Shi Y. (2001). **“Particle Swarm Optimization: Developments, Applications and Resources”**. Proceedings of the Congress on Evolutionary Computation. Vol. 1, Pp. 81-86, doi: 10.1109/CEC.2001.934374

Engeström Y., Sannino A. (2012). **“Whatever happened to process theories of learning?”** Learning, Culture and Social Interaction. V.1. Pp. 45-56. doi: 10.1016/j.lcsi.2012.03.002

Gutiérrez A., Lanza M., Barriuso I., Valle L., Domingo M., Pérez J., Batrechea J. (2011). **“Comparison of different PSO initialization techniques for high dimensional search space problems: A test with FSS and antenna arrays”**. Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP). April 2011. Pp. 965 – 969.

Helwig S., Neumann F., Wanka R. (2009). **“Particle Swarm Optimization with Velocity Adaptation”**. International Conference on Adaptive and Intelligent Systems. Pp. 146-151. doi: 10.1109/ICAIS.2009.32

Hu X., Eberhart R. (2002). **“Multiobjective optimization using dynamic neighborhood particle swarm optimization”**. Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02. Vol. 2. Pp. 1677 – 1681. doi: 10.1109/CEC.2002.1004494

Kennedy J., Eberhart R. (1995). **“Particle Swarm Optimization”**. Proceedings of IEEE International Conference on Neural Networks. V.4, Pp. 1942-1948. Perth, Australia, 1995. doi: 10.1109/ICNN.1995.488968.

Ma G., Zhou W., Chang X. (2012). **“A novel particle swarm optimization algorithm based on particle migration”**. Applied Mathematics and Computation Vol. 218, No. 11. Pp. 6620–6626. doi: 10.1016/j.amc.2011.12.032

Mendes R., Kennedy J., Neves J. (2004). **“The Fully Informed Particle Swarm: Simpler, Maybe Better”**. IEEE Transactions on Evolutionary Computation. Vol.8, No. 3. doi: 10.1109/TEVC.2004.826074

Miner D., Pickett M., DesJardins M. (2009). **“Understanding the brain’s emergent properties”**. Advances in Intelligent Systems Research. doi: 10.2991/agi.2009.4

Kennedy J., Eberhart R. (2001). **“Swarm Intelligence”** Morgan Kaufmann Publishers. ISBN-13: 978-1558605954. Pp. 512

Parsopoulos K., Vrahatis M. (2008). **“Multi-Objective Particles Swarm Optimization Approaches”**. Multi-Objective Optimization in Computational Intelligence: Theory and Practice. doi: 10.4018/978-1-59904-498-9.ch002

Ratnaweera A., Halgamuge S., Watson H. (2004). **“Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients”**. IEEE Transactions on Evolutionary Computation. Vol.8, No. 3, Pp. 240-255. doi: 10.1109/TEVC.2004.826071

Ren B., Zhong W. (2011). **“Multi-objective Optimization using Chaos Based PSO”**. Information Technology Journal 2011. Vol. 10, No. 10, Pp. 1908-1916. doi: 10.3923/itj.2011.1908.1916

Richards M., Ventura D. (2004). **“Choosing a starting configuration for particle swarm optimization”**. Proceedings IEEE International Joint Conference on Neural Networks, 2004. Vol.3. Pp. 2309-2312. doi: 10.1109/IJCNN.2004.1380986

Romanycia M., Pelletier F. (1985). **“What is a heuristic?”** Computational Intelligence. V.1. Pp. 47-58. doi: 10.1111/j.1467-8640.1985.tb00058.x

Wei P., Cao D., (2011) **“Particle Swarm Optimization with Division of Labor”**. Ministry of Education of the People’s Republic of China. http://www.paper.edu.cn/index.php/default/en_releasepaper/downPaper/201104-488

Wong T., Luk W., Heng P. (1997). **“Sampling with Hammersley and Halton Points”**. Journal of Graphics Tools. Vol. 2, No. 2, Pp. 9-24. doi: 10.1080/10867651.1997.10487471

Zhang L., Yu H., Hu S. (2005) **“Optimal choice of parameters for particle swarm optimization”**. Journal of Zhejiang University Science A. June 2005, Volume 6, Issue 6, pp 528-534. doi: 10.1631/jzus.2005.A0528

**** Capítulo IV.- EPANET. ****

Cross H. (1936). **“Analysis of flow in networks of conduits or conductors”** University of Illinois at Urbana Champaign, College of Engineering. Engineering Experiment Station. Bulletin no. 286

Giles R. (1994). **“Mecánica de los fluidos e hidráulica”**. Mc Graw-Hill. Serie Schaum. 3ra Edición. ISBN: 8448118987

Ormsbee L. (2006). **“The history of water distribution network analysis: The computer age”**. Proceedings of the 8th Annual Water Distribution Systems Analysis Symposium, 1-6. Cincinnati. doi: 10.1061/40941(247)3

Rossman L. (1999), **“Computer Models/EPANET”**. Water Distribution Systems Handbook. Cap. 12, McGraw-Hill, Pp. 12.1-12.23.

Rossman L. (2000), **“EPANET 2 User’s Manual”**, Water Supply and Water Resources Division, National Risk Management Research Laboratory, Cincinnati, OH.

Saldarriaga J. (1998). **“Hidráulica de Tuberías”**, Mc. Graw Hill. ISBN: 958-600-831-2.

Savic D. (2007), **“Water distribution system analysis: Past, present, future”**, Approvvigionamento e Distribuzione Idrica: Esperienze, Ricerca ed Innovazione, Pp. 1-12, Morlacchi Editore.

Sotelo G. (1999). **“Hidráulica General”**. Limusa. Noriega Editores. Vigésimoprimer reimpresión. ISBN: 968-18-0503-8.

Swamee, P.K. and Jain, A.K. (1976), **“Explicit equation for pipe flow problems.”**, J. Hydr. Div., ASCE, 102(5), 657-664, 1976

**** Capítulo VI.- Análisis y Modelado. ****

Alperovitz E., Shamir U. (1977). **“Design of Optimal Water Distribution Systems”**. Water Resources Research. Vol. 13, No. 6, Pp. 885-900. doi: 10.1029/WR013i006p00885

Clerc M. (2010). **“From Theory to Practice in Particle Swarm Optimization”**. Handbook of Swarm Intelligence. Adaptation, Learning, and Optimization Volume 8, 2010, pp 3-36. doi: 10.1007/978-3-642-17390-5_1

Krause P., Boyle D., Bäse F. (2005). **“Comparison of different efficiency criteria for hydrological model assessment”**. Advances in Geosciences Vol. 5. Pp. 89-97. doi: 10.5194/adgeo-5-89-2005

Moriasi D., Arnold J., Van Liew M., Bingner R., Harmel R., Veith T. (2007). **“Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations”**. Transactions of the ASABE, Vol. 50, No. 3. Pp. 885-900.

Yuan X., Wang L., Yuan Y. (2008). **“Application of enhanced PSO approach to optimal scheduling of hydro system”**. Energy Conversion and Management Vol. 49 No.11. Pp. 2966 – 2972. doi: 10.1016/j.enconman.2008.06.017

>>> ANEXOS

“Yo no creo que haya ninguna emoción que puede atravesar el corazón humano como el que siente el inventor cuando ve que alguna creación del cerebro se desarrolla hacia el éxito... tales emociones hacen que un hombre olvide alimentación, el sueño, los amigos, el amor, todo....”

-Nikola Tesla

Código en Matlab de la Rutina “Reinicialización y Mutación con división de trabajo” (RMDL).

El código mostrado a continuación es una copia fiel del usado en la última rutina desarrollada en esta investigación. A simple vista parecerá incomprensible, por lo que se recomienda copiar todo el texto y pegarlo en un archivo nuevo de Matlab o, en su defecto, pegarlo en un documento de Word con una hoja de 55 cm de ancho con orientación horizontal. Para evitar posibles errores en su funcionamiento, el autor decidió dejar el código exactamente como se programó en Matlab.

Se reitera que, si se desea el código de otras rutinas, se pueden solicitar de manera gratuita al correo del autor: iprg1981@hotmail.com

```

%Idem al archivo ID, en esta rutina se implementó
el método RMP SO-DL
%(Reinitialization and Mutation Operators in PSO
with Division of Labor).

%Datos generales Rutina
% 1.- Influencia social se da por la mejor
partícula del grupo encontrada
%      en toda la rutina
% 2.- Velocidad máxima = 4
% 3.- Factor de aptitud con penalización (por
gasto menor a la demanda y
%      por gasto mayor a la demanda más la fuga)
% 4.- Si se rebasan los límites de velocidad se
establecen velocidades en
%      los límites Vmax o Vmin
% 5.-  $\phi_{uno} = 1.494 \cdot \text{rand}$  ,  $\phi_{idos} = 1.494 \cdot \text{rand}$ 
% 6.- La inicialización de velocidades se da por
el método half-diff
%      RANDespbusq = Aleatorio en espacio de
búsqueda (entre Xmin y Xmax)
%      Vinicial = 0.50 [RANDespbusq -
Xinicial]
% 7.-Nuevos parámetros para implementar Clamping
PSO:
%      Si  $X(t+1) < X_{min}$ , entonces
 $X(t+1) = X_{min}$ ,  $V(t+1) = 0$ 
%      Si  $X(t+1) > X_{max}$ , entonces
 $X(t+1) = X_{max}$ ,  $V(t+1) = 0$ 
% 8.-Inicialización caótica de posiciones:
%      Para la primera dimensión de cada
partícula se escoge un valor
%      aleatorio entre 0 y 1; dicho valor NUNCA
puede ser 0.25, 0.50 o 0.75.
%      Para las demás iteraciones se tendrá
 $f(k+1) = L \cdot f(k) \cdot [1 - f(k)]$ ;  $L =$ 
%      [3.56 , 4.00]. Los valores "f" son los
factores del caos,
%      posteriormente se obtendrán los gastos
mediante:

```

```

%      X(i,k)= f(i,k)*qdif+QDEM(i,k)
% 9.- RMP SO-DL. Procedimiento:
%      a) División del enjambre en 5 partes S1,
S2, S3, S4, S5.
%      b) Velocidad cambia a: V= G1 + G2 + G3
%          G1 = W*V
%          G2= phiuno*(Pi -X)
%          G3= phidos*(Pg -X)
%      c) Velocidad diferente para cada enjambre:
%          S1: W=0.30
%          S2: G1=0
%          S3: G2=0
%          S4: W4=0.90
%          S5: W==0.5+random/2 (propuesto
por JP, no viene en artículo)
%      d) Establecer parámetros M, m, eps
%          d.1) M= Máximo de iteraciones sin
encontrar un mejor global (50)
%          d.2) m = Brecha de mutación. Se
buscará un mejor global con una
%          distribución gaussiana si se han
tenido "m" iteraciones
%          sin encontrar un mejor global
(10).
%          d.3) eps= Límite mínimo de
diversidad. Revisa si las partículas
%          han perdido diversidad (0.05)
%          d.4) Pm = Probabilidad de mutación
[0.1, 0.3]
%      e) Vuelo PSO:
%          e.1) Inicializar partículas y
velocidades, flag=0
%          e.2) Evaluar aptitud, actualizar Pg y
Pi
%          e.3) Si se actualiza Pg entonces
flag=0, de otra forma flag=flag+1
%          e.4) Actualizar velocidades de
acuerdo al sub-enjambre
%          e.5) Actualizar posiciones
X(t+1)=X(t) + V(t+1)
%          e.6) Si flag = M entonces revisar la
diversidad "D". Si D < eps,

```

```

%           entonces reinicilizar las
posiciones y velocidades de todas las
%           partículas de todos los
enjambres.
%            $D = (1/\text{Particulas}) * \text{SUMATORIA1} [$ 
SUMATORIA2( $X_{ij} - \text{Promedio } X_j$ ) $^2 ]^{0.50}$ 
%           La SUMATORIA1 es de 1 hasta el
número de partículas, la
%           SUMATORIA 2 es de 1 hasta todas
las dimensiones.
%           e.7) Si flag=m y si random<Pm,
calcular Pg' con :
%            $P'g = Pg * [1 + \text{RandGauss}(\text{PHI})]$ ;
PHI=0.10*(Xmax-Xmin)+Xmin
%           RandGauss hace referencia a
obtener un número aleatorio
%           generado con distribución
gaussiana con base en PHI
%           Si P'g es mejor que Pg,
establecer P'g = Pg

%% Requisitos para que la rutina se efectúe
apropiadamente

% 1.- La rutina debe tener la siguiente
localización: "C:\Matlab\PSO_JPRG\"
% 2.- El archivo epanet2d.exe deberá estar en el
mismo directorio que la
%       rutina de Matlab.
% 3.- En el directorio donde se encuentra la
rutina, se deberán tener los
%       archivos de texto siguiente:

%       a) reservoir.txt (información de la red a
analizar respecto a los
%           reservorios)
%       b) junctions.txt (información de la red a
analizar respecto a los
%           nodos)

```



```

%      c) pipes.txt (información de la red a
analizar respecto a las tuberías)
%      d) coordinates.txt (información de la red a
analizar respecto a las
%      coordenadas de los nodos)
%      e) PRESIONES.txt (presiones que deberá
alcanzar el programa para
%      detenerse). Presiones obtenidas en campo
o presiones obtenidas con
%      EPANET cuando se incluyeron las fugas.
%      Estos archivos de texto no deberán tener
encabezados y deberán estar
%      acomodados como si se fueran a importar a
EPANET. O como si se
%      exportara una red con formato INP desde
EPANET.
%      Como ejemplo: Genere la red en EPANET,
expórtela con formato
%      INP, abra el archivo en Excel y guárdelo
bajo el formato "texto deli-
%      mitado por tabulaciones". Para hacer el
archivo "junctions.txt" borre
%      toda la información que no sea de este
apartado. Borre el encabezado.
%      Guárdelo bajo el nombre "junctions.txt".
Haga lo mismo para los demás
%      archivos que la rutina solicita.
% 4.- El folder activo deberá ser en el que se
encuentra la rutina pues ahí
%      se generarán unos reportes de salida.
% 5.- Para abrir el archivo
"PSO_Información_Adicional" (para analizar la
%      rutina al fina)se recomienda abrirlo en
excel, Origen del archivo:
%      Windows(ANSI), datos delimitados,
separadores tabulación y espacio,
%      posteriormente reemplazar los guiones bajos
por espacios (ctrl+l),
%      dar formato a los números "general".

%% Limpiando datos,variables

```

```

clc
clear all
close

% Se crea el archivo PSO_Datos_Análisis para
almacenar todos los datos de
% la rutina para su posterior analisis. La 'wt'
indica que, si ya se tenía
% un archivo similar, este se sobrescriba y que
se respeten los saltos de
% línea.

%% Presentación de la rutina

fprintf('*** Rutina PSO para detección de fugas
en una red de Agua Potable *** \n Autor: Juan
Pablo Ramírez Gutiérrez. E-
mail:jprg1981@gmail.com \n');
fprintf('\n')

%% Definición de las variables de control

% Establece las partículas en la PSO (P.314 Swarm
Intelligence)
particulas= input('Número de partículas de la PSO
(10 a 50): ');

% Corroborar si el número de partículas no es
menor a cero ni fraccionario y,
% de ser así, no se deja avanzar al usuario
while particulas < 5 || fix(particulas) ~=
particulas
    fprintf('\n');
    fprintf('** El número de partículas en esta
rutina debe ser un ENTERO NO MENOR A CINCO **
\n');
    particulas= input('Número de partículas de la
PSO (10 a 50): ');
end

```

```

% Definiendo cuántas partículas tiene cada grupo.
Son 5 grupos de enjambres
% en esta rutina.
partporgrupo=fix(particulas/5);

%% Determinando el volumen de fuga

% Se cargan los gastos de los nudos (desde un
archivo) y se le pide al usuario
% el gasto de entrada a la red. Con dichos datos,
podemos determinar
% las pérdidas a buscar. Este rango de pérdidas
nos permite saber el límite
% de búsqueda de las partículas.

% Carga el archivo que contiene 3 columnas de
información, 1.-Número de
% nudo, 2.-Elevación topográfica del nudo, 3.-
Gasto en el nudo
JUNCT = load('junctions.txt');

% Se almacenan los gastos de demanda (3era
columna de JUNCT) en la matriz
% QDEM
QDEM=JUNCT(:,3);

% Cierra el archivo junctions.txt para evitar
saturar la memoria
fclose('all');

% Se obtiene la suma de los elementos de la
matriz QDEM para saber cuánta
% agua se está demandando
qsum=sum(QDEM);

% Se le pide al usuario el gasto de entrada a la
red (Gastos suministrados),
% antes de eso se le dice el gasto demandado para
evitar que dé un gasto

```

```

% menor a éste.
fprintf('\n')
fprintf('Gasto demandado por la red: %0.2f lps
\n',qsum);
qentr=input( 'Gasto de entrada a la red (lps)?:
');

% Corroborar si el gasto de entrada no es menor al
de demanda y, de ser así,
% no se deja avanzar al usuario
while qentr < qsum
    fprintf('\n');
    fprintf('** El gasto de entrada NO PUEDE SER
MENOR al gasto de demanda ** \n');
    qentr=input( 'Gasto de entrada a la red
(lps)?: ');
end

% Corroborar si el gasto de entrada es igual al de
demanda, de ser así,
% termina la rutina.
if qentr==qsum
    fprintf('\n');
    fprintf('\n');
    fprintf('No se tienen fugas, no se necesita
analizar nada \n');
    fprintf('Se concluye la rutina de la PSO\n');
    fprintf('\n');

    % Termina la rutina
    return
end

% Volumen de pérdidas (Agua suministrada - Agua
demandada)
qdif=qentr-qsum;
%% Preguntando por Vmax

vmaxuser=4;

```

```

%% Preguntando si se guardarán datos

guardardatos=input ('Desea guardar los datos de
la rutina en un archivo (S/N)?: ', 's');

% Si la opción dada por el usuario es la
esperada, se genera una variable
% que romperá el ciclo abajo descrito
opcion=0;
if guardardatos=='s' || guardardatos=='n' ||
guardardatos=='S' || guardardatos=='N'
    opcion=1;
end

% Verifica que la respuesta dada por el usuario
no sea incorrecta, de ser así,
% no se deja avanzar al usuario
while opcion==0

    fprintf('\n');
    guardardatos=input ('Desea guardar los datos
de la rutina en un archivo (S/N)?: ', 's');

    if guardardatos=='s' || guardardatos=='n'
|| guardardatos=='S' || guardardatos=='N'
        opcion=1;
    end

end

%% Preguntando si se graficarán los datos

graficardatos=input ('Desea graficar la
exploración de las partículas? GRAN CONSUMO DE
MEMORIA ! (S/N)?: ', 's');

% Si la opción dada por el usuario es la
esperada, se genera una variable
% que romperá el ciclo abajo descrito

```

```

opcion=0;
if graficardatos=='s' || graficardatos=='n' ||
graficardatos=='S' || graficardatos=='N'
    opcion=1;
end

% Verifica que la respuesta dada por el usuario
no sea incorrecta, de ser así,
% no se deja avanzar al usuario
while opcion==0

    fprintf('\n');
    graficardatos=input ('Desea graficar la
exploración de las partículas? GRAN CONSUMO DE
MEMORIA ! (S/N)? : ', 's');

    if graficardatos=='s' || graficardatos=='n'
|| graficardatos=='S' || graficardatos=='N'
        opcion=1;
    end

end

%% Preguntando cuántas veces se simulará esta
rutina
archivos=input ('Cuántas veces desea simular esta
rutina?: ');

sufijo42= clock();
nomarchivo = ['PSO_Resumen_de_todas_rutinas'
num2str(sufijo42) '.txt'];
file = fopen( nomarchivo, 'wt');
fprintf(file, 'Fecha Hora Partículas Tiempo FA
Iteraciones Gastos\n');

for arccount=1 : 1 : archivos
sufijo= clock();
% Se crea el archivo PSO_Datos_Análisis para
almacenar todos los datos de
% la rutina para su posterior analisis. La 'wt'
indica que, si ya se tenía

```

```

% un archivo similar, este se sobrescriba y que
se respeten los saltos de
% línea.
nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo, 'wt');
fprintf(file, 'Análisis_de_datos_PSO\n');
fclose ('all');

%% Registrando y almacenando el tiempo de
ejecución

% Almacenando el tiempo de arranque en la
variable tiempo
tiempo=clock();

% Grabando el tiempo de inicio de la rutina como
nombre de ejercicio
nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo, 'a');

fprintf(file, 'Ejercicio %s\n', datestr(tiempo));
fprintf(file, '\n');
fclose ('all');

%% Establecimiento de límites de velocidad

% Según Swarm Intelligence (P.296), la velocidad
se recomienda entre el
% rango [Vmín=-4, Vmax=4]. Para este ejercicio,
se optó por considerar los
% límites de velocidad como un porcentaje del
rango de búsqueda.

% Por lo que, la velocidad máxima será:
velmax= vmaxuser;

```

```

% La velocidad mínima será.
velmin= -velmax;

%% Posición inicial del enjambre

% La configuración de una partícula será:
%
%      Partícula = [GastoNodo1, GastoNodo2,
GastoNodo3 ..... GastoNodo"n"]
%
% Siendo "n" el número de nodos que contiene la
red.

% Se genera un ciclo para cada partícula que
asigna un valor aleatorio con
% una distribución normal para cada elemento de
la partícula. Dicho número
% aleatorio no será mayor a qdif, el cual es el
rango de búsqueda.

% Se obtiene el número de nodos de la red (que
será el número de elementos
% que tendrán todas las partículas.
[m,~] = size(QDEM);
numeronodos = m;

% Se definen las dimensiones de la matriz "X" y
"OWNBEST" para
% evitar consumo de memoria. La matriz "X"
almacena los gastos para cada
% elemento de las partículas. Su configuración es
filas = Partículas y
% columnas = gastos en los nodos de la red. La
matriz "OWNBEST"
% almacena las mejores posiciones alcanzadas
(gastos) para cada partícula.
% Su configuración es filas = partículas,
columnas = gastos en nodos de la
% red.
X=zeros(particulas,numeronodos);
OWNBEST=zeros(particulas,numeronodos);

```



```

PENVOL=1000*ones(particulas,numeronodos);
PENDEM=1000*ones(particulas,numeronodos);
L= zeros(particulas,numeronodos);
F = zeros(particulas,numeronodos);
W = zeros(particulas,numeronodos);
% Asignando los valores aleatorios (gastos) a los
elementos de las partículas.
for i=1:1:particulas
    for elem=1:1:numeronodos

        if elem==1
            % Para la primera dimensión, el número
            aleatorio no puede ser
            % 0.25, 0.50 ó 0.75.
            F(i,elem)= 0.25;
            while F(i,elem)==0.25 ||
F(i,elem)==0.50 || F(i,elem)==0.75
                F(i,elem) = rand;
            end
            % Para la primera iteración no hay
            valor de L
            L(i,elem) =0;
        else
            % Se escoje un valor aleatorio
            entre 3.56 y 4
            L(i,elem)=rand*(4-3.56)+3.56;
            % Se obtiene el factor del caos
            para la dimensión de la
            % partícula en cuestión.
            F(i,elem) = L(i,elem)*F(i,elem-
1)*(1-F(i,elem-1));
        end

        %Se obtiene el valor del gasto.
        X(i,elem)= F(i,elem)*qdif +
QDEM(elem,1);

        % Se revisa si el gasto calculado es
        mayor a la suma de la
        % demanda en el nodo mas el volumen
        total de fugas.
    end
end

```

```

        if X(i,elem)> (QDEM(elem,1)+ qdif)
            PENVOL(i,elem)= 900000*(X(i,elem)-
QDEM(elem,1)- qdif);
        else
            PENVOL(i,elem)=0;
        end

        % Se revisa si el gasto calculado es
menor a la demanda
        % en el nodo.
        if X(i,elem)< QDEM(elem,1)
            PENDEM(i,elem)=
900000*(QDEM(elem,1)-X(i,elem));
        else
            PENDEM(i,elem)=0;
        end

        % OWNBEST almacenará la mejor posición
alcanzada por cada una de las
        % partículas
        OWNBEST(i,elem)= X(i,elem);
    end
end

%% Velocidad inicial del enjambre

% Ciclo que asigna velocidades iniciales
aleatorias a todos los elementos
% de las partículas del enjambre.

% Se define el tamaño de la matriz "VEL" para
evitar consumo de memoria.
% "VEL" almacenará las velocidades de los
elementos de las partículas
% de la iteración en curso. Su configuración es
filas = partículas,
% columnas = velocidades de los elemento sde la
partícula.
VEL=zeros(particulas,numeronodos);
RANDESPBUSQ=zeros(particulas,numeronodos);

```

```

% Asignando los valores aleatorios (velocidades)
a los elementos de las partículas.
% Método Half-Diff
for i=1:1:particulas
    for elem=1:1:numeronodos

        Xmin=QDEM(elem,1);
        RANDESPBUSQ(i,elem)=rand*qdif+Xmin;
        valorvel= 0.50*(RANDESPBUSQ(i,elem)-
X(i,elem));

        VEL(i,elem)= valorvel;

    end
end

%% Configuración de matrices para función de
aptitud y PSO

% Este ciclo les actualiza a todas las partículas
un factor de aptitud muy,
% muy bajo (la máxima aptitud es igual a cero, a
FABEST Y FA se les asigna
% un valor de 40000). Éste se usará para que, más
adelante, se pueda actua-
% lizar la mejor posición en una partícula.

% Se define la dimensión de las siguientes
matrices para evitar consumo de
% memoria. "FAOWNBEST" contiene los mejores
valores de aptitud alcanzados
% hasta ese momento por la partícula en cuestión.
% "GLOBALBEST" incluye la configuración de gastos
alcanzada por cualquier
% partícula que, hasta ese momento, sea la de
mayor factor de aptitud de
% entre todas las partículas. Su configuración es
fila = partícula (solamente
% 1 fila), columna = gastos en los nodos.

```

```

FAOWNBEST=ones(particulas,1)*99e+99;
GLOBALBEST=zeros(1,numeronodos);
GROUPBEST=ones(1,numeronodos)*99e+99;

% Para el análisis detallado de la PSO se
almacenarán los valores de
% phiuno, phidos, componente propio, componente
social, factores de aptitud
% y presiones de las configuraciones de gastos.
Ver sección
% "Posiciones y velocidades para el nuevo ciclo
PSO".
PHIUNO=zeros(particulas, numeronodos);
PHIDOS=zeros(particulas, numeronodos);
COMPONENTEPROPIO =zeros(particulas, numeronodos);
COMPONENTESOCIAL= zeros(particulas, numeronodos);
FA=zeros(particulas,1);
PRESPSOACUM=zeros(particulas,numeronodos);

% Límites RMPSO-DL
pm=0.30;
EME = 50;
eme = 10;
eps = 0.05;

% Matrices RMPSO-DL
PROMEDIOXJ=zeros(1,numeronodos);
SUMATORIADOS=zeros(particulas,1);
DIFCUAD=zeros(particulas,numeronodos);
BESTGAUSS=zeros(1,numeronodos);
PRESPSOACUMGAUSS=zeros(1,numeronodos);

% Se voltea la matriz "X" que contiene los gastos
finales (configuración)de
% la red que cada "pájaro" ve. Ésto se hace para
hacer el archivo txt para
% evaluarlo con EPANET. "X" ahora tiene en sus
filas los nodos de la red,
% en sus columnas la configuración de la red
seleccionada por cada
% partícula.

```

```

X=X';

% Se define el mayor gasto que se pudiera tener
(Mayor demanda + fugas)
limgastos = max(QDEM)+qdif;

%% Imprimiendo datos inicialización en TXT.

if guardardatos=='s' || guardardatos=='S'

% En todos los siguientes casos: Se abre el
archivo de texto con la opción
% 'a' para agregar y no sobrescribir. Después se
escribe el encabezado.
% Con el comando save se almacenan las matrices
de datos, se cierra el
% archivo txt. Ésto se repite para cada variable.
Se hizo así porque con
% fprintf no se almacena ordenada la matriz y, si
el archivo no se cierra,
% fprintf escribe erróneamente la información en
el archivo.

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo, 'a');

fprintf(file,
'*****_Información_sobre_las_matrices_*****\n');
fprintf(file, '
X=_Contiene_los_gastos_en_la_red_(filas=partícula
s,_columnas=nodos)\n');
fprintf(file, '
VEL=_Contiene_las_velocidades_para_cada_partícula
(filas=partículas,_columnas=nodos)\n');
fprintf(file, '
W=_Inercia(fila=partículas,_columnas=nodos)\n');
fprintf(file, '
PENVOL=_Penalizaciones_al_factor_apetitud_si_el_ga
sto_propuesto_es_mayor_a_la_demanda_con_el_volume

```

```

n_total_de_fugas_(filas=partículas,_columnas=nodos)\n');
fprintf(file, '
PENDEM=_Penalizaciones_al_factor_aptitud_si_el_ga
sto_propuesto_es_menor_a_la_demanda_(filas=partíc
ulas,_columnas=nodos)\n');
fprintf(file, '
PRESPSO=_Contiene_las_presiones_obtenidas_para_lo
s_gastos_de_la_red_que_obtuvo_la_PSO\n');
fprintf(file, '
PRESTEO=_Contiene_las_presiones_medidas_en_campo,
_las_presiones_a_alcanzar_por_la_PSO\n');
fprintf(file, '
FA=_Contiene_los_factores_de_aptitud_para_cada_pa
rtícula\n');
fprintf(file, '
FAOWNBEST=_Contiene_los_mejores_factores_de_aptit
ud_alcanzados_por_esa_partícula\n');
fprintf(file, '
OWNBEST=_Contiene_las_mejores_configuraciones_de_
gastos_alcanzadas_por_esa_partícula\n');
fprintf(file, '
GLOBALBEST=_Contiene_la_mejor_configuración_de_ga
stos_alcanzada_hasta_el_momento_por_cualquier_par
tícula\n');
fprintf(file, '
GROUPBEST=_Contiene_la_mejor_configuración_de_gas
tos_alcanzada_en_esta_iteración_por_cualquier_par
tícula\n');
fprintf(file, '
PHIUNO=_Contiene_números_aleatorios_que_varían_en
tre_cero_y_1.494_que_afectan_a_la_componente_pers
onal\n');
fprintf(file, '
PHIDOS=_Contiene_números_aleatorios_que_varían_en
tre_cero_y_1.494_que_afectan_a_la_componente_soci
al\n');
fprintf(file, '\n');
fprintf(file, '*****_Fórmulas_PSO_*****\n');
fprintf(file, '
Inicialización_de_"posiciones"<gastos>_(Método_d
el_Caos):\n');

```

```

fprintf(file, '
_Para_la_primera_dimensión_"k"_de_cada_partícula_
_se_escoge_un_valor_aleatorio_"F"_entre_0_y_1;\n')
;
fprintf(file, '
_dicho_valor_NUNCA_puede_ser_0.25,_0.50_ó_0.75\n'
);
fprintf(file, '
_Para_las_demás_dimensiones_se_tendrá_F(k+1)_=_L*
_F(k)*[1-F(k)];_L_[3.56,4.00]\n');
fprintf(file, '
_Las_posiciones(gastos)_iniciales_serán:_X(i,k)=_
_F(i,k)*Volumen_total_de_fugas+Demanda_en_el_nodo\
n');
fprintf(file, '
Clamping_PSO:_Si_X(t+1)<Demanda_en_nodo,_entonces
_X(t+1)=Demanda_en_nodo;V(t+1)=0\n');
fprintf(file, '
Clamping_PSO:_Si_X(t+1)>Demanda_en_nodo+volumen_f
ugas,_entonces_X(t+1)=Demanda_en_nodo+volumen_fug
as;V(t+1)=0\n');
fprintf(file, ' FA= 1/n *_SUMATORIA[(Preal-
Ppso)^2+PENDEM+PENVOL]\n');
fprintf(file, '
Si_Xicalc<Demanda_en_nodo,_entonces:PENDEM=9x10^5
*{Demanda_en_nodo-Xicalc}\n');
fprintf(file, '
Si_Xicalc>(Demanda_en_nodo+Vol.Fugas),_entonces:P
ENVOL=9x10^5{Xicalc-Demanda_en_nodo-
Vol.Fugas}\n');
fprintf(file, '
Inicialización_de_velocidades_(Método_Half_diff):
\n');
fprintf(file, ' Vini=0.50*(randespbusq-Xini)\n');
fprintf(file, ' randespbusq=random*qdif+Xmin\n');
fprintf(file, '
Se_realiza_para_cada_dimensión_de_cada_partícula,
_qdif_es_el_volumen_total_de_fuga,_Xmin_es_la_dem
anda_en_es_e_nodo.\n');
fprintf(file, '
Xini_es_el_valor_de_gasto_aleatorio_asignado_a_es
a_misma_partícula_en_esa_misma_dimensión\n');

```

```

fprintf(file, '\n');
fprintf(file, ' *_Procedimiento_RMPSO-
DL_(Reinitialization_and_Mutation_Operators_in_PS
O_with_Division_of_Labor)\n');
fprintf(file, '
____a)_División_del_enjambre_en_5_partes_S1,_S2,_,
S3,_S4,_S5.\n');
fprintf(file, '
____b)_Velocidad_cambia_a: V= G1 + G2 + G3\n');
fprintf(file, ' _____G1 = W*V\n');
fprintf(file, ' _____G2= phiuno*(Pi_-
X) __;Componente_Propio\n');
fprintf(file, ' _____G3= phidos*(Pg_-
X) __;Componente_Social\n');
fprintf(file, '
____si_velocidad<velmin,_entonces_velocida
d=velmin\n');
fprintf(file, '
____si_velocidad>velmax,_entonces_velocida
d=velmax\n');
fprintf(file, '
____X= Xanterior+velocidad\n');
fprintf(file, '
____c)_Velocidad_diferente_para_cada_enjambre:\n'
);
fprintf(file, ' _____S1: W=0.30\n');
fprintf(file, ' _____S2: G1=0\n');
fprintf(file, ' _____S3: G2=0\n');
fprintf(file, ' _____S4: W4=0.90\n');
fprintf(file, ' _____S5: W==0.5+random/2\n');
fprintf(file, '
____d)_Establecer_parámetros_M,_m,_eps\n');
fprintf(file, '
____d.1)_M= Máximo_de_iteraciones_sin_encont
rar_un_mejor_global\n');
fprintf(file, '
____d.2)_m = Brecha_de_mutación._Se_buscará_
un_mejor_global_con_una\n');
fprintf(file, '
____distribución_gaussiana_si_se_han_te
nido_"m"_iteraciones\n');

```



```

fprintf(file, '
_____sin_encontrar_un_mejor_global.\n');
fprintf(file, '
_____d.3)_eps=_Límite_mínimo_de_diversidad._R
evisa_si_las_partículas\n');
fprintf(file, '
_____han_perdido_diversidad_\n');
fprintf(file, '
_____d.4)_Pm=_Probabilidad_de_mutación_[0.1,
_0.3]\n');
fprintf(file, ' _____e)_Vuelo_PSO:\n');
fprintf(file, ' _____e.1)_Inicializar_partículas_y_velocidade
s,_flag=0_\n');
fprintf(file, ' _____e.2)_Evaluar_aptitud,_actualizar_Pg_y_Pi
\n');
fprintf(file, '
_____e.3)_Si_se_actualiza_Pg_entonces_flag=0,
_de_otra_forma_flag=flag+1\n');
fprintf(file, ' _____e.4)_Actualizar_velocidades_de_acuerdo_a
l_sub-enjambre_\n');
fprintf(file, ' _____e.5)_Actualizar_posiciones_X(t+1)=X(t)+
_V(t+1)\n');
fprintf(file, ' _____e.6)_Si_flag=_M_entonces_revisar_la_div
ersidad_"D"._Si_D<_eps,\n');
fprintf(file, ' _____entonces_reinicializar_las_posicione
s_y_velocidades_de_todas_las\n');
fprintf(file, ' _____partículas_de_todos_los_enjambres.\n
n');
fprintf(file, '
_____D=(1/Particulas)*SUMATORIA1_[_SUMAT
ORIA2(Xij_-_Promedio_Xj)^2_] ^0.50\n');
fprintf(file, ' _____La_SUMATORIA1_es_de_1_hasta_el_núme
ro_de_partículas,_la\n');

```

```

fprintf(file, '
SUMATORIA_2_es_de_1_hasta_todas_las
dimensiones.\n');
fprintf(file, '
e.7)_Si_flag=m_y_si_random<Pm,_calcular_
MPg_con_:\n');
fprintf(file, '
MPg_=Pg*[1+RandGauss(PHI)];_PHI=0.1
0*(Xmax-Xmin)+Xmin\n');
fprintf(file, '
RandGauss_hace_referencia_a_obtener
un_número_aleatorio\n');
fprintf(file, '
generado_con_distribución_gaussiana
_con_base_en_PHI\n');
fprintf(file, '
Si_MPg_es_mejor_que_Pg,_establecer_
Pg=_MPg\n');
fprintf(file, '*****_Datos_red_*****\n');
fprintf(file, ' 1.-Demanda_de_la_red:\n');
QDEMinv=QDEM';

save (nomarchivo,'QDEMinv', '-ascii', '-append',
'-v6');
fprintf(file, ' 2.-Presiones_a_alcanzar:\n');
% Cargamos el archivo con las presiones que nos
deberían dar si no
% tuvieramos fugas.
PRESTEO=load('PRESIONES.TXT');
PRESTEOinv=PRESTEO';
save (nomarchivo,'PRESTEOinv', '-ascii', '-
append', '-v6');
fprintf(file, ' 3.-
Volumen_de_fugas:_%0.4f_\n',qdif);
fprintf(file, ' 4.-
Velocidad_máxima:_%0.4f_\n',velmax);
fprintf(file, ' 5.-
Velocidad_mínima:_%0.4f_\n',velmin);
fprintf(file, ' **_Datos_RMPSO-DL\n');

```

```

fprintf(file, '
__Valor_de_M=%0.2f,_Valor_de_m=%0.2f,_Valor_de_e
ps=%0.2f,_Valor_de_Pm=%0.2f\n', EME, eme, eps, pm);
fprintf(file, ' __División_en_sub-
enjambres:\n');
fprintf(file, ' __Primer_sub-
enjambre:_De_partícula_1_a_%d\n', partporgrupo);
fprintf(file, ' __Segundo_sub-
enjambre:_De_partícula_%d_a_%d\n',
partporgrupo+1,2*partporgrupo);
fprintf(file, ' __Tercer_sub-
enjambre:_De_partícula_%d_a_%d\n',
2*partporgrupo+1,3*partporgrupo);
fprintf(file, ' __Cuarto_sub-
enjambre:_De_partícula_%d_a_%d\n',
3*partporgrupo+1,4*partporgrupo);
fprintf(file, ' __Quinto_sub-
enjambre:_De_partícula_%d_a_%d\n',
4*partporgrupo+1,particulas);
fprintf(file, '\n');
close 'all'

```

```

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo, 'a');
fprintf(file,
'*****_Iteración_número_1_*****\n');
fprintf(file, '
L_(Factor_de_inicialización_para_posiciones._Méto
do_del_caos)\n');
save (nomarchivo, 'L', '-ascii', '-append', '-
v6');
fprintf(file, '\n');
close 'all'

```

```

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo, 'a');
fprintf(file, ' F_(Factor_del_caos)\n');
save (nomarchivo, 'F', '-ascii', '-append', '-
v6');

```

```

fprintf(file, '\n');
close 'all'

Xinversa=X';
nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo,'a');
fprintf(file, ' X_(Valores_de_gasto)\n');
save (nomarchivo,'Xinversa', '-ascii', '-append',
'-v6');
close 'all'

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo,'a');
fprintf(file, '\n');
fprintf(file, '
RANDESPBUSQ_(Valores_de_gasto_aleatorios_para_"Ha
lf_diff")\n');
save (nomarchivo,'RANDESPBUSQ', '-ascii', '-
append', '-v6');
close 'all'

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo,'a');
fprintf(file, '\n');
fprintf(file, '
VEL_(Velocidades_iniciales<Half_diff>) \n');
save (nomarchivo,'VEL', '-ascii', '-append', '-
v6');
close 'all'

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo,'a');
fprintf(file, '\n');
fprintf(file, '
PENVOL_(Penalización_por_volumen)\n');
save (nomarchivo,'PENVOL', '-ascii', '-append',
'-v6');

```

```

close 'all'

nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
file = fopen( nomarchivo,'a');
fprintf(file, '\n');
fprintf(file, '
PENDEM_ (Penalización_por_demanda)\n');
save (nomarchivo,'PENDEM', '-ascii', '-append',
'-v6');
close 'all'

end

%% Inicializando las iteraciones.

% Esta instrucción hace que, todo lo que se
grafique, se almacenará en la
% variable "fvuelo", se hace para almacenar el
vuelo de las partículas por
% el espacio de búsqueda y poder almacenarlo en
un archivo. Ver celda
% Graficando Movimientos PSO. Se establecen las
dimensiones de la ventana
% que mostrará los gráficos [Coord inicial X,
Coord final Y, Largo, Alto]
fvuelo=figure;
set(fvuelo, 'Position', [315 150 760 500])

% CICLO MAYOR: Este ciclo se hará una y otra vez
hasta que se tengan "itersinmejora"
% iteraciones sin mejora alguna. Si se encuentra
un mejor global, el contador
% se resetea a uno. El ciclo también terminasi se
alcanza elvalor esperado
% antes (factor de aptitud FA=0).

%Contador que nos permite saber exactamente
cuántas iteraciones se
%realizaron

```

```

contiter=0;
itersinmejora=801;
countflag=0;
countreini=0;
countpgprima=0;
flag=1;
iter=0;
while iter<itersinmejora
    % Aumenta el valor de iter
    iter=iter+1;

    %Indica si se ha encontrado un mejor global
por Gauss
    countpgprima=0;

    %Indica si se ha hecho una modificación por
Gauss
    modbygauss=0;

    %Indica si se han reinicializado las
posiciones y velocidades del
    %enjambre
    reinicial=0;

    % Aumenta el valor de flag RMPSO-DL
    flag=flag+1;

    % Contabiliza esta iteración en el contador
    contiter=contiter+1;

    % Cargando los archivos de origen para
evaluar las posiciones
    % iniciales del enjambre.
    RESERV=load('reservoir.txt');
    PIPE = load ('pipes.txt');
    COORD=load('coordinates.txt');

    % Cerrando los archivos txt para reducir el
consumo de memoria.
    fclose('all');

```

```

        % CICLO MENOR: Se evalúa la configuración
seleccionada por cada partícula
        % y, posteriormente, se selecciona una nueva
configuración.
        for i=1:1:particulas
            %% Armando archivo TXT para evaluarlo en
EPANET

            % La matriz JUNCT se actualiza con la
configuración de gastos que
            % ve el pájaro "i".
            JUNCT(:,3)=X(:,i);

            % Se crea el archivo llamado
"PSO_Datos_Entrada.txt", la 'w'
            % indica que, si ya existe un archivo
llamado así, se sobrescribirá.
            % Se hará referencia a este archivo
mediante la variable "file"

            file=fopen('PSO_Entrada.txt','w');

            % Se imprime en la primera línea
"[TITLE]". El parámetro "\n"
            % indica que ahí termina una línea y
genera una nueva para la
            % siguiente impresión
fprintf(file, '[TITLE]\n');
fprintf(file, 'PSO\n');

            % Genera una línea en blanco
fprintf(file, '\n');

            fprintf(file, '[RESERVOIR]\n');

            % Calcula el número de filas de la matriz
RESERV y lo almacena en
            % f, el número de columnas se desprecia
(~)

```

```

[f,~]= size(RESERV);

% Se realiza un ciclo "for" en el que se
obtendrán todos los datos
% de la matriz RESERV y se escriben en el
archivo txt. La variable
% "file" hace alusión al archivo que se
está creando. Se importarán
% 2 valores por fila "%d" y "%0.2f". El
primero (%d) hace alusión a
% un número entero, el segundo (%0.2f)
indica que es un número con
% dos decimales. El parámetro "\n" indica
que se debe concluir esa
% línea y empezar en una nueva.
    for fila=1:f
        fprintf(file, '%d %0.2f \n',
RESERV(fila,:));
    end

    fprintf(file, '\n');
    fprintf(file, '[JUNCTIONS]\n');

%Ciclo para escribir en el archivo TXT
los valores de la matriz JUNCT
[f,~]= size(JUNCT);
for fila=1:f
    fprintf(file, '%d %0.2f %0.2f \n',
JUNCT(fila,:));
end

    fprintf(file, '\n');
    fprintf(file, '[PIPES]\n');

%Ciclo para escribir en el archivo TXT
los valores de la matriz PIPE
[f,~]= size(PIPE);
for fila=1:f
    fprintf(file, '%d %d %d %0.2f %d %d
\n', PIPE(fila,:));
end

```



```

fprintf(file, '\n');
fprintf(file, '[COORDINATES]\n');

% Se obtienen el número de filas de la
matriz COORD, las columnas
% se desprecian (~)
[f,~]= size(COORD);

% Ciclo para escribir en el archivo TXT
los valores de la matriz
% COORD
for fila=1:f
    fprintf(file, '%d %0.2f %0.2f \n',
COORD(fila,:));
end

fprintf(file, '\n');
fprintf(file, '[OPTIONS]\n');
fprintf(file, 'UNITS LPS\n');
fprintf(file, 'HEADLOSS H-W\n');
fprintf(file, '\n');
fprintf(file, '[REPORT]\n');
fprintf(file, 'NODES All\n');
fprintf(file, 'LINKS All\n');
fprintf(file, '\n');
fprintf(file, '[END]\n');

% Cierra todos los archivos externos
cargados para evitar consumo
% de memoria.
fclose('all');

%% Evaluando el Archivo TXT en EPANET

% Evaluando el archivo creado en EPANET,
el nombre del archivo de
% entrada es PSO_Entrada.txt, los
resultados se exportarán en un

```

```

        % archivo TXT que se creará con el nombre
        PSO_Salida.txt
        dos('C:\MATLAB\PSO_JPRG\epanet2d.exe
        PSO_Entrada.txt PSO_Salida.txt');

        %% Obteniendo Presiones del archivo
        calculado por EPANET

        % Se obtiene el tamaño de la matriz JUNCT
        para saber cuántas filas
        % de nodos son:
        [m,~]=size(JUNCT);

        % "lineult" calcula hasta que fila del
        archivo de texto se extraerán
        % los datos de presiones. EPANET siempre
        imprime los datos de los
        % nodos (en el archivo de salida) desde
        la fila 44, aquí se suman
        % las filas de nodos con 43 para no
        duplicar la primera fila.
        lineult = 43 + m;

        % Contador que nos servirá para almacenar
        los datos en una matriz
        linea=0;

        % Se define el tamaño de la matriz TEMP
        para evitar consumo de
        % memoria
        TEMP=zeros(m,4);

        % Se realiza un ciclo para extraer los
        datos en el rango de filas
        % antes analizado.
        lineiniarranq=44;
        for lineini=lineiniarranq:1:lineult

            linea=linea+1;

```

```

        %Abre el archivo donde se
encuentran las presiones
        fid=fopen('PSO_Salida.txt');

        % Función para extraer texto del
archivo, el número "4"
        % hace alusión al número de datos
que se extraerán, la va-
        % riable "lineini-1" nos indica
la fila de donde se extrae-
        % rán los datos.
        r=textscan(fid, '%f', 4,
'delimiter',' ' , 'headerlines', lineini-1);

        % Con esta función convertimos
los valores extraídos a una
        % matriz y los almacenamos en "a"
        a=cell2mat(r);

        %en caso de tenerse presiones
negativas, EPANET emitirá una
        %advertencia en su reporte y, por
tal motivo, las presiones
        %estarán en otra línea. Primero
se revisa si la matriz está
        %vacía y, de ser así, se rompe
este ciclo y se ejecuta el siguiente.

        emptymatrix=isempty(a);
        if emptymatrix==1
            %Se rompe este ciclo for

            break;
        end

        % Almacenamos los datos en la
matriz TEMP, se iguala a una
        % matriz inversa para acomodar
los datos como los presenta

```

```

        % EPANET.
        TEMP(linea,:)=a';

        % Cierra todos los archivos
        externos cargados para evitar
        % que se sature la memoria
        fclose('all');
    end

    %Si se tuvieron presiones negativas
    en el ciclo anterior,
    %entonces se terminó ese ciclo y se
    cumplirá esta condición en
    %la que se tiene el formato de EPANET
    con advertencia de las
    %presiones negativas
    if isemptymatrix==1

        lineult = 45 + m;
        linea=0;
        TEMP=zeros(m,4);
        lineiniarranq=46;
        fid=fopen('PSO_Salida.txt');
        r=textscan(fid, '%f', 4,
        'delimiter','' , 'headerlines', lineini-1);
        a=cell2mat(r);

        for
        lineini=lineiniarranq:1:lineult
            linea=linea+1;

        fid=fopen('PSO_Salida.txt');
            r=textscan(fid, '%f', 4,
        'delimiter','' , 'headerlines', lineini-1);
            a=cell2mat(r);
            TEMP(linea,:)=a';
            fclose('all');
        end
    end
end

```

```

        % Extraemos sólo los valores de presión y
los almacenamos en la
        % matriz PRESPSO
PRESPSO(:,1)=TEMP(:,4);

        %Almacenando las presiones obtenidas
PRESPSOACUM(i,:)=PRESPSO(:,1);

        % Cargamos el archivo con las presiones
que nos deberían dar si no
        % tuvieramos fugas.
PRESTEO=load('PRESIONES.TXT');

        % Cierra todos los archivos externos
cargados para evitar que se
        % sature la memoria
fclose('all');

        % Las presiones teóricas son aquellas que
se obtienen físicamente
        % (por lo que contienen un gasto de
fuga), las presiones encontradas
        % por la PSO son aleatorias. Cuando ambas
presiones seann iguales,
        % habremos encontrado los gastos reales y
veremos donde está la fuga.
DIFPRES=PRESTEO-PRESPSO;

        %% Calculando el Factor de Aptitud

        % El factor de aptitud se calculará
mediante la fórmula :
        %      fa = SUMATORIA( Pmedidas -
Pcalculadas)^2/No Mediciones

        % Obteniendo las dimensiones de la matriz
para determinar el número
        % de mediciones
[m,~]=size(DIFPRES);

```

```

        % Elevando al cuadrado la diferencia de
presiones
        DIFPRES=DIFPRES.^2;

        % Obteniendo el factor de aptitud
        aptitudparticula =
        (sum(DIFPRES)+sum(PENVOL(i,:))+sum(PENDEM(i,:)))/
m;

        %Almacenando Factor de aptitud
        FA(i,1)=aptitudparticula;

        % Permite saber cuándo se ha alcanzado un
mejor global, si cambio
        % es cero no se ha obtenido un nuevo
mejor global, si cambio es
        % uno, se ha obtenido un nuevo mejor
global.
        cambio=0;

        % Si la aptitud de esta partícula es
mejor (menor) que la mejor
        % aptitud alcanzada hasta el momento (por
esa misma partícula)
        if aptitudparticula<FAOWNBEST(i,1)

                % Se revisa si se ha alcanzado la
mejor aptitud de toda la
                % rutina
                if aptitudparticula <
min(min(FAOWNBEST))
                        % Se almacena la
configuración de gastos como la mejor
                        % que se ha obtenido en la
rutina.

                                GLOBALBEST(1,:)=X(:,i);

```

```

                                % Se resetea el contador a
uno, se tendrán que realizar
                                % otras "itersinmejora"
iteraciones para que se acabe la rutina.
                                iter=1;

                                % Como se obtuvo un mejor
global, se resetea el valor
                                % de flag a uno RMPSO-DL
                                flag=1;

                                % Indica que se ha obtenido
un nuevo mejor global.
                                cambio=1;

                                % Se almacena en qué
iteración se obtuvo el mejor
                                % resultado
                                bestiter= contiter;
                                end

                                % Almacena el valor de aptitud como
el mejor valor alcanzado al
                                % momento por esa partícula
                                FAOWNBEST(i,1) = aptitudparticula;

                                % Almacena esa configuración de
gastos como la mejor
                                % configuración alcanzada por esa
partícula al momento
                                OWNBEST (i,:) = X(:,i);
                                end

                                end % Aquí termina el ciclo de las
partículas.

                                %Almacenando la mejor configuración de la
iteración

```

```

for i=1:1:particulas
    if FA(i,1)==min(min(FA))
        GROUPEBEST(1,:)= X(:,i);
    end
end

%% Graficando Movimientos PSO

    if graficardatos=='S' || graficardatos=='s'

        %Se definen los ejes del gráfico ([ -X X
-Y Y])
        axis([0 (contiter*1.05) 0
limgastos*1.05])

        % Igualamos la gráfica a una variable
para que, posteriormente
        % podamos aumentar el tamaño del
marcador. '.b' quiere decir que
        % grafique puntos color azul. Se
grafican todas las filas
        % (gastos) de la columna (partícula)
"contpartic"
        for contpartic=1:1:particulas
            h= plot(contiter,
X(:,contpartic), '.b');
            set(h, 'Markersize', 10)
        end

        % Evita que se borren los puntos antes
graficados al graficar
        % un nuevo punto.
hold on

        % Imprime el mejor punto hasta el
momento. Con un punto rojo de
        % mayor tamaño.
y= plot(contiter,GLOBALBEST(1,:), '.r');
set(y, 'Markersize', 10)

```



```

        % Se imprimen 4 líneas en el título: la
iteración, la mejor
        % posición, el mejor factor de aptitud,
el número de partículas.
        % Se rotulan los ejes X y Y
        title({[];['PSO Iteraciones: ',
num2str(contiter)]; ['Mejor Factor de Aptitud :'
,num2str(min(min(FAOWNBEST)))]}; ['Iteración con
el mejor resultado:',
num2str(bestiter)]; ['Partículas: ',
num2str(particulas)]; ['Flag: ',
num2str(flag)]; ['Ciclos Flag : ',
num2str(countflag)]; ['Reinicializaciones: ',
num2str(countreini)]; ['Mejor Global por Gauss: ',
num2str(countpgprima)]; []))
        xlabel('Iteración')
        ylabel('Gasto (lps)')
        legend_handle=legend('Puntos rojos =
Mejor hasta ahora');

set(legend_handle, 'location', 'SouthOutside')
set(legend_handle, 'Box', 'off')

        %Se hace una breve pausa para apreciar
los puntos.
        pause(.000001)

        % Si se alcanza el número máximo de
iteraciones, se hace una pausa
        % hasta que el usuario de ENTER y se
cierra la figura.
        if iter==itersinmejora
            clc
            % Se obtiene la diferencia entre el
tiempo en el que inició la
            % rutina y el tiempo en que terminó
la rutina y se almacena en
            % la variable "T"
            ahora=clock();
            T=etime(ahora(),tiempo);

```



```

% termina el ciclo.
if min(min(FAOWNBEST))==0
    % Borrar pantalla
    clc
    % Se obtiene la diferencia entre el
    tiempo en el que inició la
    % rutina y el tiempo en que terminó
    la rutina y se almacena en
    % la variable "T"
    ahora=clock();
    T=etime(ahora(),tiempo);

    % Nota de aviso que se concluyó antes
    la rutina.
    fprintf(' \n')
    fprintf('**** NOTA : ****\n')
    fprintf('Se alcanzó el óptimo antes
de cumplirse el número programado de
iteraciones\n')
    % Termina la rutina
    break
end

%% Posiciones y velocidades para el nuevo ciclo
PSO

% Revisando si se ha alcanzado la última
iteración. De ser así, no
% necesitan crearse nuevas posiciones ni
velocidades.
if iter==itersinmejora
    break
else
    %Obteniendo las posiciones y velocidades para
    el nuevo ciclo (PSO)
    for partic=1:1:particulas
        clampingpso=0;
        for elem=1:1:numeronodos

            % Se obtiene la velocidad anterior

```

```

    velocidad = VEL(partic,elem);

    % Identificando si la partícula
    pertenece al grupo 1 (RMPSO-DL)
    if partic<partporgrupo+1
        phiuno= rand*1.494;
        phidos= rand*1.494;
        wtemp= 0.3;
        W(partic,elem)=wtemp;

        % Para este grupo, la inercia W
        es de 0.30, por lo que:
        G1 = 0.3*velocidad;
        % Componente propio
        G2 =
        phiuno*((OWNBEST(partic,elem))-X(elem,partic));
        % Componente social
        G3 = phidos*(GLOBALBEST(1,elem)-
        X(elem,partic));

    end

    % Identificando si la partícula
    pertenece al grupo 2 (RMPSO-DL)
    if partic>partporgrupo
        if partic< 2*partporgrupo+1

            phiuno= rand*1.494;
            phidos= rand*1.494;
            wtemp= 0;
            W(partic,elem)=wtemp;

            %Para este grupo, la inercia
            = 0; G1=0
            G1=0;
            % Componente propio
            G2 =
            phiuno*((OWNBEST(partic,elem))-X(elem,partic));
            % Componente social

```

```

G3=phidos*(GLOBALBEST(1,elem)-X(elem,partic));

    end
end

    % Identificando si la partícula
pertenece al grupo 3(RMPSO-DL)
    if partic>2*partporgrupo
        if partic<3*partporgrupo+1

            phiuno= 0;
            phidos= rand*1.494;
            wtemp= 0.5 + rand/2;
            W(partic,elem)=wtemp;

            %Para este grupo, la inercia
es variable

            G1= wtemp*velocidad;
            % En este caso, las
partículas no tienen voluntad propia
            G2 = 0;
            % Componente social

G3=phidos*(GLOBALBEST(1,elem)-X(elem,partic));
        end
    end

    % Identificando si la partícula
pertenece al grupo 4(RMPSO-DL)
    if partic>3*partporgrupo
        if partic<4*partporgrupo+1

            phiuno= rand*1.494;
            phidos= rand*1.494;
            wtemp= 0.9;
            W(partic,elem)=wtemp;

```

```

        % Para este grupo, la inercia
W es de 0.90, por lo que:
        G1 = 0.9*velocidad;
        % Componente propio
        G2 =
phiuno*((OWNBEST(partic,elem))-X(elem,partic));
        % Componente social
        G3 =
phidos*(GLOBALBEST(1,elem)-X(elem,partic));

    end
end

```

```

        % Identificando si la partícula
pertenece al grupo 5(RMPSO-DL)
    if partic>4*partporgrupo

        phiuno= rand*1.494;
        phidos= rand*1.494;
        wtemp= 0.5 + rand/2;
        W(partic,elem)=wtemp;

        % Para este grupo, la inercia W es
variable, por lo que:
        G1= wtemp*velocidad;
        % Componente propio
        G2 =
phiuno*((OWNBEST(partic,elem))-X(elem,partic));
        % Componente social
        G3 = phidos*(GLOBALBEST(1,elem)-
X(elem,partic));

    end

```

```

    % Obteniendo la velocidad
    velocidadcalc = G1 + G2 + G3;

```

```

%% Revisión de velocidades

% Se revisa que la velocidad no se
salga del intervalo
%[-Vmax,Vmax].(Swarm Intelligence P.
325)

if velocidadcalc < velmin
    % Si la velocidad calculada es
menor a la velocidad
    % mínima, entonces se establece
que la nueva velocidad será
    % igual a la velocidad mínima.
    velocidadcalc = velmin;
end

if velocidadcalc > velmax
    % Si la velocidad calculada es
mayor a la velocidad
    % máxima, entonces se establece
que la nueva velocidad será
    % igual a la velocidad máxima.
    velocidadcalc = velmax;
end

% Se guardan las nuevas velocidades
calculadas en la matriz
% VEL que se usarán en la próxima
iteración.
VEL(partic,elem)=velocidadcalc;

%% Definiendo nuevas posiciones

% La nueva posición de la partícula
se define mediante la
% fórmula: Nuevaposición = Posición
Anterior + Nueva Velocidad
% (Swarm Intelligence P. 310)

Nuevapos=VEL(partic,elem)+X(elem,partic);

```



```

        % Si, la nueva posición calculada
        diera menor al gasto de demanda,
        % se cambia el valor del gasto al
        gasto de demanda y la
        % velocidad adquiere el valor de cero
        (Clamping PSO).

        % Si, la nueva posición calculada
        diera mayor al máximo valor
        % posible de fuga y demanda, se le
        asigna este valor (Clamping
        % PSO)

        if Nuevapos<QDEM(elem,1)
            Nuevapos=QDEM(elem,1);
            VEL(partic,elem)=0;
            clampingpso=1;
        else
            if Nuevapos>(QDEM(elem,1)+qdif)
                Nuevapos=(QDEM(elem,1)+qdif);
                VEL(partic,elem)=0;
                clampingpso=1;
            else

                end
        end

        % Se almacena la nueva posición en la
        matriz.
        X(elem,partic)= Nuevapos;

        % Se revisa si el gasto calculado es
        mayor a la suma de la
        % demanda en el nodo mas el volumen
        total de fugas.
        if X(elem,partic)> ( QDEM(elem,1)+
        qdif )

```

```

        PENVOL(partic,elem)=
900000*(X(elem,partic)-QDEM(elem,1)- qdif);
    else
        PENVOL(partic,elem)= 0;
    end

    % Se revisa si el gasto calculado es
menor a la demanda
    % en el nodo.
    if X(elem,partic)< QDEM(elem,1)
        PENDEM(partic,elem)=
900000*(QDEM(elem,1)-X(elem,partic));
    else
        PENDEM(partic,elem)= 0;
    end

    % Se almacenan todos los datos
calculados para su análisis
    % posterior:
    PHIUNO(partic,elem)=phiuno;
    PHIDOS(partic,elem)=phidos;
    COMPONENTEPROPIO(partic,elem)=G2;
    COMPONENTESOCIAL(partic,elem)=G3;

end

end

%% Revisando la diversidad RMPSO-DL
    % Revisando diversidad si se llavan EME
iteraciones sin mejora
    % (RMPSO-DL)

    if flag-EME==0
        % Almacena el número de veces que
flag ha sido igual a EME
        countflag=countflag+1;

        flag=1;

```

```

        %Acomodando X para que en la
filas queden partículas y en
        %la columnas queden los nodos
Xinversa=X';

        % Obteniendo la diversidad para
todos los subenjambres

        % Obteniendo promedio de todas
las partículas de cada una
        % de las dimensiones
for d=1:1:numeronodos

PROMEDIOXJ(1,d)=mean(Xinversa(:,d));
end

        % Obteniendo la diferencia entre
cada dimensión de la
        % partícula y el promedio de
todas las partículas de cada
        % dimensión (Xij -
Promedio(Xj))^2 y almacenándolas en
        % DIFCUAD
for d=1:1:numeronodos
    for p=1:1:particulas

DIFCUAD(p,d)=(Xinversa(p,d)-PROMEDIOXJ(1,d))^2;
end
end

        %Obteniendo la suma de (Xij -
Promedio(Xj))^2 de cada
        %partícula y almacenándola en
SUMATORIADOS
        for i=1:1:particulas

sumatoriadostemp=sum(DIFCUAD(i,:));

SUMATORIADOS(i,1)=sumatoriadostemp^0.50;
end

```

```

        % Calculando la diversidad
        diversidad = (1/particulas)*
sum(SUMATORIADOS(:,1));

        %Revisando si la diversidad es
menor a eps
        if diversidad<eps
        % Sirve para imprimir si se han
reinicializado las
        % partículas
        reinicial=1;

        % Cuenta el número de veces que
se han reinicializado las
        % partículas
        countreini=countreini+1;
        % *** Se reinician posiciones
por método del caos ***
        for i=1:1:particulas
            for elem=1:1:numeronodos
                if elem==1
                    F(i,elem)= 0.25;
                    while
F(i,elem)==0.25 || F(i,elem)==0.50 ||
F(i,elem)==0.75
                        F(i,elem) =
rand;
                    end
                    L(i,elem) =0;
                else
L(i,elem)=rand*(4-3.56)+3.56;
                        F(i,elem) =
L(i,elem)*F(i,elem-1)*(1-F(i,elem-1));
                    end

                    X(elem,i)=
F(i,elem)*qdif + QDEM(elem,1);

                    if X(elem,i)>
(QDEM(elem,1)+ qdif)

```

```

                                PENVOL(i,elem)=
900000*(X(elem,i)-QDEM(elem,1)- qdif);
                                else
                                PENVOL(i,elem)=0;
                                end

                                if X(elem,i)<
QDEM(elem,1)
                                PENDEM(i,elem)=
900000*(QDEM(elem,1)-X(elem,i));
                                else
                                PENDEM(i,elem)=0;
                                end

                                end
                                end

                                % *** Se reinician velocidades
por método de Half-Diff ***
                                for i=1:1:particulas
                                for elem=1:1:numeronodos
                                Xmin=QDEM(elem,1);

RANDESPBUSQ(i,elem)=rand*qdif+Xmin;
                                valorvel=
0.50*(RANDESPBUSQ(i,elem)-X(elem,i));
                                VEL(i,elem)=
valorvel;

                                end
                                end
                                end

                                %% Mejor global por distribución gaussiana
if flag>eme-1
                                if rand < pm
                                for dim=1:1:numeronodos
                                xmax=QDEM(dim,1)+qdif;
                                xmin=QDEM(dim,1);
                                sigma=0.1*(xmax-xmin)+xmin;

```

```

                                % Se obtiene un nuevo mejor
global "probable".

valorcalc=GLOBALBEST(1,dim)*(1+randn*sigma);

                                % Se revisa si el valor
calculado está dentro de los
                                % límites. Ésto se hace como
una aplicación de CLAMPING
                                % PSO, además así se evitará
generar una penalización
                                % PENVOL y PENDEM para esta
opción.

                                if valorcalc>xmax
                                    valorcalc=xmax;
                                end

                                if valorcalc<xmin
                                    valorcalc=xmin;
                                end

                                BESTGAUSS(1,dim)=valorcalc;
end

                                % ***** EVALUANDO
HIDRÁULICAMENTE EL NUEVO MEJOR PROBABLE
*****

                                % (Ver explicación en apartado
Generando reporte
                                % PSO_Reporte_Final.txt)

                                JUNCT(:,3)=BESTGAUSS(1,:)';

file=fopen('PSO_Entrada.txt','wt');
fprintf(file, '[TITLE]\n');
fprintf(file, 'PSO GAUSS \n');
fprintf(file, '\n');
fprintf(file, '[RESERVOIR]\n');
[f,~]= size(RESERV);
for fila=1:f

```

```

        fprintf(file, '%d %0.2f \n',
RESERV(fila,:));
    end
    fprintf(file, '\n');
    fprintf(file, '[JUNCTIONS]\n');
    [f,~]= size(JUNCT);
    for fila=1:f
        fprintf(file, '%d %0.2f %0.2f
\n', JUNCT(fila,:));
    end
    fprintf(file, '\n');
    fprintf(file, '[PIPES]\n');
    [f,~]= size(PIPE);
    for fila=1:f
        fprintf(file, '%d %d %d %0.2f
%d %d \n', PIPE(fila,:));
    end

    fprintf(file, '\n');
    fprintf(file, '[COORDINATES]\n');

    [f,~]= size(COORD);
    for fila=1:f
        fprintf(file, '%d %0.2f %0.2f
\n', COORD(fila,:));
    end

    fprintf(file, '\n');
    fprintf(file, '[OPTIONS]\n');
    fprintf(file, 'UNITS LPS\n');
    fprintf(file, 'HEADLOSS H-W\n');
    fprintf(file, '\n');
    fprintf(file, '[REPORT]\n');
    fprintf(file, 'NODES All\n');
    fprintf(file, 'LINKS All\n');
    fprintf(file, '\n');
    fprintf(file, '[END]\n');
    fclose('all');

```

```

dos('C:\MATLAB\PSO_JPRG\epanet2d.exe
PSO_Entrada.txt PSO_Salida.txt');

%***** EXTRAYENDO DATOS
REPORTE EPANET*****
% (Ver explicación en Obteniendo
Presiones del archivo
% calculado por EPANET)

[m,~]=size(JUNCT);
lineult = 43 + m;
linea=0;
TEMP=zeros(m,4);
lineiniarranq=44;
for
lineini=lineiniarranq:1:lineult
    linea=linea+1;
    fid=fopen('PSO_Salida.txt');
    r=textscan(fid, '%f', 4,
'delimiter','' , 'headerlines', lineini-1);
    a=cell2mat(r);
    isemptymatrix=isempty(a);
    if isemptymatrix==1
        break;
    end
    TEMP(linea,:)=a';
    fclose('all');
end

if isemptymatrix==1
    lineult = 45 + m;
    linea=0;
    TEMP=zeros(m,4);
    lineiniarranq=46;
    fid=fopen('PSO_Salida.txt');
    r=textscan(fid, '%f', 4,
'delimiter','' , 'headerlines', lineini-1);
    a=cell2mat(r);

```



```

                                for
lineini=lineiniarranq:1:lineult
                                linea=linea+1;

fid=fopen('PSO_Salida.txt');
                                r=textscan(fid, '%f', 4,
'delimiter',' ' , 'headerlines', lineini-1);
                                a=cell2mat(r);
                                TEMP(linea,:)=a';
                                fclose('all');
                                end
                                end
                                PRESPSOACUMGAUSS(1,:)=TEMP(:,4);
                                PRESTEO=load('PRESIONES.TXT');
                                fclose('all');
                                DIFPRES=PRESTEO-PRESPSO;

                                %*****          CALCULANDO FA
*****

                                % (Ver explicación en sección
"Calculando el Factor de
                                % Aptitud")
                                [m,~]=size(DIFPRES);
                                DIFPRES=DIFPRES.^2;

                                %Aquí se elimina PENVOL y PENDEM
porque se revisó que los
                                %gastos estuvieran entre los
límites permitidos
                                aptitudparticula =
(sum(DIFPRES)+0+0)/m;
                                fagauss=aptitudparticula;

                                %*****          REVISANDO SI SE OBTUVO
UN MEJOR GLOBAL *****
                                if fagauss<min(min(FAOWNBEST))

                                modbygauss=1;
                                % Se almacena la
configuración de gastos como la mejor

```

```

% que se ha obtenido en la
rutina.

GLOBALBEST(1,:)=BESTGAUSS(1,:);

%Se almacena la configuración
de gastos en la partícula
%1 como mejor alcanzado, el
FA como propio y la
%velocidad se pone en cero.
OWNBEST(1,:)=BESTGAUSS(1,:);
FAOWNBEST(1,1)=fagauss;
FA(1,1)=fagauss;
VEL(1,:)= 0;
X(:,1)=BESTGAUSS(1,:);

% Se resetea el contador a
uno, se tendrán que realizar
% otras "itersinmejora"
iteraciones para que se acabe la rutina.
iter=1;

% Como se obtuvo un mejor
global, se resetea el valor
% de flag a cero RMPSO-DL
flag=1;

% Indica que se ha obtenido
un nuevo mejor global.
cambio=1;

% Se almacena en qué
iteración se obtuvo el mejor
% resultado
bestiter= contiter;

% Sirve para saber si se ha
sustituído el mejor gobal con la
% distribución gaussiana.
countpgprima=countpgprima+1;

```

```

        end

    end

    %% Imprimiendo en TXT los datos de esta
    iteración.

    if guardardatos=='s' || guardardatos=='S'
        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo,'a');

        fprintf(file, '\n');
        fprintf(file, '
PRESPSO_(Presiones_calculadas_con_EPANET_a_partir
_de_las_configuraciones_de_gasto_X)\n');
        save (nomarchivo,'PRESPSOACUM', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo,'a');
        fprintf(file, '\n');
        if modbygauss==0
            fprintf(file, '
FA_(Factor_de_aptitud_de_la_configuración_de_gast
o_seleccionada_por_la_partícula_X)\n');
        else
            fprintf(file, '
FA_(Factor_de_aptitud_de_la_configuración_de_gast
o_seleccionada_por_la_partícula_X)---
EN_LA_PARTÍCULA_1_SE_ALMACENÓ_LA_ENCONTRADA_POR_G
AUSS\n');
        end
        save (nomarchivo,'FA', '-ascii', '-append',
'-v6');
        fclose (file);
    end

```

```

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        if modbygauss==0
            fprintf(file, '
FAOWNBEST_(Mejor_factor_de_aptitud_alcanzada_por_
esa_partícula)\n');
        else
            fprintf(file, '
FAOWNBEST_(Mejor_factor_de_aptitud_alcanzada_por_
esa_partícula)---
EN_LA_PARTÍCULA_1_SE_ALMACENÓ_LA_ENCONTRADA_POR_G
AUSS\n');
        end
        save (nomarchivo, 'FAOWNBEST', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        if modbygauss==0
            fprintf(file, '
OWNBEST_(Mejor_configuración_de_gastos_alcanzada_
por_esa_partícula)\n');
        else
            fprintf(file, '
OWNBEST_(Mejor_configuración_de_gastos_alcanzada_
por_esa_partícula)---
EN_LA_PARTÍCULA_1_SE_ALMACENÓ_LA_ENCONTRADA_POR_G
AUSS\n');
        end

        save (nomarchivo, 'OWNBEST', '-ascii', '-
append', '-v6');
        fclose (file);

```

```

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo,'a');
        fprintf(file, '\n');
        if modbygauss==0
            fprintf(file, '
GLOBALBEST_(Mejor_configuración_de_gastos_alcanza
da_al_momento_por_cualquier_partícula)\n');
        else
            fprintf(file, '
GLOBALBEST_MODIFICADA_POR_GAUSS_(Mejor_configurac
ión_de_gastos_alcanzada_al_momento:_Obtenida_alea
toriamente_por_Gauss)\n');
        end
        save (nomarchivo,'GLOBALBEST', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo,'a');
        fprintf(file, '\n');
        if reinicial==1
            fprintf(file,
'*****_Iteración_número_%d_*****;___Fla
g=%d,_Se_reinicializaron_posiciones_y_velocidades
\n', (contiter+1),flag);
        else
            fprintf(file,
'*****_Iteración_número_%d_*****;___Fla
g=%d\n', (contiter+1),flag);
        end

        fprintf(file, '
PHIUNO_(Valores_componente_personal)\n');
        save (nomarchivo,'PHIUNO', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];

```

```

        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        fprintf(file, '
PHIDOS_ (Valores_componente_social)\n');
        save (nomarchivo, 'PHIDOS', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        fprintf(file, ' COMPONENTE_PROPIO\n');
        save (nomarchivo, 'COMPONENTEPROPIO', '-
ascii', '-append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        fprintf(file, ' COMPONENTE_SOCIAL\n');
        save (nomarchivo, 'COMPONENTESOCIAL', '-
ascii', '-append', '-v6');
        fclose (file);

        if reinicial==1
            file = fopen( nomarchivo, 'a');
            fprintf(file, '
L_ (Factor_de_inicialización_para_posiciones._Méto
do_del_caos)\n');
            save (nomarchivo, 'L', '-ascii', '-
append', '-v6');
            fprintf(file, '\n');
            close 'all'

            nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
            file = fopen( nomarchivo, 'a');
            fprintf(file, ' F_ (Factor_del_caos)\n');

```

```

        save (nomarchivo, 'F', '-ascii', '-
append', '-v6');
        fprintf(file, '\n');
        close 'all'
    end

    Xinversa=X';
    nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
    file = fopen( nomarchivo, 'a');
    fprintf(file, '\n');
    if reinicial==1
        fprintf(file, '
X_(Valores_de_gasto_REINICIALIZADOS)\n');
    else
        if modbygauss==1
            fprintf(file, '
X_(Valores_de_gasto)---
EN_LA_PARTÍCULA_1_SE_ALMACENÓ_LA_ENCONTRADA_POR_G
AUSS\n');
        else
            if clampingpso==0
                fprintf(file, '
X_(Valores_de_gasto)\n');
            else
                fprintf(file, '
X_(Valores_de_gasto)_Se_modificó_por_CLAMPING_PSO
\n');
            end
        end
    end

    save (nomarchivo, 'Xinversa', '-ascii', '-
append', '-v6');
    fclose (file);

    nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
    file = fopen( nomarchivo, 'a');
    fprintf(file, '\n');
    fprintf(file, '
PENVOL_(Penalización_Volumen)\n');

```

```

        save (nomarchivo, 'PENVOL', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        fprintf(file, '
PENDEM_ (Penalización_Demanda)\n');
        save (nomarchivo, 'PENDEM', '-ascii', '-
append', '-v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        fprintf(file, ' W (Inercia)\n');
        save (nomarchivo, 'W', '-ascii', '-append', '-
v6');
        fclose (file);

        nomarchivo = ['PSO_Datos_Analisis'
num2str(sufijo) '.txt'];
        file = fopen( nomarchivo, 'a');
        fprintf(file, '\n');
        if reinicial==1
            fprintf(file, '
VEL_ (Velocidades_REINICIALIZADAS)\n');
        else
            if modbygauss==1
                fprintf(file, '
VEL_ (Velocidades)---
LAS_DE_LA_PRIMERA_PARTÍCULA_SON_CERO_POR_GAUSS\n'
);
            end

            if clampingpso==0

```



```

        fprintf(file, '
VEL_(Velocidades)\n');
    else
        fprintf(file, '
VEL_(Velocidades)_Se_modificó_por_CLAMPING_PSO\n'
);
    end
end
save (nomarchivo, 'VEL', '-ascii', '-append',
'-v6');
fclose (file);
end

end
end
%Termina el ciclo de iteraciones

%% Generando reporte PSO_Reporte_Final.txt

% Se almacena la mejor configuración de red en
la matriz JUNCT. (Se usa el
% inverso (') porque la matriz GLOBALBEST tiene
la configuración en la línea, y
% JUNCT tiene la configuración de red en columna
JUNCT(:,3)=GLOBALBEST(1,:));

% Se crea el archivo llamado
"REPORTE_PSO_FINAL.txt", la 'w'
% indica que, si ya existe un archivo llamado
así, se sobrescribirá.
% Se hará referencia a este archivo mediante la
variable "file"
file=fopen('PSO_Entrada.txt','wt');

% Se imprime en la primera línea "[TITLE]". El
parámetro "\n"
% indica que ahí termina una línea y genera una
nueva para la
% siguiente impresión
fprintf(file, '[TITLE]\n');

```

```

fprintf(file, 'PSO Mejor configuración alcanzada
\n');

% Genera una línea en blanco
fprintf(file, '\n');

fprintf(file, '[RESERVOIR]\n');

% Calcula el número de filas de la matriz RESERV
y lo almacena en
% f, el número de columnas se desprecia (~)
[f,~]= size(RESERV);

% Se realiza un ciclo "for" en el que se
obtendrán todos los datos
% de la matriz RESERV y se escriben en el
archivo txt. La variable
% "file" hace alusión al archivo que se está
creando. Se importarán
% 2 valores por fila "%d" y "%0.2f". El primero
(%d) hace alusión a
% un número entero, el segundo (%0.2f) indica
que es un número con
% dos decimales. El parámetro "\n" indica que se
debe concluir esa
% línea y empezar en una nueva.
for fila=1:f
    fprintf(file, '%d %0.2f \n', RESERV(fila,:));
end

fprintf(file, '\n');
fprintf(file, '[JUNCTIONS]\n');

% Ciclo para escribir en el archivo TXT los
valores de la matriz JUNCT
[f,~]= size(JUNCT);
for fila=1:f
    fprintf(file, '%d %0.2f %0.2f \n',
JUNCT(fila,:));
end

```

```

fprintf(file, '\n');
fprintf(file, '[PIPES]\n');

% Ciclo para escribir en el archivo TXT los
valores de la matriz PIPE
[f,~]= size(PIPE);
for fila=1:f
    fprintf(file, '%d %d %d %0.2f %d %d \n',
PIPE(fila,:));
end

fprintf(file, '\n');
fprintf(file, '[COORDINATES]\n');

% Se obtienen el número de filas de la matriz
COORD, las columnas
% se desprecian (~)
[f,~]= size(COORD);

% Ciclo para escribir en el archivo TXT los
valores de la matriz
% COORD
for fila=1:f
    fprintf(file, '%d %0.2f %0.2f \n',
COORD(fila,:));
end

fprintf(file, '\n');
fprintf(file, '[OPTIONS]\n');
fprintf(file, 'UNITS LPS\n');
fprintf(file, 'HEADLOSS H-W\n');
fprintf(file, '\n');
fprintf(file, '[REPORT]\n');
fprintf(file, 'NODES All\n');
fprintf(file, 'LINKS All\n');
fprintf(file, '\n');
fprintf(file, '[END]\n');

```

```

    % Cierra todos los archivos externos cargados
para evitar consumo
    % de memoria.
    fclose('all');

    % ***** Obteniendo datos hidráulicos de mejor
opción en EPANET *****

    % Evaluando el archivo creado en EPANET, el
nombre del archivo de
    % entrada es PSO_Entrada.txt, los resultados se
exportarán en un
    % archivo TXT que se creará con el nombre
PSO_Salida.txt
    dos('C:\MATLAB\PSO_JPRG\epanet2d.exe
PSO_Entrada.txt PSO_Reporte_Final.txt');

    % ***** Anotando el nombre del ejercicio en
el archivo EPANET *****
    file = fopen('PSO_Reporte_Final.txt','a');
    fprintf(file,'Ejercicio: %s \n ',
datestr(tiempo));
    fprintf(file,'\n');
    fclose(file);

    %% Generando Reporte
PSO_Información_Adicional.txt

nomarchivo = ['PSO_Información_Adicional'
num2str(sufijo) '.txt'];
file = fopen(nomarchivo,'wt');
    fprintf(file,'Ejercicio: %s \n ',
datestr(tiempo));
    fprintf(file,'\n');
    fprintf(file,'Rutina PSO para detección de fugas
en una Red de Agua Potable\nAutor: Juan Pablo
Ramírez Gutiérrez. E-mail:jprg1981@gmail.com
\n\n');
    fprintf(file,'**** Información Adicional de la
Rutina *****\n\n');
    fprintf(file,'**** PSO **** \n');

```

```

    fprintf(file, 'Número de Partículas: %d
\n',particulas);
    fprintf(file, 'Iteraciones alcanzadas: %d \n',
contiter);
    fprintf(file, 'Iteraciones sin mejora necesarias
para terminar rutina: %d \n',itersinmejora);
    fprintf(file, 'Mejor valor de aptitud alcanzado:
%0.9f \n',min(min(FAOWNBEST)));
    fprintf(file, 'Iteración en la que se alcanzó el
mejor FA: %d \n', bestiter);
    fprintf(file, 'Velocidad máxima de vuelo
permitida: %0.2f \n', velmax);
    fprintf(file, 'Velocidad mínima de vuelo
permitida: %0.2f \n', velmin);
    fprintf(file, 'Velocidad máxima de vuelo
alcanzada: %0.2f \n', max(max(VEL)));
    fprintf(file, 'Velocidad mínima de vuelo
alcanzada: %0.2f \n', min(min(VEL)));
    fprintf(file, 'Tiempo de cálculo: %0.2f
segundos\n', T );
    fprintf(file, '\n');
    fprintf(file, '**** RED **** \n');
    fprintf(file, 'Número de Nodos en la red: %d
\n',numeronodos );
    fprintf(file, 'Gasto de entrada: %0.2f lps \n',
qentr);
    fprintf(file, 'Demanda en los nodos: %0.2f lps
\n', qsum);
    fprintf(file, 'Pérdidas: %0.2f lps \n',qdif);
    fprintf(file, 'Número de redes analizadas: %d
\n', contiter*partic);
    fclose (file);

    nomarchivo = ['PSO_Información_Adicional'
num2str(sufijo) '.txt'];
    file = fopen( nomarchivo, 'a');
    fprintf(file, '\n');
    fprintf(file, '\n');
    fprintf(file, ' Mejor configuración de gastos
alcanzada\n');

```

```

    save (nomarchivo, 'GLOBALBEST', '-ascii', '-
append', '-v6');
    fclose (file);

    nomarchivo = ['PSO_Información_Adicional'
num2str(sufijo) '.txt'];
    file = fopen( nomarchivo, 'a');
    fprintf(file, '\n');
    fprintf(file, ' ** Datos RMP SO-DL\n');
    fprintf(file, ' Valor de M=%0.2f, Valor de
m=%0.2f, Valor de eps=%0.2f, Valor de
Pm=%0.2f\n', EME, eme, eps, pm);
    fprintf(file, ' División en sub-enjambres:\n');
    fprintf(file, '         Primer sub-enjambre: De
partícula 1 a %d\n', partporgrupo);
    fprintf(file, '         Segundo sub-enjambre: De
partícula %d a %d\n',
partporgrupo+1, 2*partporgrupo);
    fprintf(file, '         Tercer sub-enjambre: De
partícula %d a %d\n',
2*partporgrupo+1, 3*partporgrupo);
    fprintf(file, '         Cuarto sub-enjambre: De
partícula %d a %d\n',
3*partporgrupo+1, 4*partporgrupo);
    fprintf(file, '         Quinto sub-enjambre: De
partícula %d a %d\n',
4*partporgrupo+1, particulas);
    fprintf(file, '\n');
    fprintf(file, ' Reinicializaciones de partículas
a lo largo de la rutina: %0.2f\n', countreini);
    fprintf(file, ' Ocasiones en las que flag
alcanzo las %d iteraciones: %0.2f\n', EME,
countflag);
    fprintf(file, ' Veces en las que se encontró un
mejor resultado por Gauss: %0.2f\n',
countpgprima);
    fclose (file);

```

%% Resultados PSO en Pantalla

```

clc
% Se imprime el resumen de la rutina.
fprintf('*** Fin de la rutina PSO para detección
de fugas en una Red de Agua Potable*** \nAutor:
Juan Pablo Ramírez Gutiérrez. E-
mail:jprg1981@gmail.com \n\n');
fprintf( 'El mejor valor tuvo un factor de
aptitud de:  %0.9f, se alcanzó en la iteración %d
\n',min(min(FAOWNBEST)), bestiter);
fprintf( 'Se llegó a las %d iteraciones.
',contiter);
fprintf('Tiempo de cálculo: %0.2f segundos\n', T
);
fprintf( '\n');
fprintf( 'Los resultados de la mejor
configuración se exportaron al archivo
"PSO_Reporte_Final.txt"\n');
fprintf( 'Las características de esta corrida se
exportaron al archivo
"PSO_Información_Adicional.txt"\n');
fprintf( 'Todas las configuraciones de red y
factores de PSO se exportaron al archivo
"PSO_Datos_Análisis.txt"\n');
if graficardatos=='s' || graficardatos=='S'
    fprintf( 'Se creó y guardó la siguiente
gráfica "PSO_Vuelo.jpeg"\n');
end
nomarchivo = ['PSO_Resumen_de_todas_las_rutinas'
num2str(sufijo42) '.txt'];
file = fopen( nomarchivo, 'a');

fprintf(file, '%s %d %0.2f %0.9f %d',
datestr(tiempo),particulas ,T
,min(min(FAOWNBEST)), contiter);
fprintf(file, '%s %d %0.2f %0.9f %d %e',
datestr(tiempo),particulas ,T
,min(min(FAOWNBEST)), contiter, GLOBALBEST);
fprintf(file, '\n');
fclose('all');

```

```
end  
%Fin de rutina.
```




VOTO DE APROBACIÓN DE DEFENSA DE TESIS

Guanajuato, Gto. a: DÍA /MES / AÑO
18/02/2026

Programa de posgrado: Maestría en Ciencias del Agua

Nombre del alumno: Juan Pablo Ramírez Gutiérrez

Título del trabajo: DETECCIÓN Y ESTIMACIÓN DE FUGAS EN REDES DE ABASTECIMIENTO DE AGUA POTABLE MEDIANTE LA METODOLOGÍA PARTICLE SWARM

Después de revisar el trabajo de titulación resuelvo mi voto para la defensa como:

- ☒ Aprobada
- ☐ Aprobada con las observaciones que se anexan
- ☐ No aprobada: anexar los motivos de su decisión para informar al interesado

Evaluación de aspectos específicos de la tesis: 0 (pésima); 1 (muy mala); 2 (mala); 3 (aceptable); 4 (buena); 5 (excelente).

	5	4	3	2	1	0
Aporte y originalidad	☐	☐	☒	☐	☐	☐
Impacto, pertinencia y relevancia	☐	☐	☒	☐	☐	☐
Validez de la hipótesis	☐	☐	☐	☒	☐	☐
Metodología	☐	☒	☐	☐	☐	☐
Resultados y su análisis	☐	☒	☐	☐	☐	☐
Referencias bibliográficas: actualidad y vigencia	☐	☐	☐	☒	☐	☐
Organización del documento	☐	☒	☐	☐	☐	☐

Quedo enterado de que formaré parte del jurado en el lugar, fecha y hora que se me indique posteriormente.

NOMBRE Y FIRMA DEL EVALUADOR:

DR. ERIK AUGUSTO VILLAGÓMEZ REYES

Este documento debe expedirse dentro de los treinta días siguientes a la fecha en que haya sido notificado como miembro del tribunal y recibido a su consideración la tesis respectiva.

Enviar este documento debidamente firmado a la coordinación del posgrado (en original o bien en formato digital al correo que corresponda: dcta@ugto.mx; maestriacienciasagua@ugto.mx. División de Ingenierías sede Belén. Av. Juárez 77, zona centro. Para más información sobre el proceso y trámites administrativos: +52 473 10 20 100 Ext. 2293.



VOTO DE APROBACIÓN DE DEFENSA DE TESIS

Guanajuato, Gto. a: DÍA /MES / AÑO
26/02/2026

Programa de posgrado: Maestría en Ciencias del Agua

Nombre del alumno: Juan Pablo Ramírez Gutiérrez

Título del trabajo: Detección y estimación de fugas en redes de abastecimiento de agua potable mediante la metodología Particle Swarm Optimization (PSO) y EPANET

Después de revisar el trabajo de titulación resuelvo mi voto para la defensa como:

- ☒ Aprobada
- ☐ Aprobada con las observaciones que se anexan
- ☐ No aprobada: anexar los motivos de su decisión para informar al interesado

Evaluación de aspectos específicos de la tesis: 0 (pésima); 1 (muy mala); 2 (mala); 3 (aceptable); 4 (buena); 5 (excelente).

	5	4	3	2	1	0
Aporte y originalidad	☒	☐	☐	☐	☐	☐
Impacto, pertinencia y relevancia	☐	☒	☐	☐	☐	☐
Validez de la hipótesis	☒	☐	☐	☐	☐	☐
Metodología	☒	☐	☐	☐	☐	☐
Resultados y su análisis	☒	☐	☐	☐	☐	☐
Referencias bibliográficas: actualidad y vigencia	☐	☒	☐	☐	☐	☐
Organización del documento	☒	☐	☐	☐	☐	☐

Quedo enterado de que formaré parte del jurado en el lugar, fecha y hora que se me indique posteriormente.

NOMBRE Y FIRMA DEL EVALUADOR: *Xitlali Delgado*

Este documento debe expedirse dentro de los treinta días siguientes a la fecha en que haya sido notificado como miembro del tribunal y recibido a su consideración la tesis respectiva.

Enviar este documento debidamente firmado a la coordinación del posgrado (en original o bien en formato digital al correo que corresponda: dcta@ugto.mx; maestriacienciasagua@ugto.mx. División de Ingenierías sede Belén. Av. Juárez 77, zona centro. Para más información sobre el proceso y trámites administrativos: +52 473 10 20 100 Ext. 2293.



VOTO DE APROBACIÓN DE DEFENSA DE TESIS

Guanajuato, Gto. a: DÍA /MES / AÑO
24/02/2026

Programa de posgrado: Maestría en Ciencias del Agua

Nombre del alumno: Juan Pablo Ramírez Gutiérrez

Título del trabajo: Detección y Estimación de Fugas en Redes de Abastecimiento de Agua Potable Mediante la Metodología Particle Swarm Optimization (PSO) y Epanet

Después de revisar el trabajo de titulación resuelvo mi voto para la defensa como:

- ☒ Aprobada
- ☐ Aprobada con las observaciones que se anexan
- ☐ No aprobada: anexar los motivos de su decisión para informar al interesado

Evaluación de aspectos específicos de la tesis: 0 (pésima); 1 (muy mala); 2 (mala); 3 (aceptable); 4 (buena); 5 (excelente).

	5	4	3	2	1	0
Aporte y originalidad	☒	☐	☐	☐	☐	☐
Impacto, pertinencia y relevancia	☒	☐	☐	☐	☐	☐
Validez de la hipótesis	☒	☐	☐	☐	☐	☐
Metodología	☒	☐	☐	☐	☐	☐
Resultados y su análisis	☒	☐	☐	☐	☐	☐
Referencias bibliográficas: actualidad y vigencia	☒	☐	☐	☐	☐	☐
Organización del documento	☒	☐	☐	☐	☐	☐

Quedo enterado de que formaré parte del jurado en el lugar, fecha y hora que se me indique posteriormente.

NOMBRE Y FIRMA DEL EVALUADOR:

Dr. Daniel Hernández Cervantes

Este documento debe expedirse dentro de los treinta días siguientes a la fecha en que haya sido notificado como miembro del tribunal y recibido a su consideración la tesis respectiva.

Enviar este documento debidamente firmado a la coordinación del posgrado (en original o bien en formato digital al correo que corresponda: dcta@ugto.mx; maestriacienciasagua@ugto.mx. División de Ingenierías sede Belén. Av. Juárez 77, zona centro. Para más información sobre el proceso y trámites administrativos: +52 473 10 20 100 Ext. 2293.



«30 años de autonomía universitaria, un legado de
responsabilidad y libertad»

Guanajuato, Gto., 27 de febrero de 2026.
Asunto: Cumplimiento de requisitos para
obtención de grado.

LIC. JUAN BARDO RODRÍGUEZ DE LA VEGA
CORDINACIÓN DE ASUNTO ESCOLARES
CAMPUS GUANAJUATO
UNIVERSIDAD DE GUANAJUATO
P R E S E N T E.

El que suscribe, Director de la División de Ingenierías de la Universidad de Guanajuato, hace constar que el alumno **JUAN PABLO RAMÍREZ GUTIÉRREZ**, con NUA: **198269**, ha cumplido íntegramente con los requisitos Académico-Administrativos, incluyendo el requerimiento del idioma extranjero necesario para que le sea autorizado la sustentación de su Obtención del Grado de **MAESTRÍA EN CIENCIAS DEL AGUA**, bajo la modalidad de **TESIS**.

Sin otro particular, aprovecho la oportunidad para reiterarle las seguridades de mi consideración distinguida.

Atentamente

“La Verdad os Hará Libres”

El Director

Universidad de Guanajuato
División de Ingenierías
Campus Guanajuato
DIRECCIÓN

Dr. Gilberto Carreño Aguilera

DR.GCA/mlmA

CAMPUS GUANAJUATO
DIVISIÓN DE INGENIERÍAS

Sede Belén: Av. Juárez 77; Centro. C.P.: 36,000; Guanajuato, Gto.
Sede San Matías: Ex hacienda de San Matías s/n; Col. San Javier. C.P.: 36020;
Guanajuato, Gto.

Teléfono: 473 102 0100 Ext.: 2212
Teléfono: 473 732 7277 / 473 732 7278