

**EVALUACIÓN DEL MÉTODO DE COLONIA DE HORMIGAS PARA DISEÑOS
DE SISTEMAS DE DISTRIBUCIÓN DE AGUA**

JESSIKA TATIANA ORTÍZ APONTE

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA CIVIL
BUCARAMANGA**

2013

**EVALUACIÓN DEL MÉTODO DE COLONIA DE HORMIGAS PARA DISEÑOS
DE SISTEMAS DE DISTRIBUCIÓN DE AGUA**

JESSIKA TATIANA ORTÍZ

**Trabajo de Grado para optar por el título de:
Ingeniero Civil**

Director:

OSCAR JAVIER BEGAMBRE CARRILLO

ingeniero civil, M. Sc, PhD.

Co- director

MARIO GARCÍA SOLANO

Ingeniero civil

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA CIVIL
BUCARAMANGA**

2013

DEDICATORIA

***•A mis padres por toda la ayuda y confianza que depositaron en mí
durante todos estos años.***

***•A mis hermanos por ser mis compañeros y amigos durante toda la vida
en especial a mi hermano Alexander que me acompañó y me guió
durante mi vida universitaria.***

***•A todos mis amigos y seres queridos por hacer de esta etapa de mi vida
una experiencia gratificante e inolvidable.***

Jessika Tatiana Ortiz Aponte

AGRADECIMIENTOS

A la Universidad Industrial de Santander por la formación académica y social durante mi carrera.

A los profesores Oscar Begambre y Luis Fernando Castañeda por su orientación en este proyecto de grado.

A mi hermano Rony Alexander Ortiz Aponte por su incondicional apoyo en todos los momentos de mi vida.

CONTENIDO

INTRODUCCIÓN	16
1. PRELIMINARES	18
1.1 METODOS DE OPTIMIZACIÓN MATEMATICA	18
2. JUSTIFICACIÓN.....	20
2.1 FORMULACIÓN DEL PROBLEMA.....	20
2.2 OBJETIVOS.....	20
2.2.1 Objetivo general	20
2.2.2 Objetivos específicos	21
3. MARCO CONCEPTUAL.....	22
3.1 LAS COLONIAS DE HORMIGAS.....	22
3.2 METAHEURÍSTICA DE OPTIMIZACIÓN MEDIANTE COLONIA DE HORMIGAS.....	23
3.2.1 CARACTERÍSTICAS DE LAS HORMIGAS ARTIFICIALES	26
3.3 ALGORITMOS UTILIZADOS DE LA METAHEURISTICA OCH.....	27
3.3.1 Algoritmo sistema de hormigas	27
3.3.2 Ejemplo de aplicación	30
3.3.1 Algoritmo máximos y mínimos sistema de hormigas.....	38
3.4 ANALISIS DE REDES DE TUBERÍAS: REDES CERRADAS	41
3.4.1 Principios fundamentales de análisis de redes cerradas.....	41
3.4.2 METODOS TRADICIONALES DE DISEÑO EN LAS RDA ´S.....	45
3.4.3 EPANET 2.0 “TOOLKIT”	48
3.4.4 LLAMAR A EPANET DESDE MATLAB.....	49
4. APLICACIÓN DEL ALGORITMO OCH-EPANET A LAS RDA ´S	51

4.1 LAS RDA's COMO UN PROBLEMA DE OPTIMIZACIÓN.	51
4.2 EJEMPLO RED DE HANOI	57
4.2.1 Descripción de la red.	57
5. VALIDACIÓN DEL PROGRAMA Y RESULTADOS.	61
5.1 RESULTADOS.	62
5.1.1 RESULTADOS DEL ALGORITMO SH.	62
5.1.2 Resultados de la variación de <i>Pbest</i> con un $d= 0.5$.	65
5.1.3 Resultados Del Algoritmo MMSH.	67
5.1.3.1 Resultados MMSH sin tener en cuenta la actualización de la feromona de la mejor solución global	67
5.1.3.2 Resultados MMSH con la actualización de la feromona de la mejor solución global	76
5.1.4 ANALISIS DE RESULTADOS	80
6. CONCLUSIONES	83
7.RECOMENDACIONES Y TRABAJOS FUTUROS	88
8. BIBLIOGRAFÍA	90
ANEXO A CÓDIGO OCH-EPANET PARA EL PROBLEMA DEL AGENTE VIAJERO	93
ANEXO B CÓDIGO OCH-EPANET PARA LAS RDA'S	96

LISTA DE TABLAS

Tabla 1 Lista de ciudades con sus coordenadas ejemplo tsp.	32
Tabla 2 Datos de entrada ejemplo tsp	32
Tabla 3 Matriz de distancias dij ejemplo tsp	33
Tabla 4 Matriz de visibilidad nij ejemplo tsp	33
Tabla 5 Aplicación de la función de probabilidad para la hormiga h1 en t=0 ejemplo tsp.....	34
Tabla 6 Matriz de la función de probabilidad para las 6 hormigas en t=0 ejemplo tsp	34
Tabla 7 Lista tabú en la primera iteración para cada hormiga k ejemplo tsp	35
Tabla 8 Acumulado de la lista tabú en la primera iteración para cada hormiga k ejemplo tsp.....	35
Tabla 9 Delta de feromonas ejemplo tsp	36
Tabla 10 Actualización de feromonas ejemplo tsp	37
Tabla 11 Lista tabú final para cada hormiga k ejemplo tsp	37
Tabla 12 Acumulado final para cada hormiga k ejemplo tsp.....	38
Tabla 13 Valores recomendados para el coeficiente de rugosidad.	44
Tabla 14 Resultados de epanet 2.0 para los nodos de la red de hanoi.	47
Tabla 15 Resultados de epanet 2.0 para las tuberías de la red de hanoi.....	47
Tabla 16 Demanda de los nodos de la red de hanoi,(el depósito tiene una altura de 100 m).....	59
Tabla 17 Longitudes de las tuberías en la red de hanoi.(todas las tuberías tienen una coeficiente de rugosidad de 130)	60
Tabla 18 Diámetros comerciales y costos para la red de hanoi.....	60
Tabla 19 Resultados del algoritmo sh.	63
Tabla 20 Probabilidades iniciales del algoritmo sh.....	64
Tabla 21 Probabilidades finales del algoritmo sh.	64
Tabla 22 Resultado de la variación del parámetro $d=0.05$	69

Tabla 23 Resultado de la variación del parámetro $d=0.005$	69
Tabla 24 Resultado de la variación del parámetro $d=0.0005$	70
Tabla 25 Resultado de la variación del parámetro $d=0.0000005$	70
Tabla 26 Resultado de la variación del parámetro $d=0.05$ con la actualización de la feromona de la mejor solución global.	77
Tabla 27 Resultado de la variación del parámetro $d=0.005$ con la actualización de la feromona de la mejor solución global.	77
Tabla 28 Resultado de la variación del parámetro $d=0.005$ con la actualización de la feromona de la mejor solución global.	78
Tabla 29 Comparación del mínimo costo encontrado por och-epanet y mmsh en [1]	84
Tabla 30 Parámetros de la mejor solución encontrada del algoritmo mmsh	85
Tabla 31 Comparación de och-epanet y diseño tradicional en epanet 2.0.	86

LISTA DE FIGURAS

Figura 1 Comportamiento de las hormigas reales.	23
Figura 2 Ubicaciones en el plano de las ciudades a, b, c, d, e, f ejemplo tsp	31
Figura 3 Caudales consumidos en los nodos y caudales de alimentación a la red.	42
Figura 4 Red de hanoi en epanet 2.0	46
Figura 5 Representación grafica de la optimización de distribución de agua.	54
Figura 6 Red de hanoi.	59
Figura 7 Comportamiento del algoritmo sistema de hormigas costo (us \$) vs número de iteraciones	62
Figura 8 Presiones del algoritmo sistema de hormigas en cada nodo.....	63
Figura 9 Comportamiento del algoritmo mmsh con un pbest=0.1	65
Figura 10 Comportamiento del algoritmo mmsh con un pbest=0.2.....	66
Figura 11 Comportamiento del algoritmo mmsh con un pbest=0.5.....	66
Figura 12 Comportamiento del algoritmo mmsh con un pbest=0.8.....	67
Figura 13 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un d=0.05 y pbest=0.5.....	71
Figura 14 Presiones del algoritmo mmsh en cada nodo con un d=0.05 y pbest=0.5	71
Figura 15 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un d=0.005 y pbest=0.5.....	72
Figura 16 Presiones del algoritmo mmsh en cada nodo con un d=0.005 y pbest=0.5.....	72
Figura 17 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un d=0.0005 y pbest=0.5	73
Figura 18 Presiones del algoritmo mmsh en cada nodo con un d=0.0005 y pbest=0.5.....	73

Figura 19 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un $d=0.0000005$ y $pbest=0.5$	74
Figura 20 Presiones del algoritmo mmsh en cada nodo con un $d=0.0000005$ y $pbest=0.5$	74
Figura 21 Comportamiento de la feromona maxima en el algoritmo mmhs.	75
Figura 22 Comportamiento de la feromona mínima en el algoritmo mmhs.	75
Figura 23 Número de nodos que cumplen la restricción en la mejor solución vs iteraciones.....	76
Figura 24 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un $d=0.05$ $pbest=0.5$ y una frecuencia de la feromona de la mejor solución..	78
Figura 25 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un $d=0.005$ $pbest=0.5$ y una frecuencia de la feromona de la mejor solución global=5.....	79
Figura 26 Comportamiento del algoritmo mmsh (costo vs número de iteraciones) con un $d=0.0005$ $pbest=0.5$ y una frecuencia de la feromona de la mejor solución global=5.....	79
Figura 27 Comportamiento del algoritmo mmsh (mejor solución global vs número de iteraciones) con un $d=0.005$ $pbest=0.5$	80

LISTA DE ANEXOS

ANEXO A CODIGO OCH-EPANET PARA EL PROBLEMA DEL AGENTE VIAJERO	93
ANEXO B CÓDIGO OCH-EPANET PARA LAS RDA'S.....	96

RESUMEN

TITULO: EVALUACIÓN DEL MÉTODO DE COLONIA DE HORMIGAS PARA DISEÑOS DE SISTEMAS DE DISTRIBUCIÓN DE AGUA*

AUTOR: JESSIKA TATIANA ORTIZ APONTE**

PALABRAS CLAVES: ACO, HORMIGA ARTIFICIAL, FEROMONA, REDES DE DISTRIBUCIÓN DE AGUA.

DESCRIPCIÓN: Los métodos evolutivos como la Optimización de la Colonia de Hormigas (OCH) o Ant Colony (AC en inglés) son métodos basados en el comportamiento de alimentación de las hormigas. Estos métodos se han empleado en el estudio y diseño óptimo de Redes de Distribución de Agua (RDA's) durante los últimos años.

Este trabajo tiene como objetivo evaluar el método OCH para el diseño de RDA's. Para realizar esta evaluación, el OCH programado en MATLAB, fue sincronizado con el programa de código abierto EPANET 2.0, que en este estudio funcionó como motor de cálculo hidráulico. Adicionalmente, se presenta una guía en español del funcionamiento del algoritmo OCH-EPANET, para futuras aplicaciones de OCH en ingeniería civil.

Se presentan dos ejemplos conocidos en la literatura con el fin de evaluar el desempeño del OCH y OCH-EPANET. El primer ejemplo, es conocido como el problema de agente viajero, que es del tipo NP-hard y su solución con el OCH muestra las ventajas de la optimización evolutiva. El segundo ejemplo, es el problema de la tuberías de Hanoi [1], el cual es solucionado en este trabajo mediante el algoritmo de optimización OCH-EPANET.

Los resultados obtenidos muestran la aplicabilidad del esquema planteado para dos variantes del OCH, conocidos como sistema de Hormigas (SH) y algoritmo de Máximos y Mínimos Sistema de Hormigas (MMSH). Finalmente, se presentan las constantes y los parámetros heurísticos necesarios para correr con éxito el OCH.

*Proyecto de grado: Modalidad investigación

**Facultad de ingenierías físico-mecánicas, Escuela de ingeniería civil. Director: Oscar Javier Begambre. Co-Director: Mario Garcia Solano

ABSTRACT

TITLE: EVALUACIÓN DEL MÉTODO DE COLONIA DE HORMIGAS PARA DISEÑOS DE SISTEMAS DE DISTRIBUCIÓN DE AGUA*

AUTHOR: JESSIKA TATIANA ORTÍZ APONTE**

KEYWORDS: Plywood, steel post-tensioning, laminations, wood bridge, wooden board post-tensioned.

ABSTRACT: The evolutionary methods as Ant Colony Optimization (ACO) are based on the feeding behavior of the ants. These methods have been used in the study and optimal design of Water Distribution Systems (WDS) during the last years.

This study aims to evaluate the ACO method to WDS designing. In order to do this evaluation, the ACO was programmed in MATLAB and synchronized with the open source program EPANET 2.0, which in this study worked as hydraulic calculation engine. Additionally, an operation guide of the algorithm OCH-EPANET is presented in Spanish in order to collaborate to future applications in civil engineering.

In order to evaluate the performance of OCH and OCH-EPANET there are two examples known in the literature. The first example is known as the traveling salesman problem, which is NP-hard and OCH sample solution with the advantages of evolutionary optimization. The second example is the problem of Hanoi pipes [1], which is solved in this work by the optimization algorithm OCH-EPANET. The results show the applicability of the proposed scheme for two ACO variants, known as Ant System (SH) and Minimum and Maximum algorithm Ant System (MMSH). Finally, the heuristic constants and parameters necessary to run a successful ACO are presented.

*Degree Project. Investigation Modality

**Faculty of Physical – Mechanical Engineering. Civil Engineering School. Director: Ing PhD. Oscar Javier Begambre. Codirector: Ing. Mario García Solano

INTRODUCCIÓN

Este trabajo tiene como finalidad realizar el diseño de RDA'S usando OCH (ACO por sus siglas en inglés) evaluando sus ventajas y desventajas. Adicionalmente, se hace una descripción detallada de los procedimientos que caracterizan a los algoritmos evolutivos en los problemas de optimización combinatoria buscando llenar el vacío de la literatura en español sobre el tema en específico [3].

En primera instancia se explica el proceso de la OCH para buscar las soluciones más factibles dentro de un espacio de búsqueda, los algoritmos derivados de OCH utilizados son el Sistema de Hormigas (SH) y Máximos y Mínimos Sistema de Hormigas (MMSH) su naturaleza iterativa de la generación de soluciones encontradas a través de la búsqueda se incrementa con las soluciones aprendidas anteriormente se puede caracterizar por dos rasgos principales [1]. (i) La exploración que es la capacidad del algoritmo de buscar una solución general en el espacio y (ii) la explotación que es la capacidad del algoritmo para buscar más a fondo donde las buenas soluciones han sido encontradas previamente.

Para dar una mayor comprensión de los pasos a seguir se muestra un ejemplo conocido en la literatura como el problema del agente viajero, mostrando pasos puntuales para encontrar las soluciones más favorables [3]. Es importante recalcar que las programaciones están hechas en el software MATLAB ya que esta es una herramienta con un lenguaje de programación propio (lenguaje M) fácil de manejar y conocido por los estudiantes de ingeniería en general.

Debido a los altos costos asociados con la construcción de las RDA's muchas investigaciones en los últimos años se han dedicado al desarrollo de técnicas para reducir al mínimo los costos de dicha infraestructura. Este proceso se ha dado el título de "optimización" o "diseño óptimo" de las RDA's, se ha demostrado que los

problemas de RDA's es de complejidad NP-DURO [9] lo cual quiere decir que es un problema intratable para el cuál no se conoce ningún método determinístico para solucionarlo en un tiempo polinomial. El problema de diseño de RDA's es bastante complicado debido a la relación no lineal entre flujo, las pérdidas de cabeza (carga), y a la presencia de variables discretas como los diámetros de tubería comerciales. Adicional mente la función de costos de tubería también tienen una relación no lineal con los diámetros [10]. La variables de decisión pueden tomarse por diferentes criterios de diseño, sin embargo la literatura sobre la optimización de las RDA's, ha tratado tradicionalmente con un problema simplista e idealizado. Las variables de decisión han sido principalmente asociados con las tuberías del sistema donde más concretamente las opciones de decisión han sido la selección de (i) un diámetro de tubería nueva, (ii) un diámetro de tubería duplicada y (iii) la limpieza de una tubería existente para reducir la resistencia hidráulica.

La metodología utilizada para la implementación del algoritmo OCH es la utilización de un motor hidráulico que le permita calcular los caudales en las tuberías y las presiones en los nodos una vez se haya fijado los diámetros de la red. El motor más eficiente es el código fuente de Epanet 2.0 que es un software libre.

1. PRELIMINARES

1.1 METODOS DE OPTIMIZACIÓN MATEMATICA

La optimización consiste en la selección de una alternativa mejor en algún sentido que las demás alternativas posibles. Los problemas de optimización se componen en general de estos tres ingredientes.

Función objetivo: Es la medida cuantitativa del funcionamiento del sistema que desea optimizar (maximizar o minimizar).

Variables: Representan las decisiones que se pueden tomar para afectar el valor de la función objetivo. Desde un punto de vista funcional se pueden clasificar en variables independientes o principales o de control y variables dependientes o auxiliares o de estado, aunque matemáticamente todas son iguales.

Restricciones: Representan el conjunto de relaciones (expresadas mediante ecuaciones) que ciertas variables están obligadas a satisfacer.

Resolver un problema de optimización consiste en encontrar el valor que deben tomar las variables para hacer óptima la función objetivo satisfaciendo el conjunto de restricciones. Los métodos de optimización los podemos clasificar en métodos clásicos (que son algoritmos que se explican en los libros de optimización) y métodos metaheurísticos (que aparecieron ligados a lo que se denominó inteligencia artificial e imitan fenómenos observados en la naturaleza).

Se conocen como optimización combinatoria compleja los problemas que no pueden resolverse en un algoritmo conocido con tiempo polinomial, se definen NP-Duro [3]. Un ejemplo clásico de un problema NP-duro que se conoce como el problema "vendedor ambulante". Esto requiere el cálculo de la ruta óptima para un vendedor de viajar a un número de destinos diferentes y volver a su punto de partida. Para resolver completamente este problema, incluso para un pequeño número de destinos, requiere más potencia de cálculo que está disponible para cualquier equipo moderno.

2. JUSTIFICACIÓN

Debido a la naturaleza combinatoria del problema de diseño óptimo de RDA's se hace necesario explorar alternativas de diseño basadas en algoritmos heurísticos. Lo anterior sumando a la necesidad de reducir costos de diseño y construcción de RDA's ha hecho que Algoritmos Genéticos (Simpsons et. al 1994) y OCH (Maier et. al, 2003) hayan tenido una gran aceptación en los últimos años.

2.1 FORMULACIÓN DEL PROBLEMA

¿Es posible sincronizar el algoritmo OCH con el motor hidráulico EPANET para obtener diseños óptimos de RDA's?

Si lo anterior es posible ¿cuál es la variante del OCH y cuáles son los parámetros heurísticos de la misma que producen soluciones óptimas o casi óptimas?

2.2 OBJETIVOS

2.2.1 Objetivo general

Evaluar el diseño de Sistemas de distribución de Agua por el método de Optimización de la Colonia de Hormigas (OCH) o Ant Colony Optimization (ACO)

al compararlo con un método tradicional de diseño, para buscar las ventajas o las desventajas en la obtención de los resultados.

2.2.2 Objetivos específicos

Encontrar el algoritmo derivado del OCH, que sea adecuado para la optimización de las Redes de Distribución de agua (RDA's).

Establecer una red hidráulica como ejemplo para obtener el conjunto de diámetros estableciendo el costo de la red para que sea mínimo (costos de construcción) respetando las restricciones hidráulicas.

Analizar la eficiencia de los métodos ACO en el diseño de sistemas combinatorios para justificar su uso en los diseños de las RDA's.

Verificar si la herramienta OCH puede proporcionar un mayor rendimiento comparado con el diseño tradicional.

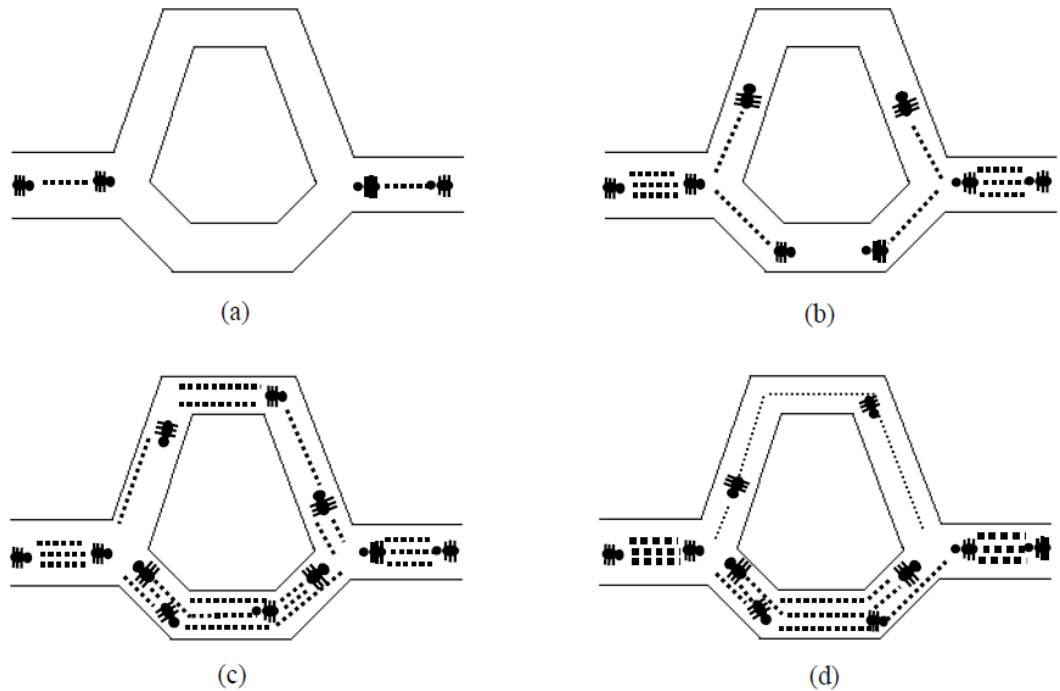
3. MARCO CONCEPTUAL

3.1 LAS COLONIAS DE HORMIGAS

En el comportamiento de la hormigas en la búsqueda de alimentos se distingue trazando el camino más corto entre el hormiguero y los alimentos, el eje de esa búsqueda es el depósito de feromona como rastro que orienta el recorrido, las hormigas prefieren seguir la mayor concentración de feromona la misma que se consigue por el recorrido más corto hecho por las hormigas (que por esta razón pueden repetirlo en forma más seguida). En principio siguen rutas aleatorias, pero las que han utilizado las de menor longitud pueden regresar más rápidamente (pues transitan a una velocidad uniforme) y repetir el camino dejando una mayor densidad de huellas. Las depositadas en las otras rutas se van evaporando y dejan de tener interés en los nuevos recorridos.

En la figura 1(a) las hormigas llegan a un punto donde tienen que decidir cuál camino deben elegir lo resuelven de una manera aleatoria en consecuencia la mitad de las hormigas toman un camino y la otra mitad el otro como se muestra en la figura1 (b). Al moverse a una velocidad constante las hormigas que escogieron el camino más corto tendrá un mayor número de feromonas depositadas como se ilustra en la figura 1(c). Por último en la figura 1(d) el camino con mayor depósito de feromona se convierte en la elección más deseable por las hormigas haciendo que las feromonas depositadas en el otro camino desaparezcan y todas decidan tomar esa ruta [4].

Figura 1 Comportamiento de las hormigas reales.



Fuente: Metaheurística de Optimización mediante Colonias de Hormigas y Aplicaciones

3.2 METAHEURÍSTICA DE OPTIMIZACIÓN MEDIANTE COLONIA DE HORMIGAS

La metaheurística de Optimización mediante Colonias de hormigas (OCH) o Ant Colony Optimization (ACO) [5] propuesta para resolver problemas complejos de optimización combinatoria tiene su fuente de inspiración en el comportamiento de Colonias de Hormigas reales. En la analogía con el tiempo biológico, OCH se basa en la comunicación indirecta de una colonia de agentes simples llamados hormigas (artificiales) por medio de huella de feromona (artificial) . La huella de feromona en OCH sirve como información numérica distribuida que las hormigas

usan como distribución probabilística de soluciones del problema a resolver y la adaptan mediante la ejecución del algoritmo para reflejar su experiencia de búsqueda [5]. La estructura genérico de la metaheurística OCH es la siguiente:

1. procedimiento metaheurística_OCH()
2. inicialización_ de_ parámetros
3. mientras (criterio _ de _ terminación _ no _satisfecho
4. programación _ de_ actividades
5. hormigas _ y_ actividad()
6. evaporación _de_ feromona ()
7. acciones _ de optimización opcionales() {opcional}
8. fin de programación _de_ actividades
9. fin mientras
10. fin procedimiento

1. procedimiento hormigas _y_ actividad()
2. repetir en paralelo desde k=1 hasta número _ hormigas
3. nueva_ hormiga(k)
4. fin repetir en paralelo
5. fin procedimiento

1. procedimiento nueva hormiga(id_ hormiga)
2. inializa _ hormiga (id_ hormiga)
3. L=actualiza_ memoria_ hormiga()
4. mientras (estado_ actual \neq estado_ objetivo)
5. P= calcular_ probabilidades_ de_ transición (A,L,W)
6. siguiente_ estado= aplicar_ política_ decisión (P,W)
7. mover_ al_ siguiente_ estado (siguiente _estado)
si (actualización _feromona _en _línea _paso _a _paso)
8. depositar_ feromona_ en_ el_ arco_ visitado()

```

    fin si
9. L=actualizar_estado_interno()
10. fin mientras
    si (actualización_feromona_en_línea_a_posteriori)
11. fin mientras
12. depositar_feromona_en_el_arco_visitado()
13. fin para
    fin si
14. liberar_recursos_hormigas (id_Hormiga)
15. fin procedimiento

```

La inicialización de parámetros depende del algoritmo específico, generalmente deben tenerse en cuenta parámetros como: el rastro inicial de feromona asociado a cada transición o arco, el número de hormigas en la colonia, los pesos que definen la proporción en la que afectarán la información heurística y memorística en la regla de transición probabilística. En programación de actividades se controla la planificación de tres componentes: la generación y puesta en funcionamiento de las hormigas artificiales; la evaporación de feromona, que se usa como un mecanismo para evitar el estancamiento en la búsqueda y permitir que las hormigas busquen y exploren nuevas regiones del espacio; y las acciones del dominio, utilizadas para implementar tareas desde una perspectiva global que no pueden llevar a cabo las hormigas, por ejemplo, observar la calidad de todas las soluciones generadas y depositar una nueva cantidad de feromona adicional en las transiciones asociadas a algunas soluciones. El procedimiento `actualiza_memoria_hormiga()` se encarga de especificar el estado inicial desde el que la hormiga comienza su camino y además almacenar la componente correspondiente en la memoria de la hormiga L. La decisión sobre cuál será el nodo inicial depende del algoritmo específico. En los procedimientos `calcular_probabilidades_de_transición` y `aplicar_política_decisión` se tienen en consideración el estado actual de la hormiga y el conjunto de arcos del grafo (A),

los los valores actuales de la feromona visibles en dicho nodo y las restricciones del problema para establecer el proceso de transición probabilístico hacia otros estados válidos. La actualización_feromona_en_línea_paso_a_paso es el procedimiento donde se actualiza el rastro de feromona asociado a un arco, cuando la hormiga se mueve entre los nodos que este conecta. Una vez que la hormiga ha construido la solución puede reconstruir el camino recorrido y actualizar los rastros de feromona de los arcos visitados mediante el procedimiento llamado actualización_feromona_en_línea_a_posteriori [4,5].

3.2.1 CARACTERISTICAS DE LAS HORMIGAS ARTIFICIALES

Como se indica en [3], una hormiga artificial es un agente computacionalmente simple que intenta construir una solución a un problema explorando para ello los rastros de feromona disponibles y la información heurística, sin embargo en algunos problemas puede también construir soluciones no validas que podrán ser penalizadas (por ejemplo la optimización de las RDA's). Tiene las siguientes propiedades:

- Busca soluciones validas de costo mínimo para el problema a resolver.
- Tiene una memoria LT en la que almacena el camino recorrido para poder construir soluciones validas , evaluar las soluciones generadas o bien reconstruir el camino completo llevado por la hormiga.
- Comienza en el estado inicial y se mueve siguiendo estados válidos, construyendo la solución asociada incrementalmente.
- Lleva a cabo los movimientos aplicando reglas de transición en función de los rastros de feromona que están disponibles, de los valores heurísticos de la memoria privada de la hormiga y de las restricciones del problema.

- Durante su recorrido modifica el entorno depositando una cantidad de feromona.
- Una vez que una hormiga ha construido una solución válida puede recorrer el camino de vuelta actualizando los espacios visitados.

3.3 ALGORITMOS UTILIZADOS DE LA METAHEURISTICA OCH

3.3.1 Algoritmo sistema de hormigas

En el algoritmo Sistema de Hormigas (SH) se construyen las soluciones de la siguiente forma: Para cada hormiga k en cada paso de construcción se escoge ir del nodo i al siguiente nodo j , $\forall j$, con una probabilidad $P_{k(ij)}$

$$P_{k(i,j)} = \frac{[\tau(i,j)]^\alpha \cdot [\eta(i,j)^\beta]}{\sum_{j \in LT_k} [\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta} \quad \text{si } j \in LT_k$$

Ecuación (1)

donde: LT_k es el vecindario alcanzable por la hormiga k cuando se encuentra en el nodo i ; α es el factor de escalado de feromona y β el de visibilidad ambos se usan para afinar el proceso de búsqueda; $\tau(i,j)$ el valor de feromona en el arco a_{ij} que une los nodos i y j ; $\eta(i,j)$ se denomina función de visibilidad que depende totalmente de las características del problema que se va a resolver, por ejemplo para el problema del agente viajero es $1/d_{ij}$, donde d_{ij} es la distancia entre las ciudades i y j . Luego se comparan para todas las hormigas sus soluciones encontradas con la mejor hasta el momento. En la actualización de la huella de

feromona se evapora una proporción constante de feromona en cada arco y luego para cada hormiga una vez que la solución está completa deposita una cantidad de feromona en dependencia de la calidad de su solución, o sea actualización en línea a posteriori [4, 5]. Como se ha dicho la deposición de feromona se realiza una vez que todas las hormigas han acabado de construir sus soluciones. Primero, los rastros de feromona asociados a cada arco se evaporan reduciendo todos los rastros de feromona en un factor constante:

$$\tau_{i,j}(t) \leftarrow (1 - \rho) \cdot \tau_{i,j}(t)$$

Ecuación (2)

Donde $\rho \in (0,1]$ es la tasa de evaporación. El siguiente paso de cada hormiga es recorrer de nuevo el camino que ha seguido (el camino está almacenado en su memoria local (lista tabú) y deposita una cantidad de feromona $\Delta\tau_{i,j}$ en cada conexión por la que ha viajado:

$$\tau_{i,j}(t + 1) = \tau_{i,j}(t) + \Delta\tau_{i,j}(t), \quad \forall a_{ij} \in S_k$$

Ecuación (3)

Donde $\Delta\tau_{i,j} = f(C(S_k))$, es decir, la cantidad de feromona que se deposita depende de la calidad $C(S_k)$ de la solución (S_k) construida por la hormiga K.

Para un mayor entendimiento:

$$\Delta\tau_{i,j}(t) = \sum_{k=1}^m \Delta\tau_{i,j}^k(t)$$

Ecuación (4)

donde m es número de hormigas. La incorporación de feromonas individual aportado por cada hormiga está dada [5]:

$$\Delta\tau_{i,j}^k = \frac{Q}{f(Sk)}$$

Ecuación (5)

Donde Q es un factor de adición de la feromona (constante), Sk es el conjunto de opciones seleccionados por la hormiga K.

Como se muestra en [3] la descripción del algoritmo sistema de hormigas, se presenta a continuación de una manera resumida:

Nueva_ Hormiga para este algoritmo OCH en particular:

1. Procedimiento Nueva_ Hormiga (id_ hormiga)
2. K=id_ Hormiga; r= generar_ estado_ inicial; Sk=i
3. Lk=i
4. Mientras (estado_ actual)
5. Para cada s \in Nk hacer $P_{k(ij)} = \frac{[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}{\sum_{j \in LT_k} [\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}$
6. Siguiete_ est= aplicar_ política _de_ decisión (P, Nk(i))
7. l= siguiete_ est; Sk = (Sk, i)
8. -----

9. $L_k = L_k \cup i$
10. Fin mientras
 - {se ejecuta el procedimiento
 - Evaporación de feromona () se lanza y evapora la
 - feromona en cada arco $a_{ij} : \tau_{i,j} \leftarrow (1 - \rho) \cdot \tau_{i,j}$
11. Para cada arco $a_{ij} \in S_k$ hacer
12. $\tau_{i,j}(t + 1) = \tau_{i,j}(t) + f(c(S_k))$
13. *fin para*
14. Liberar _ recursos _ hormiga (id_ Hormiga)
15. Fin Procedimiento

La línea 8 vacía se incluye para hacer notar que no existe una actualización de feromona en línea paso a paso y que antes de la línea 12. El dominio debe haber aplicado la evaporación de feromona. De hecho este es un ejemplo donde la construcción Programación_ de _ actividades interfiere con el funcionamiento de los procedimientos fundamentales de la metaheurística OCH.

3.3.2 Ejemplo de aplicación

El problema del agente viajero (TSP) consiste en encontrar el camino más corto posible entre un número variable de ciudades o sitios que se deben visitar y por los cuales solamente se debe pasar una sola vez, con la condición que al final del recorrido se debe retornar al punto de origen. Se ha demostrado que la complejidad del problema es $(n-1)!$ por ejemplo si se tiene 6 ciudades se tienen 120 posibles soluciones. A continuación, el funcionamiento del algoritmo se mostrara paso a paso en un sencillo ejemplo.

Tomado de [3], un agente viajero tiene que visitar 6 ciudades diferentes para tener la certeza de la solución óptima del problema se ubican las ciudades en una circunferencia de 100 km, centros de coordenada (0,0), se dan las coordenadas de cada ciudad para calcular sus distancias. Las matrices mostradas en el ejemplo son producto de la programación hecha en el software MATLAB, esto se hizo con el fin de corroborar el funcionamiento básico del algoritmo con un ejercicio tomado de la literatura estudiada.

Figura 2 Ubicaciones en el plano de las ciudades A, B, C, D, E, F ejemplo TSP

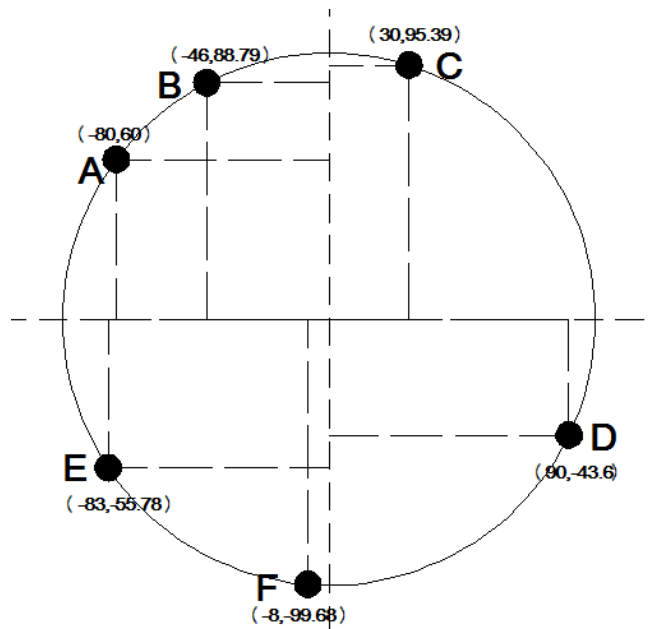


Tabla 1 Lista de ciudades con sus coordenadas ejemplo TSP

RADIO	CENTRO	0,0
100 Km	COORDENADAS	
CIUDAD	X	Y
A	-80	60
B	-46	88,79
C	30	95,39
D	90	-43,6
E	-83	-55,78
F	-8	-99,68
LONGITUD OPTIMA		587,859

Fuente: Tesis Implementación en hidroinformática de un método de optimización matemática basada en la colonia de hormigas [3].

Tabla 2 Datos de entrada ejemplo TSP

CONSTANTES							Valores de entrada	
m = No. Hormigas	ciudades (NC)	α	β	ρ	Q	NC max	τ inicial	iteraciones
6	6	1	2	0,5	10	6	0,03	100

Fuente: Tesis Implementación en hidroinformática de un método de optimización matemática basada en la colonia de hormigas [3].

Tabla 3 Matriz de distancias dij ejemplo TSP

	A	B	C	D	E	F
A	0	44,5518137	115,552811	199,080285	175,161932	115,81886
B	44,5518137	0	76,2860407	189,797556	192,262687	149,229638
C	115,552811	76,2860407	0	151,387648	198,736773	188,736242
D	199,080285	189,797556	151,387648	0	112,911321	173,428234
E	175,161932	192,262687	198,736773	112,911321	0	86,9034522
F	115,81886	149,229638	188,736242	173,428234	86,9034522	0

Fuente: Programación en MATLAB

Las distancias entre ciudades se hallaron trigonométricamente de la siguiente manera $dij = \sqrt{(xj - xi)^2 + (yj - yi)^2}$

Tabla 4 Matriz de visibilidad nij ejemplo TSP

	A	B	C	D	E	F
A	0	0,02244578	0,00865405	0,0050231	0,005709	0,00863417
B	0,02244578	0	0,01310856	0,00526877	0,00520122	0,00670108
C	0,00865405	0,01310856	0	0,00660556	0,00503178	0,0052984
D	0,0050231	0,00526877	0,00660556	0	0,00885651	0,00576607
E	0,005709	0,00520122	0,00503178	0,00885651	0	0,01150702
F	0,00863417	0,00670108	0,0052984	0,00576607	0,01150702	0

Fuente: Programación en MATLAB

La visibilidad nij se calcula como el inverso de la distancia $nij = 1/dij$

Es necesario crear una matriz con las mismas dimensiones de dij y nij que contenga la feromona inicial para todas las posibles opciones con un mismo valor

que será en este caso 0,03 al correr el programa las feromonas adheridas tomaran valores diferentes y solo se agregaran a las rutas con mayor probabilidad. En la siguiente tabla se muestra la elección de la hormiga A1 en tiempo cero escogiendo la que tiene una mayor probabilidad en este caso será la ciudad B (se muestra con mas detalles en [3]).

Tabla 5 Aplicación de la función de probabilidad para la hormiga H1 en t=0 ejemplo TSP

1	2	3	4	5	6	7	8	9	10	11	12	13
Tiempo	Hormiga	nodo i	j	tramo	τ_{ij}	n_{ij}	τ_{ij}^α	n_{ij}^β	$\tau_{ij}^\alpha * n_{ij}^\beta$	$\sum \tau_{ij}^\alpha * n_{ij}^\beta$	10/ 11	nj destino
0	A1	A	B	A-B	0,03	0,0224	0,03	0,000502	1,505E-05	2,1267E-05	0,70780	B
			C	A-C	0,03	0,0087	0,03	0,000076	2,271E-06		0,10677	
			D	A-D	0,03	0,005	0,03	0,000025	7,5E-07		0,03527	
			E	A-E	0,03	0,0057	0,03	0,000032	9,747E-07		0,04583	
			F	A-F	0,03	0,0086	0,03	0,000074	2,219E-06		0,10433	

Fuente: Programación en MATLAB

Tabla 6 Matriz de la función de probabilidad para las 6 Hormigas en t=0 ejemplo TSP

H1 [A]	0,70851917	0,10532255	0,03548345	0,0458356	0,10483923
H2 [B]	0,64977569	0,22161752	0,03580247	0,03489026	0,05791407
H3 [C]	0,21786804	0,49987843	0,12693274	0,0736543	0,0816665
H4 [D]	0,12112475	0,13326257	0,20946358	0,37654288	0,15960622
H5 [E]	0,11017995	0,09145174	0,08559049	0,2651594	0,44761842
H6 [F]	0,23803432	0,1433798	0,08963705	0,10615942	0,4227894

Fuente: Programación en MATLAB

Cada una de las columnas representa la siguiente ciudad en la lista en orden alfabético (por ejemplo H2 sale de la ciudad B por lo tanto 0,6497.. corresponde a la probabilidad de ir a la ciudad A y 0,2216 corresponde a la probabilidad de ir

ciudad C). Las iteraciones terminan cuando todas las hormigas han recorrido las ciudades sin repetir, la hormiga artificial genera una lista tabú para indicar que ciudades ha visitado y cuál distancia representa cada opción como se muestra en la siguiente tabla.

Tabla 7 Lista tabú en la primera iteración para cada hormiga k ejemplo TSP

H1	C	B	A	F	E	D
H2	E	F	A	B	C	D
H3	D	E	F	A	B	C
H4	C	B	A	F	E	D
H5	B	A	C	D	E	F
H6	D	E	F	A	B	C

Fuente: Programación en MATLAB

Tabla 8 Acumulado de la lista tabú en la primera iteración para cada hormiga k ejemplo TSP

							Lon total
H1	76,2860407	44,5518137	115,81886	86,9034522	112,911321	151,387648	587,859136
H2	86,9034522	115,81886	44,5518137	76,2860407	151,387648	112,911321	587,859136
H3	112,911321	86,9034522	115,81886	44,5518137	76,2860407	151,387648	587,859136
H4	76,2860407	44,5518137	115,81886	86,9034522	112,911321	151,387648	587,859136
H5	44,5518137	115,552811	151,387648	112,911321	86,9034522	149,229638	660,536684
H6	112,911321	86,9034522	115,81886	44,5518137	76,2860407	151,387648	587,859136

Fuente: Programación en MATLAB

Al terminar cada iteración se obtiene una longitud total del recorrido de cada hormiga, en la tabla anterior 7 de las 8 hormigas lograron un viaje optimo en el

primer ciclo. Cuando todas las hormigas han completado el viaje, se actualiza el rastro de feromona adicionado a cada tramo usando la sumatoria de feromona que cada hormiga aporta en los tramos de su recorrido y que es inversamente proporcional a la longitud del recorrido total. La actualización de la feromona se obtiene aplicando la siguiente fórmula:

$$\Delta\tau_{i,j} = \frac{Q}{LT_k}$$

Ecuación (6)

Donde Q se toma de la tabla de constantes (es un valor aleatorio) y Lk se toma de la lista tabú y acumulado de distancia al terminar el tour (ver tabla 8).

Tabla 9 Delta de feromonas ejemplo TSP

H1	0,06430017	0	0	0	0,11907615
H2	0,13608702	0,03402176	0	0	0,0151392
H3	0,03027841	0,13608702	0,01701088	0	0
H4	0	0	0,14935455	0,05103263	0
H5	0	0	0	0,14935455	0,03402176
H6	0,01701088	0,0151392	0	0	0,16636543

Fuente: Programación en MATLAB

Para calcular la feromona que queda en el tramo se evapora la feromona inicial $\tau_{i,j}$ y se adiciona la feromona que se calcula para cada tramo; esto es:

$$\tau_{i,j}(t+n) = \rho \cdot \tau_{i,j}(t) + \Delta\tau_{i,j}$$

Ecuación (7)

Tabla 10 Actualización de feromonas ejemplo TSP

H1	0,07930017	0,015	0,015	0,015	0,13407615
H2	0,15108702	0,04902176	0,015	0,015	0,0301392
H3	0,04527841	0,15108702	0,03201088	0,015	0,015
H4	0,015	0,015	0,16435455	0,06603263	0,015
H5	0,015	0,015	0,015	0,16435455	0,04902176
H6	0,03201088	0,0301392	0,015	0,015	0,18136543

Fuente: Programación en MATLAB

Al adherirse la feromona en los tramos se aumenta la probabilidad de elección para ese recorrido cuando se han terminado los ciclos el algoritmo muestra los recorridos óptimos encontrados con las hormigas artificiales de la siguiente manera.

Tabla 11 Lista tabú final para cada hormiga k ejemplo TSP

H1	B	A	F	E	D	C
H2	C	B	A	F	E	D
H3	C	B	A	F	E	D
H4	D	E	F	A	B	C
H5	A	B	C	D	E	F
H6	E	F	A	B	C	D

Fuente: Programación en MATLAB

Tabla 12 Acumulado final para cada hormiga k ejemplo TSP

							Long total
H1	44,5518137	115,81886	86,9034522	112,911321	151,387648	76,2860407	587,859136
H2	76,2860407	44,5518137	115,81886	86,9034522	112,911321	151,387648	587,859136
H3	76,2860407	44,5518137	115,81886	86,9034522	112,911321	151,387648	587,859136
H4	112,911321	86,9034522	115,81886	44,5518137	76,2860407	151,387648	587,859136
H5	44,5518137	76,2860407	151,387648	112,911321	86,9034522	115,81886	587,859136
H6	86,9034522	115,81886	44,5518137	76,2860407	151,387648	112,911321	587,859136

Fuente: Programación en MATLAB

Cualquiera de los caminos escogidos por las hormigas cumplen con la longitud optima del problema en los anexos del trabajo se presentan los pasos detallados de la programación con la cual se obtuvieron las matrices presentadas anteriormente.

3.3.1 Algoritmo máximos y mínimos sistema de hormigas

En el algoritmos MMSH la probabilidad de ir a un nodo a otro es igual que en el algoritmo SH. La base de MMSH es proporcionar una evolución dinámica vinculada en la intensidad de los rastros de feromonas las cuales deben estar dentro un límite determinado de la ruta con mayor intensidad de feromona esto fue desarrollado en [7]. MMSH utiliza límites superiores e inferiores para asegurar la intensidad de la feromona dentro de un rango de la siguiente manera.

$$\tau_{min}(t) \leq \tau_{i,j}(t) \leq \tau_{max}(t) \text{ El limite superior } \tau_{max}(t) \text{ es obtenido por [7]}$$

$$\tau_{max}(t) = \frac{1}{1 - \rho} \frac{Q}{f(S^{gb}(t-1))}$$

Ecuación (8)

Donde S^{gb} Es el mejor camino global encontrado en la iteración t y el límite inferior $\tau_{min}(t)$ es obtenido por [16].

$$\tau_{min}(t) = \frac{\tau_{max}(t)(1 - \sqrt[n]{P_{best}})}{(NO_{avg} - 1)\sqrt[n]{P_{best}}}$$

Ecuación (9)

Donde P_{best} es la probabilidad que S^{gb} sea seleccionado por una hormiga en la iteración t , n es el número de puntos de decisión y NO_{avg} es el número promedio de opciones en cada punto de decisión. Dentro de las rutas de feromonas MMAS se inicializan un valor arbitrariamente alto de manera que en la segunda iteración los caminos se ponen a $\tau_{max}(t)$ las justificaciones teóricas de los límites se dan en [7]. Para explotar la mejor solución durante una iteración o durante una corrida del algoritmo después de cada iteración solamente una simple hormiga adiciona feromona esta puede ser una de las que encuentre la mejor solución de una iteración o la mejor solución global. Un análisis de $\tau_{min}(t)$ mostro que P_{best} define el ancho de los límites entre las feromonas si se ponen valores muy bajos su búsqueda se estanca produciendo que las probabilidades tiendan a permanecer igual. Los límites de las feromonas sirven para fomentar la exploración y proveer un énfasis sobre la explotación de las mejores soluciones, MMSH actualiza la mejor hormiga de cada iteración y periódicamente la mejor solución global asegurando que la buena información se retenga y refuerza de la siguiente manera.

$$\Delta t_{i,j}(t) = \Delta t_{i,j}^{ib}(t) + \Delta t_{i,j}^{gb}(t)$$

Ecuación (10)

La adición de la feromona es determinado en las siguientes ecuaciones.

$$\Delta \tau_{i,j}^{ib}(t) = \frac{Q}{f(S^{ib}(t))} \quad \text{si } (i,j) \in S^{ib}(t)$$

Ecuación (11)

Si S^{ib} es la iteración del mejor camino que se encuentra en la iteración t . La adición de la feromona de la mejor hormiga global $\Delta t_{i,j}^{gb}(t)$ es obtenida por.

$$\Delta \tau_{i,j}^{gb}(t) = \frac{Q}{f(S^{gb}(t))} \quad \text{si } (i,j) \in S^{gb}(t) \text{ y } t = pf_{global} \text{ donde } p \in N$$

Ecuación (12)

Donde f_{global} es la frecuencia de la feromona en la mejor actualización global y N es el conjunto de números naturales MMAS también utiliza otro mecanismo conocido como rastro de feromona suavizado (RFS) Esto reduce la diferencia relativa entre de la feromona Las intensidades, y la mejor fomentación de la exploración. La RFS es obtenido en [7].

$$\tau_{i,j}^*(t) = \tau_{i,j}(t) + d(\tau_{max}(t) - \tau_{i,j}(t))$$

Ecuación (13)

Donde $0 \leq \alpha \leq 1$ RFS es el coeficiente y $t_{i,j}^*(t)$ es la intensidad de la de feromonas después de la suavización si $\alpha=0$ La RFS no tiene ningún efecto, mientras que si $\alpha=1$ todos los niveles de feromonas escalan hasta $\tau_{max}(t)$.

Es importante recalcar que se han propuesto varios modelos de la metaheurística OCH, entre ellos se encuentran: Sistema de Hormigas (SH) o Ant System (AS) [14], Sistema Colonia de Hormigas (SCH) o Ant Colony System (ACS) [11], Max-Min Sistema de Hormigas (MMSH) o Max-Min Ant System (MMAS) [19], Sistema de Hormigas con Ordenación(SHO) o Rank-Based Ant System [20], Sistema Mejor-Peor Hormiga (SMPH) o Best-Worst Ant System [21, 22] y ACO en Dos Etapas o Two-Step Ant Colony Optimization (TS-ACO) [4].

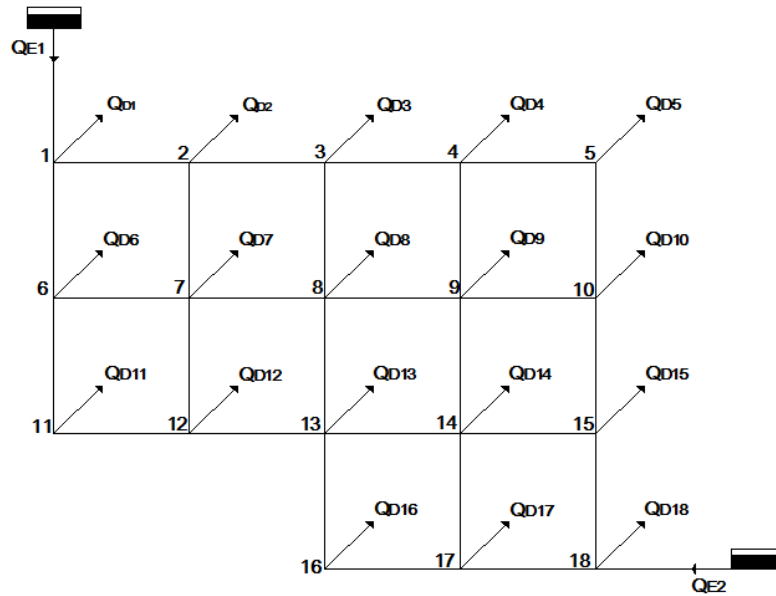
3.4 ANALISIS DE REDES DE TUBERÍAS: REDES CERRADAS

3.4.1 Principios fundamentales de análisis de redes cerradas

Las redes cerradas son conductos ramificados que forman anillos o circuitos se alimentan desde uno o varios suministros y conducen el agua entre ellos o desde ellos para los nudos y extremos finales por más de un recorrido posible.

En puntos determinados de la red pueden ocurrir descargas o salidas de agua además de las posibles ramificaciones. Esos puntos se denominan nudos de consumo. Pero también es un nudo el punto donde cambian las características del conducto con su diámetro o su rugosidad así no haya consumo o ramificación.

Figura 3 Caudales consumidos en los nodos y caudales de alimentación a la red.



Fuente: Hidráulica de tuberías, Abastecimientos de redes, riegos.

Si se considera la red cerrada que aparece en la figura (3) y se tienen en cuenta que $Q_{D1}, Q_{D2}, Q_{D3}, \dots, Q_{DNU}$ son los caudales consumidos en cada uno de los nodos, algunos de los caudales podrían tener un valor nulo en un momento dado, y que $Q_{E1}, Q_{E2}, \dots, Q_{EM}$ son los caudales que alimentan la red de distribución, se puede establecer la siguiente ecuación de conservación de la masa:

$$\sum_{i=1}^m Q_E = \sum_{i=1}^{NU} Q_D$$

Ecuación (14)

Donde NU es el número de conexiones (nodos) existentes en la red.

La ecuación (13) es una ecuación de conservación de masa para la red como un todo. Sin embargo, para cada uno de los nodos de la red se pueden establecer una ecuación similar, debido a que localmente también se debe cumplir el hecho de que la masa se conserve. Dicha ecuación es:

$$\sum_{j=1}^{NTi} Q_{ij} - Q_{Di} = 0$$

Ecuación (15)

Donde NTi es el número de tuberías que llegan al nodo i , y Q_{ij} representa el caudal que pasa por la tubería ij hacia el nodo i desde el nodo j . La convención adoptada por la práctica de la ingeniería hidráulica es que puede ser positivo (va hacia el nodo i) o negativo (sale de dicho nodo).

Para cada uno de los caudales Q_{ij} de la ecuación (14) se puede plantear la siguiente ecuación de conservación de la energía entre los nodos i y j , incluyendo las pérdidas por fricción y las pérdidas menores, en términos de las alturas piezométrica en dichos nodos:

$$H_j - H_i = \frac{10,674 * Q_{ij}^{1,852} * L_{ij}}{C^{1,852} * D_{ij}^{4,871}}$$

Ecuación (16)

En donde se ha utilizado la ecuación de Hazen-William para el cálculo de las pérdidas por fricción. En donde L es la longitud de la tubería en metros, Q_{ij} es el caudal de la tubería en metros/s y D es el diámetro en metros. Al aplicar esta fórmula debemos tener en cuenta las siguientes restricciones:

- El fluido debe ser agua a temperaturas normales.
- El diámetro debe ser superior o igual a 75 mm (3 pulg).

- La velocidad en las tuberías debe ser inferior a 3 m/s (10 pies/seg).
- Desarrollar únicamente para flujo turbulento.

Al utilizar esta fórmula se recomienda utilizar los siguientes coeficientes de rugosidad de acuerdo al material.

Tabla 13 Valores recomendados para el coeficiente de rugosidad.

MATERIAL	C
Acero y Asbesto Cemento	130
Acero con recubrimiento de mortero centrifugado	140
Cloruro de polivinilo (PVC)	140
Cobre	130
Hierro dúctil con recubrimiento de mortero centrifugado	140

Fuente: Manual de EPANET 2.0

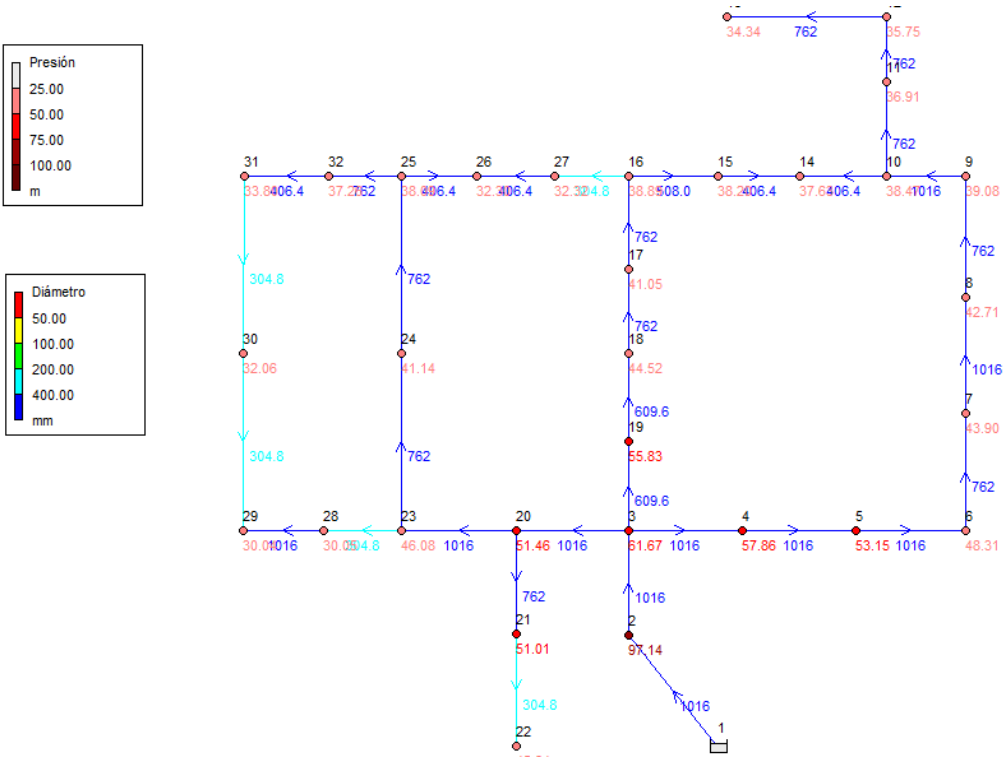
Los métodos de análisis de redes cerradas en general están diseñados para llevar a cabo esos cálculos de caudales en cada tubería y altura piezométrica en cada nodo. Esto implica que se deben conocer todas las demás variables relacionadas con las tuberías (diámetros, rugosidades, coeficientes de perdidas menores, accesorios especiales y bombas) y con los nodos (caudales de consumo, altura topográfica, tanques, caudales de emisor y embalses) [11].

3.4.2 METODOS TRADICIONALES DE DISEÑO EN LAS RDA'S.

El diseño de las RDA's consiste en asignar diámetros de tubería que cumplan con las condiciones hidráulicas del diseño, estos diseños se realizan con los caudales máximos horarios que depende de las zonas a servir además de la expansión de la población teniendo en cuenta la información topografía y geológica del terreno. Para el análisis hidráulico se puede utilizar el método de Hardy-Cross, el método del gradiente hidráulico o cualquier otro método racional. Los métodos de análisis de redes están diseñados para llevar a cabo esos cálculos de caudales en cada tubería y altura piezométrica en cada nodo. Esto implica que deben conocerse todas las demás variables que relacionadas con las tuberías (diámetros, rugosidades, coeficientes de perdidas menores, accesorios especiales y bombas) y con los nodos (caudales de consumo, altura topográfica, tanques, caudales de emisor y embalses). Por consiguiente, el cálculo de redes es por esencia una comprobación de diseño y no un diseño en sí. Sin embargo dicho calculo es vital ya que aun en el caso de programas para el diseño optimizado de redes o para su control en tiempo real, los métodos de comprobación de diseño deben conformar algunas de las unidades o subrutinas del programa [11].

EPANET permite realizar un análisis hidráulico de RDA's de manera semiautomática a partir de las características físicas de las tuberías y dinámicas de los nodos (consumos) para obtener la presión y caudales en nodos y tuberías respectivamente, sin embargo para realizar diferentes análisis hidráulicos se deben cambiar las variables de diseño manualmente haciendo que el número de iteraciones para encontrar una solución optima sea mínima además no cuenta con una herramienta que permita relacionar el costo del diseño con la combinación de tuberías que se seleccione. A continuación se presenta un diseño realizado de la red de Hanoi utilizando EPANET 2.0.

Figura 4 Red de Hanoi en EPANET 2.0



Fuente: Figura obtenida de EPANET 2.0.

Tabla 14 Resultados de EPANET 2.0 para los nodos de la red de Hanoi.

Tabla de Red - Nodos							
ID Nudo	Demanda LPS	Altura m	Presión m	ID Nudo	Demanda LPS	Altura m	Presión m
Conexión 2	247.22	97.14	97.14	Conexión 18	373.61	44.52	44.52
Conexión 3	236.11	61.67	61.67	Conexión 19	16.67	55.83	55.83
Conexión 4	36.11	57.86	57.86	Conexión 20	354.17	51.46	51.46
Conexión 5	201.39	53.15	53.15	Conexión 21	258.33	51.01	51.01
Conexión 6	279.17	48.31	48.31	Conexión 22	134.72	45.84	45.84
Conexión 7	375	43.9	43.9	Conexión 23	290.28	46.08	46.08
Conexión 8	152.78	42.71	42.71	Conexión 24	227.78	41.14	41.14
Conexión 9	145.83	39.08	39.08	Conexión 25	47.22	38.09	38.09
Conexión 10	145.83	38.47	38.47	Conexión 26	250	32.3	32.3
Conexión 11	138.89	36.91	36.91	Conexión 27	102.78	32.32	32.32
Conexión 12	155.56	35.75	35.75	Conexión 28	80.56	30.05	30.05
Conexión 13	261.11	34.34	34.34	Conexión 29	100	30.04	30.04
Conexión 14	170.83	37.63	37.63	Conexión 30	100	32.06	32.06
Conexión 15	77.78	38.2	38.2	Conexión 31	29.17	33.8	33.8
Conexión 16	86.11	38.89	38.89	Conexión 32	223.61	37.26	37.26
Conexión 17	240.28	41.05	41.05	Embalse 1	-5538.9	100	0

Fuente: Tabla obtenida de EPANET 2.0.

Tabla 15 Resultados de EPANET 2.0 para las tuberías de la red de Hanoi.

Tabla de Red - Líneas							
ID Línea	Longitud m	Diámetro mm	Caudal LPS	ID Línea	Longitud m	Diámetro mm	Caudal LPS
Tubería 1	100	1016	5538.9	Tubería 18	800	609.6	-988.43
Tubería 2	1350	1016	5291.68	Tubería 19	400	609.6	-1005.1
Tubería 3	900	1016	1975.13	Tubería 20	2200	1016	2075.35
Tubería 4	1150	1016	1939.02	Tubería 21	1500	762	393.05
Tubería 5	1450	1016	1737.63	Tubería 22	500	304.8	134.72
Tubería 6	450	762	1458.46	Tubería 23	2650	1016	1328.13
Tubería 7	850	1016	1083.46	Tubería 24	1230	762	900.65
Tubería 8	850	762	930.68	Tubería 25	1300	762	672.87
Tubería 9	800	1016	784.85	Tubería 26	850	406.4	229.51
Tubería 10	950	762	555.56	Tubería 27	300	406.4	-20.49
Tubería 11	1200	762	416.67	Tubería 28	750	304.8	-123.27
Tubería 12	3500	762	261.11	Tubería 29	1500	304.8	137.2
Tubería 13	800	406.4	83.46	Tubería 30	2000	1016	56.64
Tubería 14	500	406.4	-87.37	Tubería 31	1600	304.8	-43.36
Tubería 15	550	508	-165.15	Tubería 32	150	304.8	-143.36
Tubería 16	2730	762	-374.54	Tubería 33	860	406.4	-172.53
Tubería 17	1750	762	-614.82	Tubería 34	950	762	-396.14

Fuente: Tabla obtenida de EPANET 2.0.

Los signos negativos en los caudales de las tuberías indican que van en sentido contrario del orden de los nodos, todas las presiones cumplen con las restricción impuesta por el diseño (presiones por encima de 30 mca) y el costo de la red con los diámetros mostrados en la tabla (18) es de \$7, 357,039.18.

3.4.3 EPANET 2.0 “TOOLKIT”

EPANET es un programa de ordenador que realiza simulaciones en periodo extendido del comportamiento hidráulico y de la calidad del agua en redes de distribución a presión, es de dominio público y puede descargarse de manera gratuita en internet. Se puede encontrar en dos formas ejecutable y código fuente (toolkit)⁵. Este último contiene un kit de herramientas la cual es una librería de enlace dinámico (DLL) de funciones que permiten a los desarrolladores personalizar el motor computacional EPANET de acuerdo a sus propias necesidades. Las funciones pueden ser incorporados en aplicaciones de 32 bits de Windows escrito en C / C + +, Delphi, Pascal, Visual Basic, MATLAB o cualquier otra lengua que pueda llamar a funciones dentro de una DLL de Windows. Hay más de 50 funciones que se pueden utilizar para abrir un archivo inp de descripción de la red, leer y modificar el diseño de redes y diversos parámetros operativos ejecutar múltiples simulaciones de periodo extendido acceso a resultados a medida que se generan o guardarlos en un archivo y escribir los resultados seleccionados a un presentar en un formato especificado por el usuario.

Los archivos de extensión inp tienen la información que describen las redes, la biblioteca contenida en el toolkit permiten cambiar y ajustar parámetros o hacer

⁵ Disponibles en <<http://www.epa.gov/nrmrl/wswrd/dw/epanet.html>>

simulaciones de acuerdo a nuestras necesidades además de acceder fácilmente a los resultados y guardar la información

3.4.4 LLAMAR A EPANET DESDE MATLAB

MATLAB cuenta con una página de internet en la cual se resuelven dudas y demostraciones de varias aplicaciones que pueden generarse en este software [16]. Para acceder específicamente a esta función y enlazar MATLAB con EPANET es necesario agregar el archivo EPANET.DLL en la carpeta donde se almacenara la información del algoritmo y la red a tratar, el código utilizado para llamar EPANET desde el algoritmo creado en MATLAB se encuentra en la pagina central de ⁶MATLAB.

Esta es una extensión de la `getwdsdata.m` originalmente creado por Philip Jonkergouw en 2007. Capta la mayor parte de los comandos permitidos por EPANET esta modificación fue realizada por el señor Demeterio Eliades en el 2009 investigador Centro de Investigación KIOS para Sistemas Inteligentes y Redes, en el Electrical and Computer Engineering Department , Universidad de Chipre.

Sin embargo en este caso en particular en la cual los diámetros deben variar de acuerdo a las soluciones encontradas por el algoritmo OCH se debe hacer una modificación en este código agregando las siguientes líneas:

```
-index = int32 (índice);% el índice de enlace  
-paramcode = int32 (0);% 0 es el código para diámetros
```

⁶ Disponibles en <<http://www.mathworks.com/matlabcentral/fileexchange/25100-epanet-matlab-toolkit>>

-valor único (valor)= valor% es el valor que desea establecer el diámetro
-[errcode] = calllib ('epanet2', 'ENsetlinkvalue', índice, paramcode, valor)

Si se presentan problemas para llamar esta función también se pueden modificar los archivos inp en MATLAB con las funciones *fopen* y *fclose* que permiten abrir y cerrar ficheros. La función *fopen* tiene la siguiente forma:

$$[fi, texto] = fopen('filename', 'w')$$

donde *fi* es un valor de retorno que sirve como identificador del fichero, *texto* es un mensaje para caso de que se produzca un error, y *w* es un carácter (o dos) que indica el tipo de operación que se desea realizar. Las opciones más importantes son las siguientes:

- 'r' → lectura (de *read*)
- 'w' → escritura reemplazando (de *write*)
- 'a' → escritura a continuación (de *append*)
- 'r+' → lectura y escritura

Es importante recalcar que se pueden hacer diferentes modificaciones dependiendo del problema que se quiera realizar. Con estos códigos se obtienen los caudales y presiones, las segundas son de gran importancia ya que serán estas las que permitan verificar si el algoritmo está cumpliendo con las restricciones que se le adjudiquen a la red de distribución de agua.

4. APLICACIÓN DEL ALGORITMO OCH-EPANET A LAS RDA's

4.1 LAS RDA's COMO UN PROBLEMA DE OPTIMIZACIÓN.

Una Red de Distribución de Agua (RDA's) es una red de componentes (por ejemplo tuberías, bombas, válvulas, tanques etc) que transporta agua desde una fuente (por ejemplo, depósitos, plantas de tratamiento, tanques, etc) a los consumidores. La optimización de las RDA's son definidas por lo general como la selección de menor costo generado por sus componentes de tal amera que los criterios de demanda y limitaciones de diseño sean satisfechas. En la practica el diseño de las RDA's pueden tomar muchas formas y tomar diferentes criterios. Por ejemplo, la combinación de diferentes diámetros de tubería, el tamaño de los tanques, la selección de ajuste de presión de válvula y su ubicación entre otras. Además de las posibles variables de diseño, las demandas del sistema podrían incluir una variedad de casos incluyendo las horas pico. Las restricciones del sistema se pueden especificar con las presiones mínimas y máximas en cada nodo de la red, las velocidades máximas permitidas para cada tubería y la calidad del agua. Sin embargo, la literatura sobre la optimización de RDA's ha tratado tradicionalmente con un problema más simplista e idealizado. Las variables de decisión han sido principalmente asociadas a las tuberías [2].

Diseñar consiste en determinar el diámetro que debe tener cada una de las tuberías de la red, de manera que sea posible llevar el agua desde fuentes hasta cada nodo manteniendo una presión por encima de la presión minina permitida. El diseño de redes de acueducto se puede modelar como problema de optimización combinatoria en donde las variables de decisión son los diámetros de cada uno de

las tuberías. El problema consiste en determinar un conjunto de diámetros tal que se minimice una función de costos (basada en la longitud y costo de tubería) sujeta a restricciones hidráulicas, comerciales, etc. [11]

Restricciones Hidráulicas

- Ley de conservación de masa en cada nodo.
- Ley de conservación de energía en cada circuito.
- La presión en cada nodo debe ser mayor al valor mínimo.

Restricciones comerciales:

- Los diámetros solo pueden tomar valores discretos dados por la disponibilidad de diámetros de cada fabricante de tuberías.

El problema de optimización es decir la minimización de costos se puede expresar de la siguiente manera.

$$\min C(\varphi) = \sum_{i=1}^n UC_{\phi} LEi$$

Ecuación (17)

Sujeto a

$$H_i(\varphi) \geq \bar{H}_i$$

Ecuación (18)

$$QD_i + \sum_{j=1}^{NTi} Qi_{out} - \sum_{j=1}^{NTi} Qi_{in}$$

Ecuación (19)

$$H_i(\varphi) = 10,674 \frac{LEi}{C^{1,852} D_i^{4,871}} Q_i^{1,852}$$

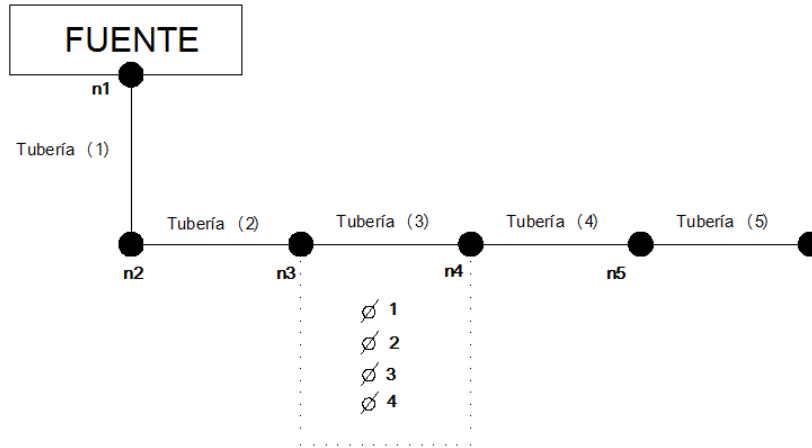
Ecuación (20)

Donde la ecuación 16 es la función objetivo, $C(\varphi)$ es el costo del diseño, LE_i es la longitud de la tubería i y UC_{\emptyset} es el costo unitario por metro de cada una de las tuberías; La ecuación 17 es la restricción de diseño donde $H_i(\varphi)$ es la cabeza de presión real en el nodo i y diseño φ , \overline{H}_i es la cabeza de presión mínima permitida en el nodo i . Además de las restricciones de diseño las ecuaciones para el análisis de redes cerradas se deben cumplir. Estas son las ecuaciones de continuidad nodal y pérdida de carga en las ecuaciones 18 y 19. En 18 QD_i es el caudal de demanda en el nodo i , $Q_{j_{out}}$ es la sumatoria de caudales de salida en el nodo i y $Q_{i_{in}}$ es la sumatoria de caudales de entrada en el nodo i . 19 En este estudio se utilizó la ecuación de Hazen-Williams donde $H_i(\varphi)$ es la pérdida de carga en el nodo i , D_i es el diámetro de la tubería seleccionada para el diseño φ , C es el coeficiente de rugosidad de la tubería del diseño [2].

Es importante aclarar que para la práctica solo tendremos en cuenta la ecuación 17 ya que las ecuaciones 18 y 9 estarán presentes en el solucionador hidráulico del cuál obtendremos $H_i(\varphi)$ y por correspondiente $Q_i(\varphi)$.

Como se señala en [5], la clave para la aplicación de los algoritmos OCH a los nuevos problemas es identificar una representación adecuada a través de un grafo $G = (D, L, C)$. Para la aplicación de los algoritmos OCH a los problemas de optimización de las RDA's, el grafo debe tomar una forma mostrada en la figura 5:

Figura 5 Representación gráfica de la optimización de distribución de Agua.



Fuente: Application of two ant colony optimization algorithms to water distribution system optimization

En el ejemplo de la figura 5, hay 5 tuberías y 5 puntos de decisión (n1,n2,n3,n4,n5) en los cuales existen 4 posibles opciones de diámetro ($\emptyset 1, \emptyset 2, \emptyset 3, \emptyset 4$) para cada una de ellas, correspondientes a 4 opciones en cada punto de decisión ($li(1), li(2), li(3), li(4), i = 1, \dots, 5$). Los costos asociados a cada una de estas opciones ($ci(1), \dots, ci(4), i = 1, \dots, 4$) son el producto unitario por metro de largo de cada uno de los diámetros de las tuberías $UC_{\emptyset j}$ y la longitud del segmento de tubería en cuestión LE_i . El costo asociado con una solución de prueba en particular viene dado por:

$$C(\varphi) = \sum_{i=1}^n UC_{\emptyset j} \cdot LE_i$$

Ecuación (21)

Como el objetivo es minimizar el costo las opciones de menor costo son mas deseables. Por lo tanto, la conveniencia de una opción se toma como la inversa de los gastos de ejecución, en otras palabras:

$$n_{ij} = \frac{1}{C(\varphi)}$$

Ecuación (22)

La optimización de las RDA's es diferente de la utilizada en las otras optimizaciones combinatorias como en el caso del problema del viajante mostrado en [5], en las restricciones realizadas. Por ejemplo, en el problema del viajante de comercio las únicas restricciones son que cada ciudad debe ser visitada una sola vez y que el punto final debe ser el mismo que el punto de partida. En esta situación, las listas tabú se utilizan para garantizar que sólo las soluciones factibles se generan. Sin embargo, las limitaciones que deben ser satisfechas en el diseño óptimo de RDA's son de naturaleza diferente. La viabilidad de una solución de prueba (por ejemplo, si las restricciones de presión mínima se han satisfecho) sólo puede ser evaluada después de que ha construido en su totalidad y en consecuencia, las restricciones no pueden ser tomadas en cuenta explícitamente durante la construcción de las soluciones de prueba.

Los algoritmos heurísticos dirigen exclusivamente su búsqueda sobre la información proporcionada por la función objetivo. Para dirigir la búsqueda lejos de los valores restringidos hacia los que cumplen con las restricciones la función de penalización aumenta el valor de los costos en los que se consideran indeseables. El problema de optimización sin restricciones para los RDA's toma la forma de minimizar la suma del costo real más el costo de penalización. Con el fin de dar un refuerzo negativo a las opciones de diámetro de la tubería que se traducen en

soluciones que violan las restricciones de presión, se modifica la ecuación de la feromona de la siguiente manera:

$$\Delta\tau_{i,j} = \frac{Q}{LT_k} - Ppher \cdot \Delta H_{max}$$

Ecuación (23)

Donde $Ppher=$ es un factor de penalización de las feromonas y ΔH_{max} es el déficit de presión máxima en la RDA's que se obtiene mediante el solucionador hidráulico para cada solución de prueba (es decir la combinación de tuberías generadas) [2]. Con el fin de minimizar la suma del costo real más el costo de penalización se modifica la ecuación 20 de la siguiente manera:

$$C(\varphi) = \sum_{i=1}^n UC_{\emptyset_j} \cdot LE_i + PC$$

Ecuación (24)

Donde $C(\varphi)$ es el costo de la red, $UC_{\emptyset_j} \cdot LE_i$ es el costo del material y la instalación de la red y PC es el costo incurrido por la penalización, como se indica en [1] PC fue llevado a ser proporcional al déficit de presión máxima nodal, esto es:

$$PC(\Omega) = \Delta H_{max} \cdot PEN$$

Ecuación (25)

PEN es un parámetro definido por el usuario y los valores apropiados de PEN son diferentes para cada caso de estudio. Para reducir los requisitos de calibración,

una expresión semi - determinístico para este factor se encuentra en [8] en la siguiente ecuación:

$$PEN = \frac{[C(\Omega^{max}) - C(\Omega^{min})]}{d}$$

Ecuación (26)

Donde Ω^{max} y Ω^{min} son el máximo y el mínimo costo de material de los diseños de red, respectivamente, y d es un déficit de presión seleccionado por el usuario. El valor de PEN asegura que todas las redes con una violación presión mayor que o igual a d (un valor extremadamente pequeño) se hacen más caro que el costo de la red factible máximo.

4.2 EJEMPLO RED DE HANOI

El algoritmo OCH fue desarrollado en MATLAB junto con el motor hidráulico que en este caso se utilizó, el TOOLKIT de EPANET, los parámetros se tomaron de [1] ya que de este artículo se tomó el ejemplo para poder comparar los resultados con un ejemplo conocido en la literatura.

4.2.1 Descripción de la red

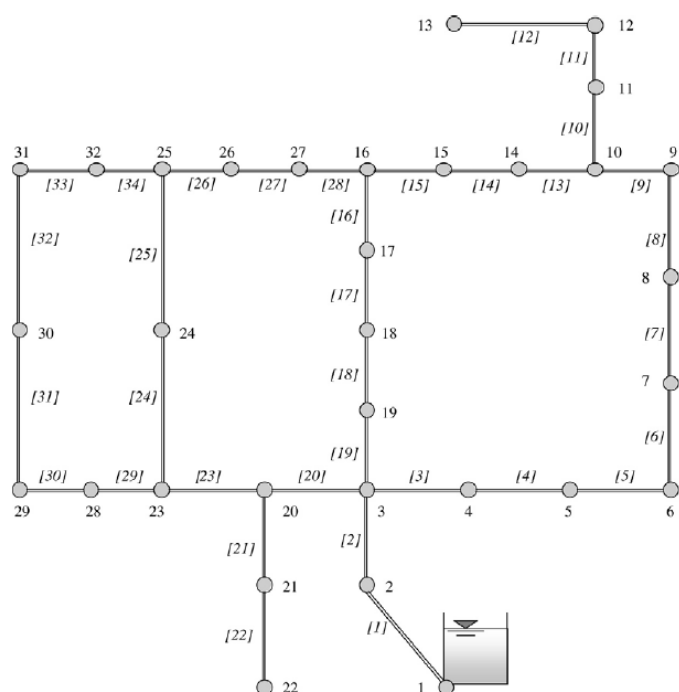
El problema de la red de Hanoi, es un caso conocido en la literatura ya que ha sido considerado por varios autores para la aplicación de soluciones a las optimizaciones combinatorias [1, 11, 12, 13,14].

El problema de Hanoi es un nuevo diseño ya que no hay tuberías existentes en el sistema. La red consta de 34 tuberías y 32 nodos organizados en 3 ciclos fig.(5). El sistema está alimentado por gravedad por un solo depósito, los detalles de la red se dan en la tabla (16) y la tabla (17). Para cada enlace hay 6 opciones diferentes de tuberías nuevas en donde es obligatorio una restricción de diámetro mínimo (es decir sin opción nula). En la tabla (18) se presentan las opciones de diseño de la red: En este caso el problema tiene un tamaño aproximado de $2,78 \times 10^{26}$ posibles diseños (combinación de tuberías). Los parámetros se establecen de la siguiente manera: $\tau_0 = 26$ (para SH), $\alpha=1.0$, $\beta=0.5$, $\rho=0.98$, $m=80$, $Q = 1.1 \times 10^7$ (para ambos algoritmos), $P_{best} = 0.5$, $d = 0$ (para MMSH).

La mejor solución conocida en la literatura es de US \$ 6,182 millones encontrado por el método del algoritmo genético [11,12]. Es importante señalar que otros autores han encontrado soluciones más económicas [13,14]. Pero estos se determinaron como inviable por EPANET 2.0 es decir no cumple con las restricciones de diseño.

En este ejemplo para el cálculo de pérdidas por fricción se usa la ecuación de Hazen-williams con un coeficiente de 130 para todas las tuberías de la red. La línea del gradiente hidráulico es de 100 mca y la presión mínima es de 30 mca.

Figura 6 Red de Hanoi.



Fuente: Application of two ant colony optimization algorithms to water distribution system optimization.

Tabla 16 Demanda de los nodos de la red de Hanoi, (el depósito tiene una altura de 100 m).

ID	Demanda (L/s)	ID	Demanda (L/s)	ID	Demanda (L/s)	ID	Demanda (L/s)
2	247.22	10	145.83	18	373.61	26	250.00
3	236.11	11	138.89	19	16.67	27	102.78
4	36.11	12	155.56	20	354.17	28	80.56
5	201.39	13	261.11	21	258.33	29	100.00
6	279.17	14	170.83	22	134.72	30	100.00
7	375.00	15	77.78	23	290.28	31	29.17
8	152.78	16	86.11	24	227.78	32	223.61
9	145.83	17	240.28	25	47.22		

Fuente: Application of two ant colony optimization algorithms to water distribution system optimization [1].

Tabla 17 Longitudes de las tuberías en la red de Hanoi.(Todas las tuberías tienen una coeficiente de rugosidad de 130)

ID	Longitud (m)	ID	Longitud (m)	ID	Longitud (m)	ID	Longitud (m)
1	100	10	950	19	400	28	750
2	1350	11	1200	20	2200	29	1500
3	900	12	3500	21	1500	30	2000
4	1150	13	800	22	500	31	1600
5	1450	14	500	23	2650	32	150
6	450	15	550	24	1230	33	860
7	850	16	2730	25	1300	34	950
8	850	17	1750	26	850		
9	800	18	800	27	300		

Fuente: Application of two ant colony optimization algorithms to water distribution system optimization [1].

Tabla 18 Diámetros comerciales y costos para la red de Hanoi

# de diámetro	1	2	3	4	5	6
Diámetro (mm)	304.8	406.4	508.0	609.6	762.0	1016.0
Costo (US \$/m)	45.726	70.4	98.378	129.333	180.748	278.280

Fuente: Application of two ant colony optimization algorithms to water distribution system optimization [1].

5. VALIDACIÓN DEL PROGRAMA Y RESULTADOS.

El algoritmo SH y MMSH fueron codificados en ⁷MATLAB con EPANET 2.0 como solucionador hidráulico. En cuanto a la configuración de los parámetros se tomaron de [1] y [8] de los cuales se encuentran los valores de $\alpha, \beta, \rho, \tau_0, m$ y Q (nota para MMAS τ_0 , es establecida en un elevado número arbitrario en este estudio se tomó el valor de 50, haciendo posible que para las siguientes iteraciones los valores de la feromona fueran τ_{max}) Para MMAS f_{global} Se establece en 5, y ρ_{best} y d se tomaron de [1] sin embargo se hicieron pruebas donde se variaron los valores de ρ_{best} para observar su relevancia en la solución. El déficit de presión no se incluye en [1] aclarando que es un valor estipulado por el usuario, se tomaron diferentes valores con el fin de encontrar cuál generaba un mejor combinación económica como eficiente.

Para ejecutar el algoritmo OCH es necesario utilizar un ordenador con un procesador con alto rendimiento. En este trabajo se utilizó un computador con un procesador Intel® Core™ i5-3330 Processor (6M Cache, up to 3.20 GHz) y dos núcleos.

⁷ Información Licencia MATLAB: MATLAB R2012b, MATrix LABoratory [Software License: 691503]. The Math Works, Inc. 2012.

5.1 RESULTADOS.

5.1.1 RESULTADOS DEL ALGORITMO SH.

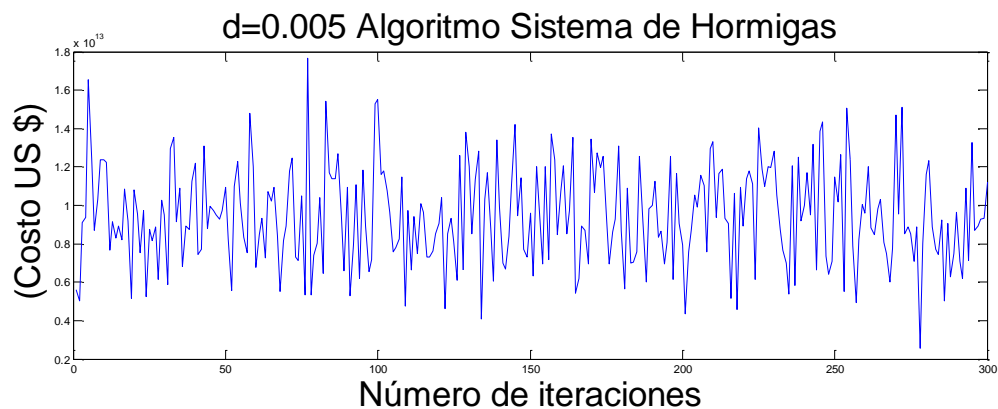
La mejor solución que ha encontrado el algoritmo después de 300 iteraciones solo 7 de los 32 nodos cumplen con la restricción, es decir no es una solución factible ver tabla (19).

El tiempo que tardo el computador en realizar las 300 iteraciones del SH fue de 2.4137 horas.

Luego de las 300 iteraciones no ha ocurrido un cambio significativo en la probabilidad de elección del diámetro para cada tubería ver tablas (20) y (21) .

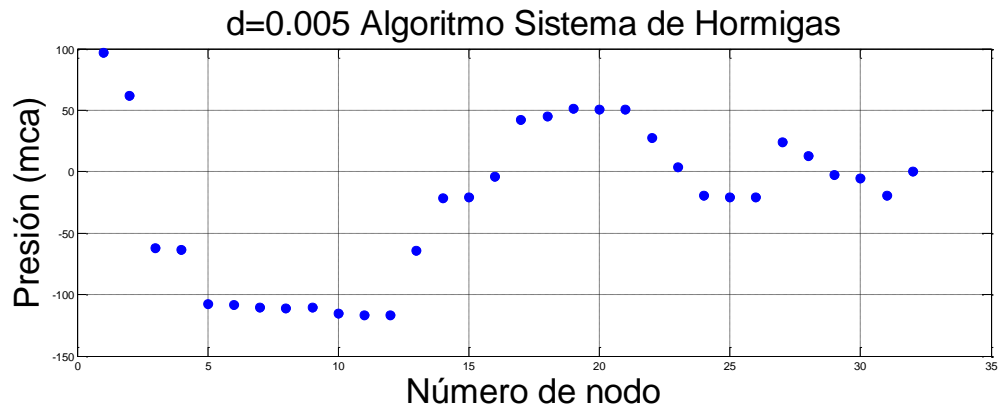
Debido a que no se evidencia un aprendizaje por parte de SH es necesario acudir a otro derivado de OCH que direcciona la búsqueda de una forma más rápida. Siguiendo lo sugerido en [1] se busco solución del problema mediante MMSH.

Figura 7 Comportamiento del algoritmo Sistema de Hormigas Costo (US \$) vs número de iteraciones



Fuente: Figura obtenida de MATLAB.

Figura 8 Presiones del algoritmo Sistema de hormigas en cada nodo



Fuente: Figura obtenida de MATLAB.

Tabla 19 Resultados del algoritmo SH.

Algoritmo Sistema de Hormigas d=0.005 300 iteraciones									
nodo	presión (mca)	>30 mca	tubería	# tubería	nodo	presión (mca)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	45.0021	ok	18	6
3	61.6711	ok	2	6	20	51.0256	ok	19	4
4	-62.1303	no	3	2	21	50.7062	ok	20	6
5	-63.8499	no	4	6	22	50.6915	ok	21	6
6	-107.8933	no	5	3	23	27.716	no	22	6
7	-108.1333	no	6	6	24	3.3885	no	23	5
8	-110.7801	no	7	3	25	-19.5638	no	24	2
9	-110.8586	no	8	5	26	-20.7011	no	25	1
10	-110.7448	no	9	2	27	-20.7016	no	26	4
11	-115.3688	no	10	4	28	24.1626	no	27	6
12	-116.525	no	11	5	29	13.0512	no	28	6
13	-116.8745	no	12	6	30	-2.3308	no	29	5
14	-64.2385	no	13	2	31	-5.1506	no	30	4
15	-21.3281	no	14	2	32	-19.1849	no	31	3
16	-20.6939	no	15	6	deposito	0	no	32	2
17	-3.7111	no	16	5				33	2
18	42.2813	ok	17	4				34	4
restricción	cumplen	3							
	no cumplen	14	Tiempo de SH	2.4137(hr)					

Tabla 20 Probabilidades iniciales del algoritmo SH.

Probabilidades iniciales de algoritmo Sistema de Hormigas									
Columns 1 through 10									
0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506
0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020
0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708
0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490
0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260
0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016
Columns 11 through 20									
0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506
0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020
0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708
0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490
0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260
0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016
Columns 21 through 30									
0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506
0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020	0.2020
0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708	0.1708
0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490	0.1490
0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260	0.1260
0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016	0.1016
Columns 31 through 34									
0.2506	0.2506	0.2506	0.2506						
0.2020	0.2020	0.2020	0.2020						
0.1708	0.1708	0.1708	0.1708						
0.1490	0.1490	0.1490	0.1490						
0.1260	0.1260	0.1260	0.1260						
0.1016	0.1016	0.1016	0.1016						

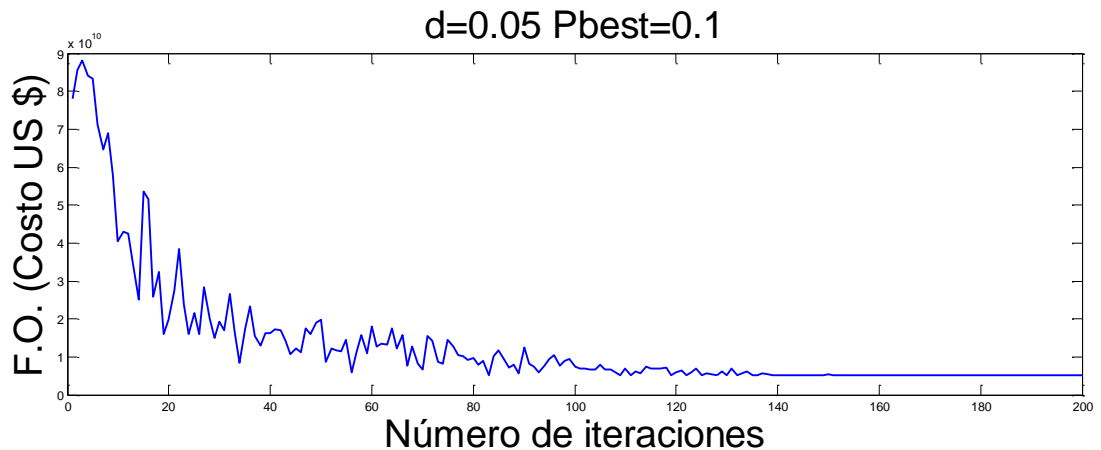
Tabla 21 Probabilidades finales del algoritmo SH.

Probabilidades finales de algoritmo Sistema de Hormigas										
Columns 1 through 11										
0.2502	0.2461	0.2525	0.2521	0.2526	0.2531	0.2537	0.2533	0.2536	0.2528	0.2532
0.2028	0.2006	0.2027	0.2029	0.2025	0.2027	0.2023	0.2027	0.2026	0.2028	0.2029
0.1716	0.1716	0.1709	0.1709	0.1709	0.1707	0.1707	0.1706	0.1706	0.1706	0.1704
0.1490	0.1510	0.1483	0.1484	0.1486	0.1484	0.1481	0.1483	0.1481	0.1486	0.1482
0.1257	0.1280	0.1250	0.1252	0.1250	0.1250	0.1248	0.1249	0.1251	0.1249	0.1250
0.1006	0.1027	0.1005	0.1004	0.1003	0.1002	0.1003	0.1002	0.1000	0.1002	0.1002
Columns 12 through 22										
0.2533	0.2527	0.2535	0.2537	0.2528	0.2525	0.2525	0.2525	0.2514	0.2529	0.2538
0.2025	0.2030	0.2028	0.2027	0.2023	0.2026	0.2029	0.2029	0.2025	0.2028	0.2027
0.1705	0.1708	0.1705	0.1701	0.1707	0.1709	0.1708	0.1708	0.1713	0.1706	0.1707
0.1482	0.1486	0.1483	0.1482	0.1486	0.1486	0.1485	0.1484	0.1490	0.1485	0.1482
0.1252	0.1248	0.1248	0.1249	0.1253	0.1251	0.1251	0.1252	0.1252	0.1249	0.1246
0.1003	0.1001	0.1001	0.1003	0.1004	0.1003	0.1002	0.1003	0.1006	0.1003	0.1001
Columns 23 through 33										
0.2520	0.2539	0.2536	0.2537	0.2537	0.2536	0.2538	0.2535	0.2534	0.2534	0.2532
0.2028	0.2026	0.2027	0.2026	0.2024	0.2027	0.2026	0.2024	0.2027	0.2028	0.2026
0.1709	0.1706	0.1705	0.1705	0.1704	0.1705	0.1703	0.1704	0.1707	0.1708	0.1707
0.1485	0.1481	0.1483	0.1481	0.1482	0.1481	0.1482	0.1485	0.1483	0.1480	0.1482
0.1253	0.1245	0.1249	0.1250	0.1251	0.1247	0.1251	0.1249	0.1248	0.1249	0.1250
0.1006	0.1003	0.1000	0.1002	0.1002	0.1004	0.1000	0.1003	0.1001	0.1001	0.1003
Column 34										
0.2533										
0.2025										
0.1709										
0.1483										
0.1249										
0.1002										

5.1.2 Resultados de la variación de P_{best} con un $d= 0.5$

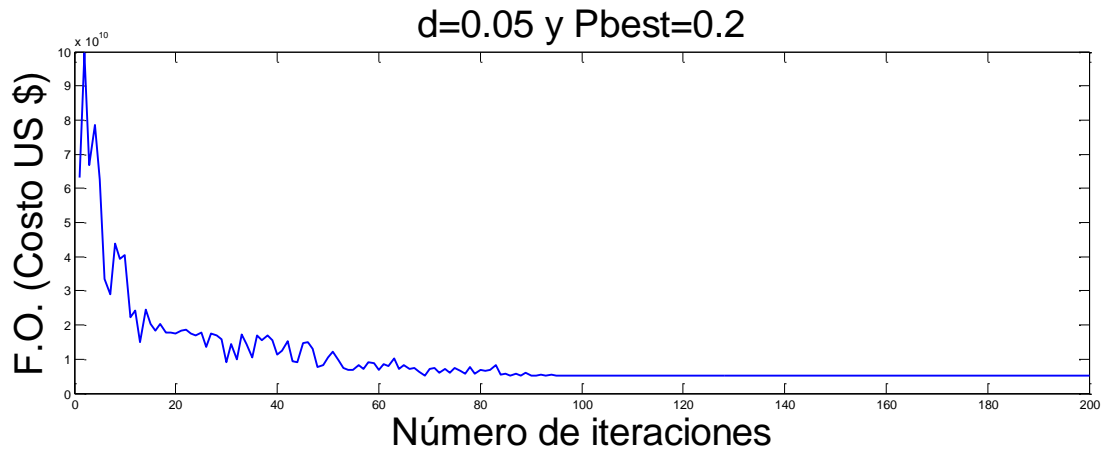
El factor P_{best} define el ancho entre los límites de las feromonas, como se muestra en la figura (9) un valor de P_{best} muy bajo hace que el algoritmo se demore más en generar las soluciones factibles por el contrario un valor de P_{best} muy grande como se ve en la figura (12) hace que estas soluciones descarten rápidamente aquellas que no cumplan las restricciones, en [1] se utiliza un P_{best} de 0.5 este valor está representado en la figura (11).

Figura 9 Comportamiento del algoritmo MMSH con un $P_{best}=0.1$



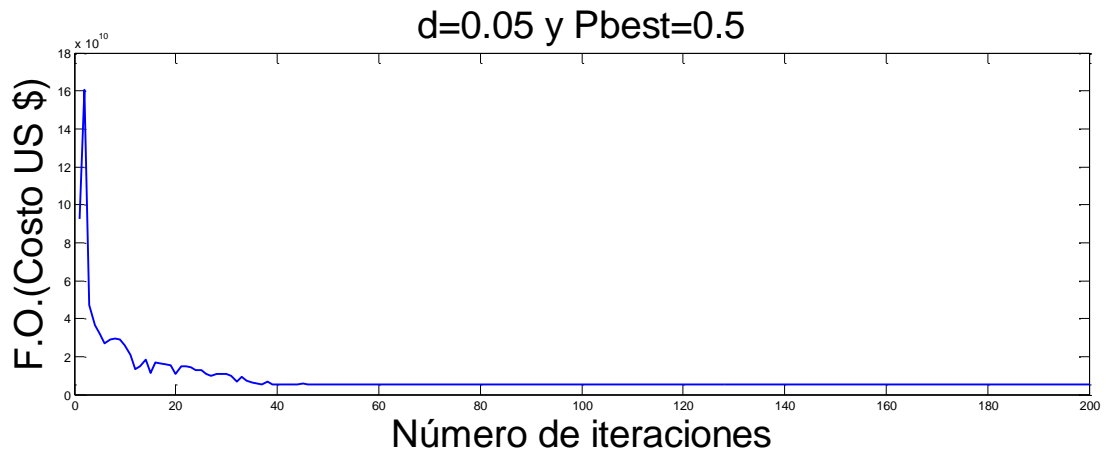
Fuente: Figura obtenida de MATLAB.

Figura 10 Comportamiento del algoritmo MMSH con un Pbest=0.2



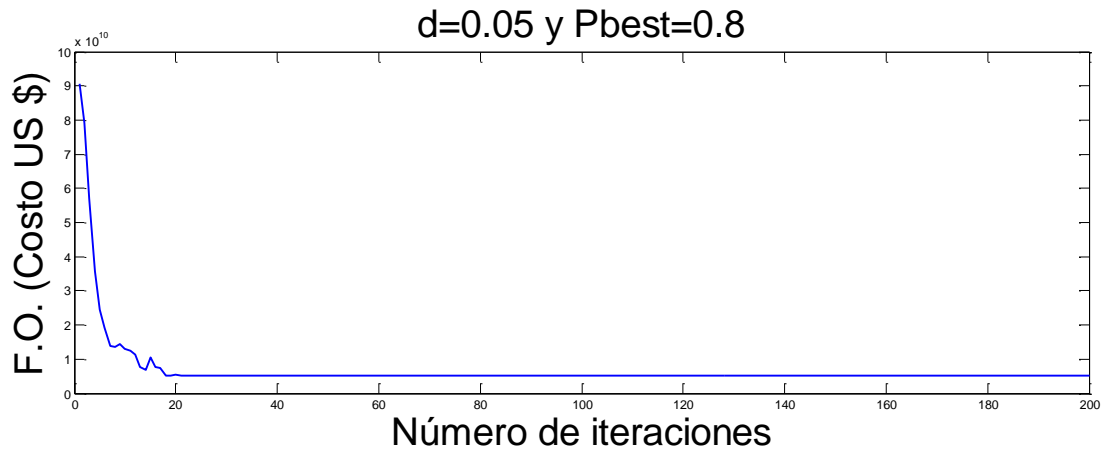
Fuente: Figura obtenida de MATLAB.

Figura 11 Comportamiento del algoritmo MMSH con un Pbest=0.5



Fuente: Figura obtenida de MATLAB.

Figura 12 Comportamiento del algoritmo MMSH con un Pbest=0.8



Fuente: Figura obtenida de MATLAB.

5.1.3 Resultados Del Algoritmo MMSH

En la bibliografía consultada no definen un valor para la frecuencia de actualización de la feromona para la mejor solución global. Con el fin de encontrar un valor adecuado para este parámetro se realizaron algunas pruebas del desempeño del algoritmo cuando este tiende a infinito, es decir solo se realiza la actualización de la feromona para la mejor solución encontrada en cada iteración.

5.1.3.1 Resultados MMSH sin tener en cuenta la actualización de la feromona de la mejor solución global

Para los diferentes valores de d el algoritmo presentaba un comportamiento similar. El algoritmo mejora sus soluciones más o menos hasta las iteraciones 125 a 130, luego de esto se desestabiliza y empieza una búsqueda similar a la realizada por SH.

El valor de d que consiguió la solución más económica fue 0.005 y el costo total de la solución final fue US \$6, 889,953.66.

Después de las 300 iteraciones la matriz de probabilidades fue igual a la matriz de probabilidad inicial para todos los valores de d .

Después de 300 iteraciones el rastro de feromona para todas las rutas es igual al feromona mínima final.

Los valores de feromona máxima y feromona mínima sufren un aumento abrupto en la iteración en que el algoritmo encuentra una solución factible (en la cual todos los nodos cumplen con la restricción de presión) ver figuras (21), (22), (23).

En la iteración 125 para un valor de $d=0.005$ el algoritmo encontró por primera vez una solución factible ver figura (15).

La duración de las 300 iteraciones del MMSH es aproximadamente 2.3 horas ver tablas (20, 21, 22,23).

Tabla 22 Resultado de la variación del parámetro d=0.05.

d=0.05 y Pbest=0.5 203 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	59.6205	ok	18	4
3	61.6711	ok	2	6	20	51.8732	ok	19	5
4	57.7751	ok	3	6	21	50.576	ok	20	6
5	52.9622	ok	4	6	22	50.1471	ok	21	5
6	47.9964	ok	5	6	23	46.8307	ok	22	3
7	46.8769	ok	6	6	24	40.5295	ok	23	6
8	45.6451	ok	7	6	25	37.9343	ok	24	4
9	41.846	ok	8	5	26	34.0964	ok	25	4
10	34.0479	ok	9	4	27	34.302	ok	26	2
11	33.6639	ok	10	6	28	42.4651	ok	27	2
12	33.3791	ok	11	6	29	41.1407	ok	28	5
13	31.96	ok	12	5	30	37.1319	ok	29	4
14	33.6005	ok	13	3	31	36.7205	ok	30	5
15	33.9201	ok	14	2	32	36.204	ok	31	3
16	34.4381	ok	15	3	deposito	0		32	2
17	36.8423	ok	16	5				33	3
18	47.8228	ok	17	4				34	2

costo total US \$ 7,379,470.76 / Tiempo de MMSH 1.981 (horas)

Tabla 23 Resultado de la variación del parámetro d=0.005.

d=0.005 y Pbest=0.5 300 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	52.7737	ok	18	6
3	61.6711	ok	2	6	20	52.7775	ok	19	4
4	58.2351	ok	3	6	21	48.931	ok	20	6
5	54.0007	ok	4	6	22	43.7663	ok	21	4
6	49.6978	ok	5	6	23	48.4619	ok	22	1
7	45.8688	ok	6	5	24	38.4942	ok	23	6
8	41.9084	ok	7	5	25	35.6074	ok	24	3
9	39.0192	ok	8	5	26	35.4924	ok	25	3
10	37.1242	ok	9	5	27	36.6632	ok	26	4
11	35.5648	ok	10	5	28	37.1488	ok	27	2
12	34.4086	ok	11	5	29	35.7155	ok	28	3
13	30.2005	ok	12	4	30	35.094	ok	29	3
14	37.4674	ok	13	1	31	34.9269	ok	30	5
15	38.3184	ok	14	3	32	34.2756	ok	31	5
16	39.0355	ok	15	4	Deposito	0		32	3
17	44.7156	ok	16	5				33	3
18	51.3316	ok	17	5				34	2

costo total US \$ 6,889,953.66 / Tiempo de MMSH 2.4141 (horas)

Tabla 24 Resultado de la variación del parámetro $d=0.0005$.

d=0.0005 y Pbest=0.5 300 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	59.6769	ok	18	6
3	61.6711	ok	2	6	20	53.8774	ok	19	6
4	56.3918	ok	3	5	21	44.5286	ok	20	6
5	50.0487	ok	4	5	22	39.3639	ok	21	3
6	44.6154	ok	5	5	23	50.417	ok	22	1
7	42.1796	ok	6	4	24	40.7376	ok	23	6
8	40.471	ok	7	3	25	32.6725	ok	24	3
9	38.4733	ok	8	1	26	34.1925	ok	25	2
10	38.5141	ok	9	5	27	37.6012	ok	26	1
11	33.8901	ok	10	4	28	32.5263	ok	27	2
12	33.6053	ok	11	6	29	31.8866	ok	28	3
13	29.3973	no	12	4	30	30.6067	ok	29	2
14	39.13	ok	13	6	31	30.5989	ok	30	5
15	39.6839	ok	14	6	32	30.5989	ok	31	3
16	42.5431	ok	15	5	deposito	0		32	3
17	49.7169	ok	16	6				33	5
18	55.7454	ok	17	6				34	3

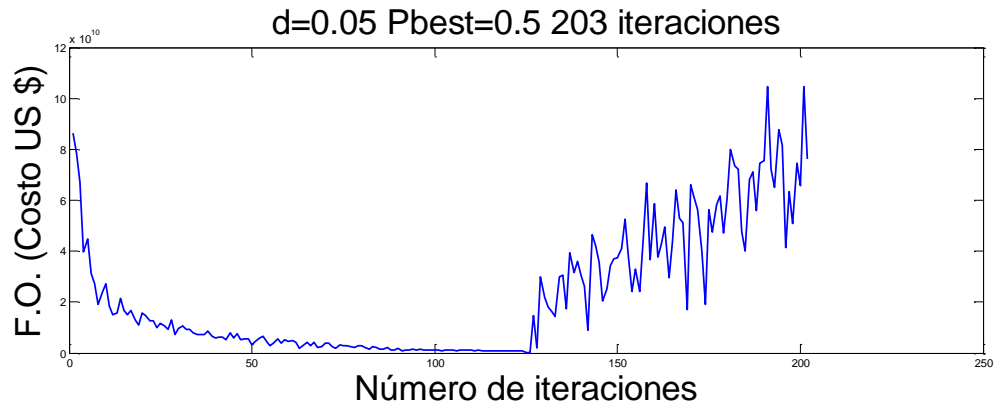
costo total US \$ 7,075,320.87 / Tiempo de MMSH 2.3388 (horas)

Tabla 25 Resultado de la variación del parámetro $d=0.0000005$.

d=0.0000005 y Pbest=0.5 300 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	52.5403	ok	18	5
3	61.6711	ok	2	6	20	53.7739	ok	19	4
4	57.8798	ok	3	6	21	49.9274	ok	20	6
5	53.1985	ok	4	6	22	44.7627	ok	21	4
6	48.384	ok	5	6	23	50.2346	ok	22	1
7	44.0028	ok	6	5	24	38.3556	ok	23	6
8	39.2423	ok	7	5	25	34.3102	ok	24	3
9	38.359	ok	8	6	26	35.2123	ok	25	3
10	35.9022	ok	9	5	27	36.2732	ok	26	1
11	35.5181	ok	10	6	28	36.3122	ok	27	3
12	32.0897	ok	11	4	29	34.972	ok	28	4
13	30.6706	ok	12	5	30	33.0267	ok	29	2
14	35.8002	ok	13	4	31	33.0382	ok	30	4
15	36.0164	ok	14	3	32	33.9216	ok	31	2
16	38.1889	ok	15	2	deposito	0		32	1
17	39.6615	ok	16	6				33	1
18	46.5294	ok	17	5				34	5

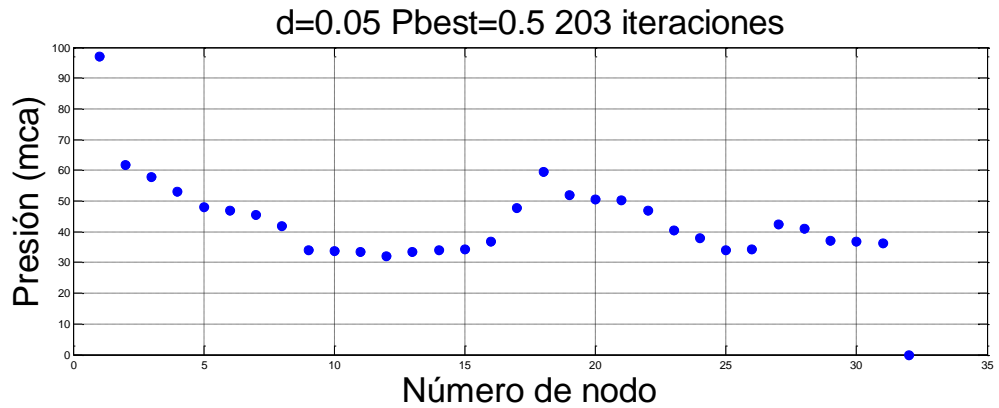
costo total US \$ 7,185,095.75 / Tiempo de MMSH 2.281 (horas)

Figura 13 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.05$ y $P_{best}=0.5$



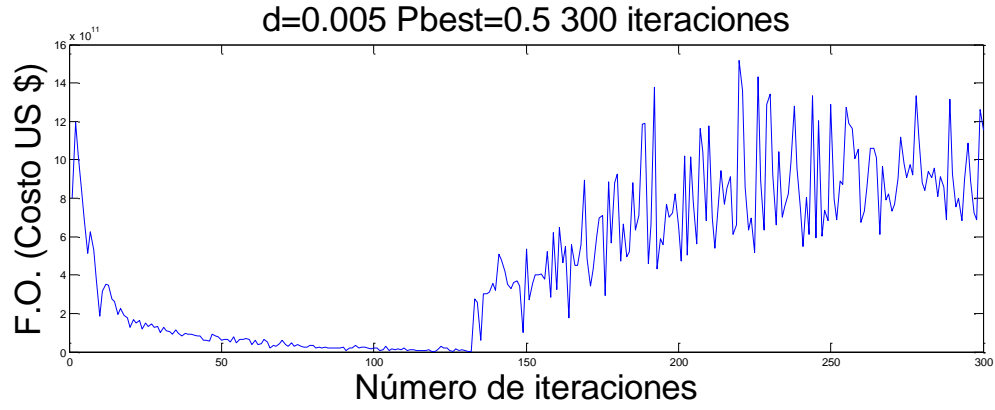
Fuente: Figura obtenida de MATLAB.

Figura 14 Presiones del algoritmo MMSH en cada nodo con un $d=0.05$ y $P_{best}=0.5$



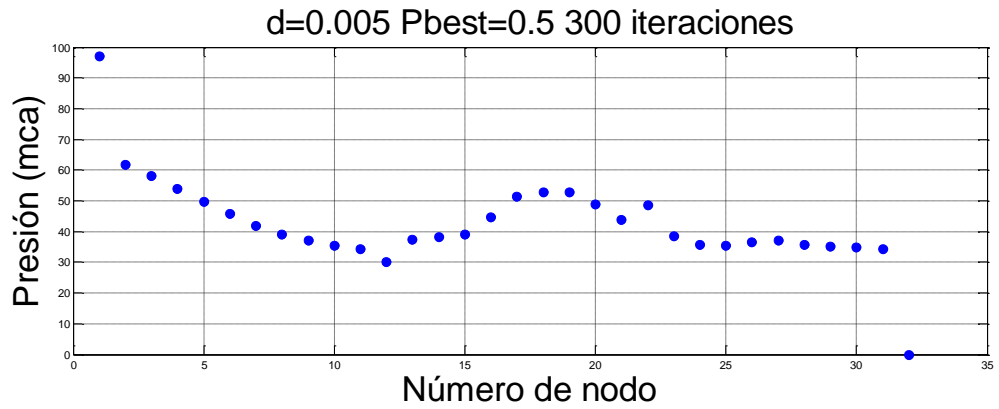
Fuente: Figura obtenida de MATLAB.

Figura 15 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.005$ y $P_{best}=0.5$



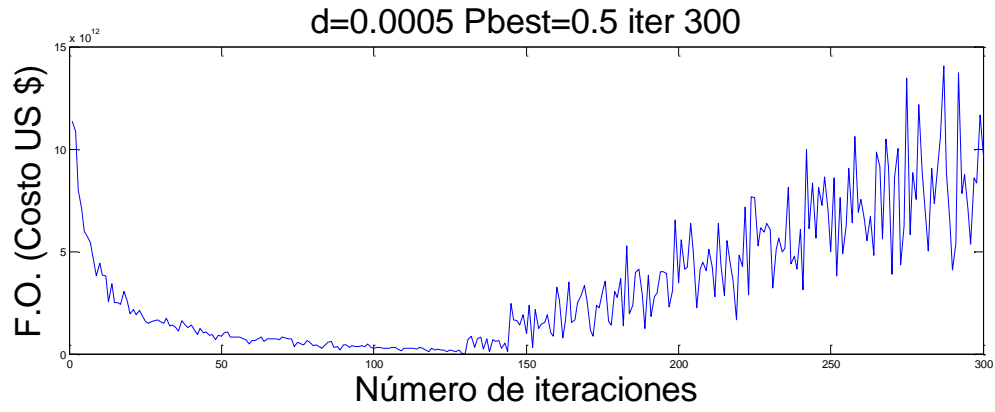
Fuente: Figura obtenida de MATLAB.

Figura 16 Presiones del algoritmo MMSH en cada nodo con un $d=0.005$ y $P_{best}=0.5$



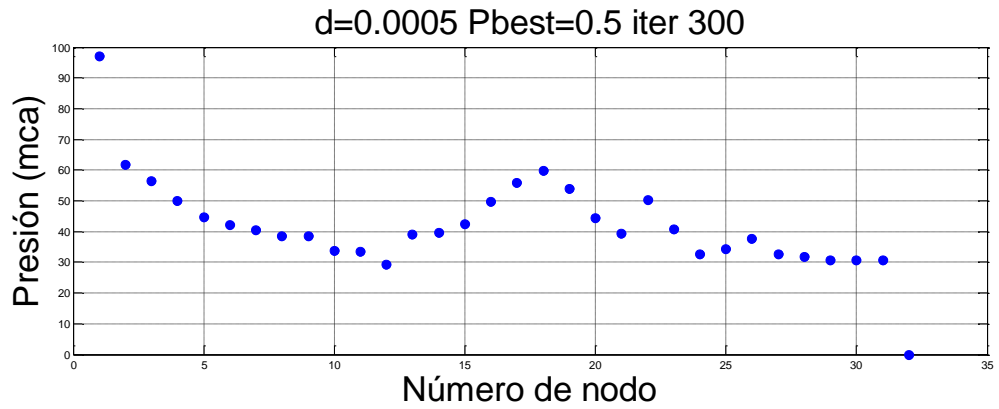
Fuente: Figura obtenida de MATLAB.

Figura 17 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.0005$ y $P_{best}=0.5$



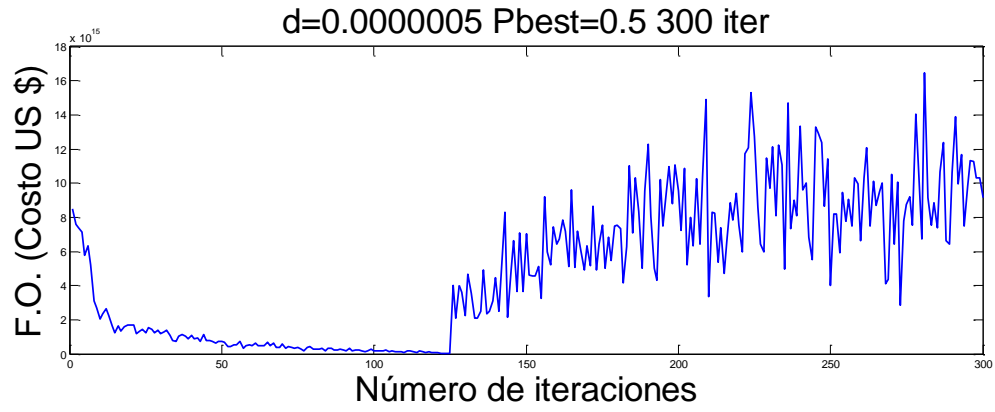
Fuente: Figura obtenida de MATLAB

Figura 18 Presiones del algoritmo MMSH en cada nodo con un $d=0.0005$ y $P_{best}=0.5$



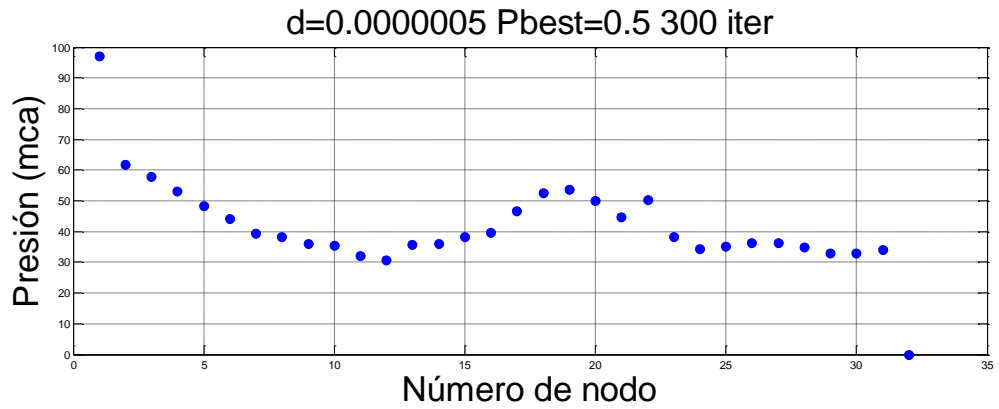
Fuente: Figura obtenida de MATLAB.

Figura 19 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.0000005$ y $P_{best}=0.5$



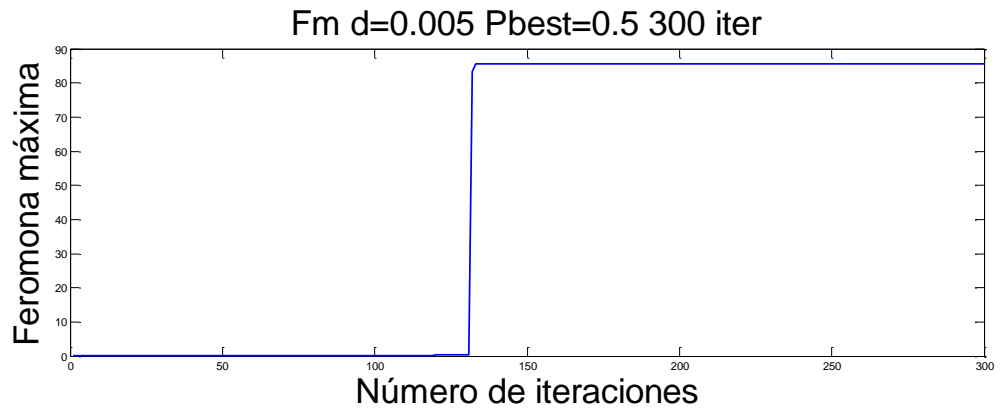
Fuente: Figura obtenida de MATLAB.

Figura 20 Presiones del algoritmo MMSH en cada nodo con un $d=0.0000005$ y $P_{best}=0.5$



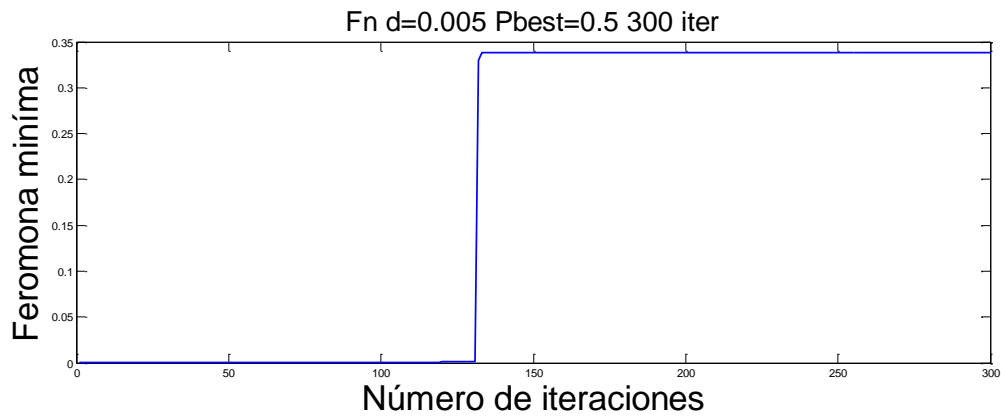
Fuente: Figura obtenida de MATLAB.

Figura 21 Comportamiento de la feromona máxima en el algoritmo MMHS.



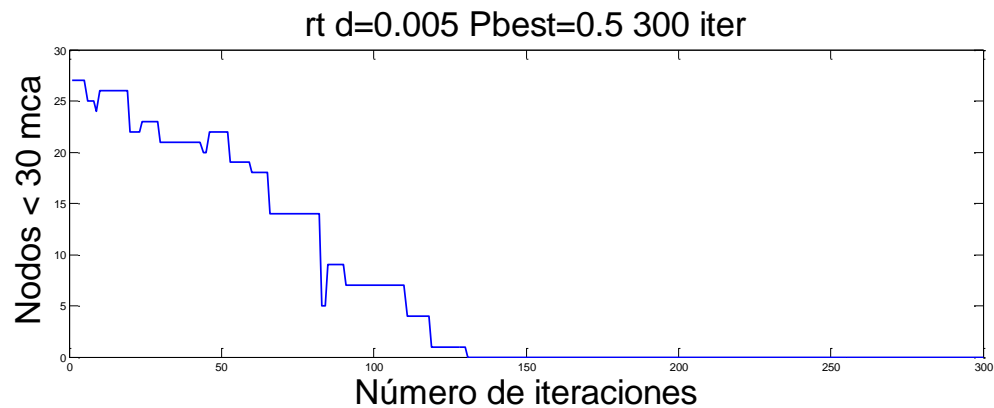
Fuente: Figura obtenida de MATLAB.

Figura 22 . Comportamiento de la feromona mínima en el algoritmo MMHS.



Fuente: Figura obtenida de MATLAB.

Figura 23 Número de nodos que cumplen la restricción en la mejor solución vs iteraciones.



Fuente: Figura obtenida de MATLAB.

5.1.3.2 Resultados MMSH con la actualización de la feromona de la mejor solución global

Todos los casos se corrieron con un valor de frecuencia de actualización de feromona para la mejor solución global igual a 5 y 200 iteraciones, tomando valores de d igual a 0.05, 0.005 y 0.0005.

Para los tres valores de d utilizados el algoritmo encontró soluciones factibles ver tablas (26), (27) y (28).

La mejor solución se encontró con d igual a 0.005, obteniendo una solución con costo igual a \$6, 411,316.18 ver tabla (27).

Tabla 26 Resultado de la variación del parámetro $d=0.05$ con la actualización de la feromona de la mejor solución global.

d=0.05 y Pbest=0.5 200 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	60.0527	ok	18	6
3	61.6711	ok	2	6	20	54.8999	ok	19	6
4	52.7581	ok	3	5	21	45.5511	ok	20	6
5	50.08	ok	4	6	22	44.2792	ok	21	3
6	47.5306	ok	5	6	23	43.9356	ok	22	2
7	47.0397	ok	6	6	24	36.1504	ok	23	5
8	42.6375	ok	7	4	25	34.4614	ok	24	3
9	41.8	ok	8	5	26	38.2906	ok	25	3
10	41.4409	ok	9	5	27	38.6015	ok	26	3
11	39.8814	ok	10	5	28	33.7769	ok	27	5
12	31.5488	ok	11	3	29	32.9364	ok	28	3
13	31.1993	ok	12	6	30	30.4899	ok	29	2
14	43.817	ok	13	4	31	30.7554	ok	30	4
15	46.5918	ok	14	4	32	30.7653	ok	31	1
16	46.9099	ok	15	6	deposito	0		32	1
17	52.2114	ok	16	6				33	6
18	56.8676	ok	17	6				34	3

costo total US \$7,467,114.69 / Tiempo de MMSH 1.5282(horas)

Tabla 27 Resultado de la variación del parámetro $d=0.005$ con la actualización de la feromona de la mejor solución global.

d=0.005 y Pbest=0.5 200 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	54.0566	ok	18	5
3	61.6711	ok	2	6	20	52.4124	ok	19	4
4	58.0298	ok	3	6	21	43.0636	ok	20	6
5	53.5372	ok	4	6	22	37.8989	ok	21	3
6	48.939	ok	5	6	23	47.8057	ok	22	1
7	47.9177	ok	6	6	24	38.4109	ok	23	6
8	43.4987	ok	7	5	25	33.5691	ok	24	4
9	40.211	ok	8	5	26	31.7544	ok	25	4
10	37.9964	ok	9	5	27	34.5504	ok	26	2
11	36.4369	ok	10	5	28	38.1575	ok	27	1
12	35.2807	ok	11	5	29	32.5673	ok	28	2
13	31.0727	ok	12	4	30	30.6797	ok	29	2
14	37.306	ok	13	1	31	31.0077	ok	30	2
15	38.5922	ok	14	2	32	31.9898	ok	31	1
16	39.6974	ok	15	3	deposito	0		32	1
17	43.7994	ok	16	5				33	2
18	49.0566	ok	17	5				34	4

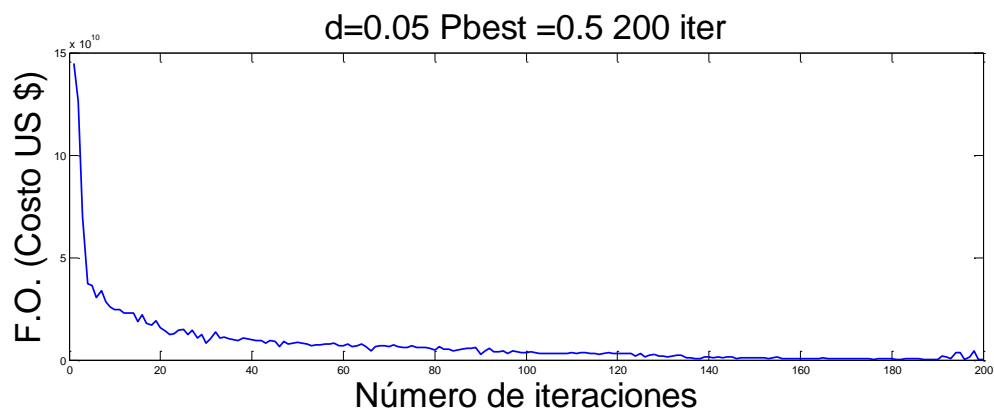
costo total US \$6,411,316.18 / Tiempo de MMSH 1.6256(horas)

Tabla 28 Resultado de la variación del parámetro $d=0.005$ con la actualización de la feromona de la mejor solución global.

d=0.0005 y Pbest=0.5 200 iteraciones									
nodo	Presión (m)	>30 mca	tubería	# tubería	nodo	Presión (m)	>30 mca	tubería	# tubería
2	97.1408	ok	1	6	19	55.8584	ok	18	5
3	61.6711	ok	2	6	20	52.5844	ok	19	4
4	57.3853	ok	3	6	21	43.2357	ok	20	6
5	52.0819	ok	4	6	22	38.0709	ok	21	3
6	46.5506	ok	5	6	23	48.1154	ok	22	1
7	45.279	ok	6	6	24	38.5009	ok	23	6
8	39.3577	ok	7	5	25	36.8115	ok	24	4
9	38.2213	ok	8	6	26	31.498	ok	25	5
10	34.92	ok	9	5	27	32.3982	ok	26	1
11	33.3606	ok	10	5	28	45.5973	ok	27	2
12	32.2043	ok	11	5	29	30.6764	ok	28	4
13	30.7852	ok	12	5	30	30.4585	ok	29	3
14	33.3236	ok	13	3	31	30.6282	ok	30	1
15	32.7363	ok	14	1	32	32.2907	ok	31	1
16	33.2305	ok	15	1	deposito	0		32	2
17	48.6156	ok	16	3				33	2
18	52.0576	ok	17	5				34	3

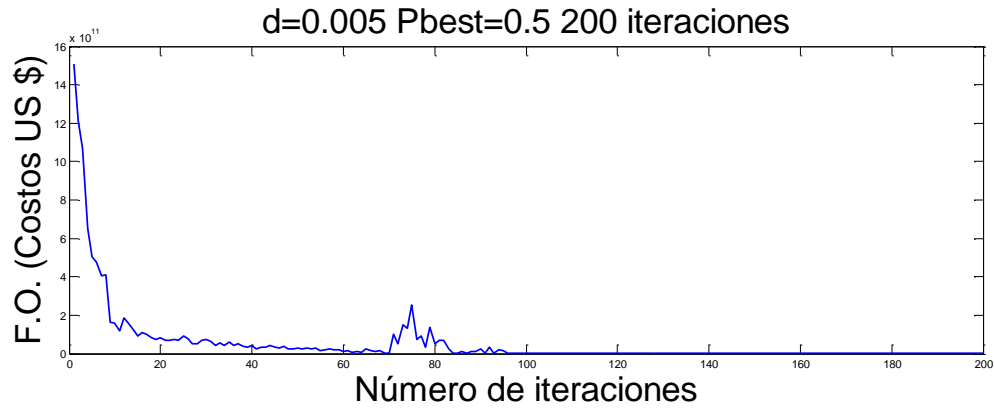
costo total US \$6,514,508.18 / Tiempo de MMSH 1.5047 (horas)

Figura 24 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.05$ $P_{best}=0.5$ y una frecuencia de la feromona de la mejor solución



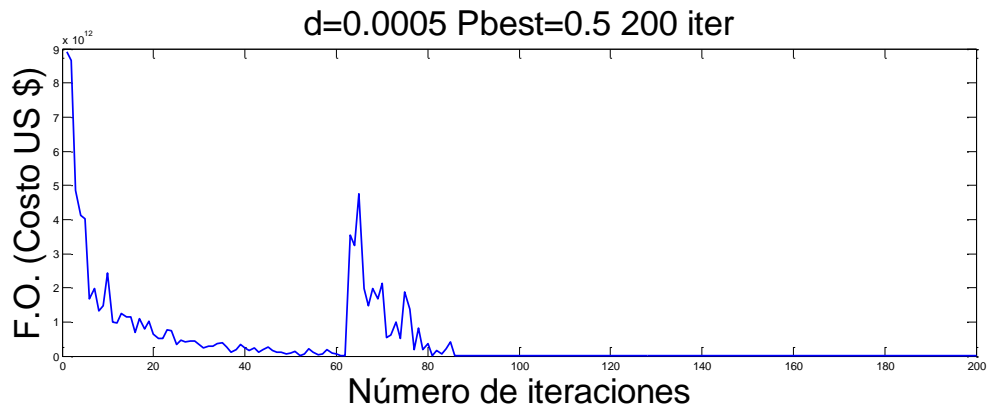
Fuente: Figura obtenida de MATLAB

Figura 25 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.005$ $P_{best}=0.5$ y una frecuencia de la feromona de la mejor solución global=5



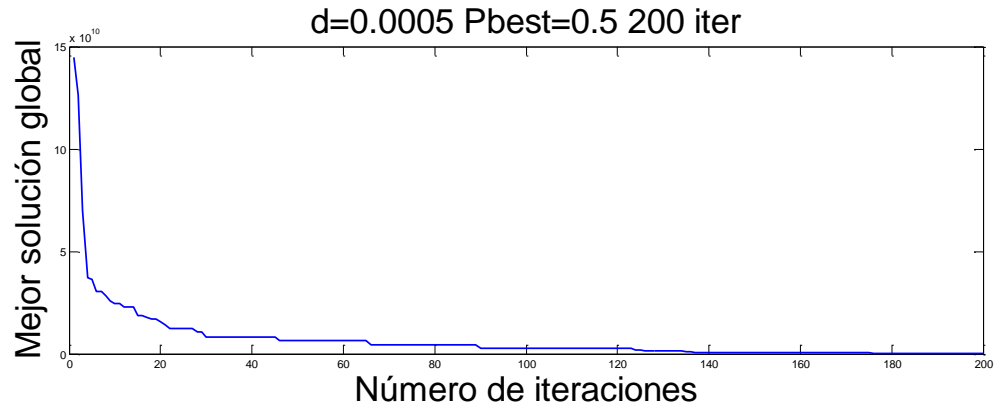
Fuente: Figura obtenida de MATLAB.

Figura 26 Comportamiento del algoritmo MMSH (Costo vs número de iteraciones) con un $d=0.0005$ $P_{best}=0.5$ y una frecuencia de la feromona de la mejor solución global=5



Fuente: Figura obtenida de MATLAB.

Figura 27 Comportamiento del algoritmo MMSH (Mejor solución global vs número de iteraciones) con un $d=0.0005$ $P_{best}=0.5$



Fuente: Figura obtenida de MATLAB.

5.1.4 ANALISIS DE RESULTADOS

Debido a las restricciones impuestas en el problema, la presión en todas las tuberías deben ser mayor a 30 mca, el campo de soluciones factibles se reducen a una región muy pequeña comparada con el campo de exploración del total [1]. Este comportamiento se puede observar en la figura (7) en donde se evidencia que después de 300 iteraciones el comportamiento del algoritmo aún no presenta ninguna tendencia, y además en la mejor solución que ha encontrado hasta el momento solo 7 de los 32 nodos cumplen con la restricción (ver tabla (17)).

Los valores de las primeras iteraciones en las figuras de Costo (US \$) vs Número de iteraciones tienen costos por el orden de 10^{12} debido al aumento de penalización agregado al costo real del diseño es decir los costos teóricos

exceden el mayor costo que puede adquirir la red de Hanoi (si todas las tuberías tuvieran un diámetro de 1016 mm).

Cuando el algoritmo MMSH solo tiene en cuenta la actualización de la feromona de la mejor hormiga de cada iteración este direcciona la búsqueda hacia soluciones que cumplan con la restricción, pero una vez encuentra una solución factible su comportamiento se convierte en errático. Esto se debe a que los precios de las soluciones no factibles están afectados por un valor de penalización que aumenta el costo real de la misma, y como se puede ver en las ecuaciones (8) y (9) que los valores de feromona máxima y feromona mínima son inversamente proporcionales al costo de la solución, entonces cuando el algoritmo encuentra una solución factible el precio de esta es mucho menor a una solución que está afectada a la penalización provocando que los valores de feromonas máximo y mínimo sufran un gran aumento en su valor ver figuras (21),(22) y (23). En la iteración siguiente a la cual el algoritmo encuentra una solución factible todos los rastros de feromona toman el valor de la feromona mínima provocando que las probabilidades vuelvan a ser iguales a los valores iniciales perdiendo así todo lo aprendido.

Al agregar la actualización periódica de feromona de la mejor solución global, el algoritmo sufre una pequeña desestabilización al momento de encontrar la primera solución factible, pero gracias a la actualización antes mencionada el algoritmo estabiliza su comportamiento luego de algunas iteraciones ver figuras (24), (25) y (26).

La actualización periódica de la feromona de la mejor solución global no solo estabiliza el sistema de la feromona antes mencionada, sino que además hace que la curva de aprendizaje sea más pronunciada (el algoritmo aprende más rápido) comparar figuras (18), (19), (20) con (24), (25), (26).

Un menor valor de d provoca una desestabilización más grande al encontrar la primera solución factible, figuras (24), (25) y (26).

Los algoritmos SH y MMSH tuvieron el mismo gasto computacional (tardaron el mismo tiempo en realizar el mismo número de iteraciones) a pesar de la enorme diferencia en su velocidad de aprendizaje.

El valor de la mejor solución encontrado en este trabajo fue mayor al encontrado en [1]. Esto se debe a que en la bibliografía buscada no se encontraron parámetros para el valor de d ni de frecuencia de actualización de la feromona de la mejor solución global, y teniendo en cuenta que el objetivo de este trabajo fue evaluar el comportamiento del algoritmo y no optimizar sus parámetros, mediante un análisis superfluo, se obtuvieron valores que no necesariamente entregaban la mejor solución.

6. CONCLUSIONES

Los algoritmos OCH aplicados a las redes de distribución de agua para la optimización del diseño de tuberías presentan diferentes desempeños, el caso de SH no cumple con las restricciones que se imponen en el problema por lo tanto no es factible la solución que se obtiene al finalizar las iteraciones. El algoritmo MMSH muestra un mayor rendimiento para esta clase de optimización generando soluciones factibles y cercanas a los costos conocidos en la literatura.

MATLAB es una útil herramienta para desarrollar este tipo de algoritmos ya que tiene un lenguaje de programación sencillo y cuenta con funciones que facilitan el acceso a otros programas como en este caso el código fuente de EPANET. La duración del número de iteraciones era aproximadamente de 2.3 horas para 300 iteraciones y 1.45 para 200 iteraciones, estos valores eran iguales para SH como para MMSH.

Los diámetros obtenidos por MMSH para la red de Hanoi cumplieron con las restricciones de diseño y se acercaron a los valores encontrados por la literatura, no se encontraron valores menores debido a la falta de datos que afectan la penalización como lo es d y la frecuencia de feromona de la mejor solución global.

El costo obtenido por el algoritmo OCH-EPANET fue de US \$6,411 millones mayor al costo reportado en [1] que es de US \$6,182 millones. A continuación se presentan los diámetros de tubería para cada diseño tabla 29.

Tabla 29 Comparación del mínimo costo encontrado por OCH-EPANET y MMSH en [1]

tubería	MMSH [1]	OCH-EPANET
	ϕ (mm)	ϕ (mm)
1	1016	1016
2	1016	1016
3	1016	1016
4	1016	1016
5	1016	1016
6	1016	1016
7	1016	762
8	1016	762
9	1016	762
10	762	762
11	609.6	762
12	609.6	609.6
13	508	304.8
14	304.8	406.4
15	304.8	508
16	304.8	762
17	508	762
18	609.6	762
19	508	609.6
20	1016	1016
21	508	508
22	304.8	304.8
23	1016	1016
24	762	609.6
25	762	609.6
26	508	406.4
27	304.8	304.8
28	304.8	406.4
29	406.4	406.4
30	304.8	406.4
31	304.8	304.8
32	304.8	304.8
33	406.4	406.4
34	508	609.6
Costo de la RDA's (US M\$)	6,182	6, 411

El algoritmo MMSH mostro un buen desempeño en la búsqueda de soluciones factibles disminuyendo el campo de búsqueda solo para las mejores soluciones. Como MMHS es solo uno de los derivados de los algoritmos OCH, una futura investigación debería centrarse en los demás derivados de la OCH para evaluar su desempeño en la optimización de las RDA´s.

Los parámetros que obtuvieron la mejor solución ver tabla (29) con el algoritmo MMSH se muestran en la siguiente tabla 30.

Tabla 30 Parámetros de la mejor solución encontrada del algoritmo MMSH

Parámetros de la mejor solución del algoritmo MMSH		
m	(número de hormigas)	80
α	(factor de escalado de feromona)	1.0
β	(factor de visibilidad)	0.5
ρ	(tasa de evaporación)	0.98
τ_o	(feromona inicial)	50
P_{best}	(probabilidad que S^{gb} sea seleccionado por una hormiga)	0.5
$f(S^{gb})$	(frecuencia de la feromona de la mejor solución global)	5
Q	(factor de adición de la feromona)	1.1×10^7
d	(coeficiente de feromona suavizado)	0
d	(déficit de presión seleccionado por el usuario)	0.005
ITER	(número de iteraciones)	200

El motor hidráulico más eficiente para este tipo de algoritmo es el código fuente de EPANET, ya que a diferencia de otros métodos como por el ejemplo el Método del gradiente o el método de Hardy-Croos permiten evaluar rápidamente la hidráulica de la red.

El algoritmo OCH-EPANET es una herramienta útil para la optimización de diseños en la RDA's. A continuación se presentan un paralelo entre OCH-EPANET y un diseño tradicional en EPANET 2.0 (ver tablas (14), (29) y (30)):

Tabla 31 Comparación de OCH-EPANET y Diseño tradicional en EPANET 2.0.

ALGORITMO OCH (MMSH)	DISEÑO TRADICIONAL EPANET 2.0
<p>Asigna los diámetros de manera automática realizando una búsqueda exhaustiva en la región factible de la solución permitiendo que se utilicen un número amplio de iteraciones.</p>	<p>Los diámetros se asignan manualmente por lo tanto las iteraciones no pueden realizarse rápidamente y la búsqueda de soluciones factibles se disminuye.</p>
<p>Al buscar soluciones no solo tiene en cuenta que cumplan con las restricciones hidráulicas además considera el costo del diseño para encontrar soluciones óptimas.</p>	<p>No contiene una herramienta que considere el costo total de la red, este valor solo se puede evaluar al finalizar el diseño.</p>
<p>Después de colocar las condiciones</p>	<p>AL ser un programa semiautomático el</p>

<p>físicas de la red de distribución de agua y sus características de consumo el algoritmo OCH tarda en encontrar un diseño óptimo en un tiempo razonable dependiendo del número de iteraciones seleccionado (200 iteraciones aproximadamente 1.45 horas) tiempo que puede ser utilizado por el diseñador en otras actividades relacionadas con la red.</p> <p>Para ejecutar el algoritmo se necesita un ordenador con un procesador de alto rendimiento.</p>	<p>diseñador debe dedicar su tiempo frente al computador hasta encontrar los diámetros que se acerquen más a una solución factible.</p> <p>Se puede ejecutar en la gran mayoría de ordenadores comerciales.</p>
---	---

La red de Hanoi es una red estándar conocida en la literatura para hacer comparaciones de los resultados obtenidos por diferentes métodos. A continuación se presenta una tabla con resultados encontrados por otros autores.

Tabla 32 Comparación de las soluciones para la red de Hanoi.

Autores	año	Costo US\$ M	Factible
Savic & Walters GA2	1997	6.195	Si
Wu et al. FmGA1	2001	6.182	si
AOC (Uniandes).	2005	6.161	si
Zecchin et al. MMAS	2005	6.134	si
OCH-EPANET (UIS)	2013	6.411	si

7. RECOMENDACIONES Y TRABAJOS FUTUROS

Optimizar el valor de déficit de presión (d), en [1] se hace referencia al valor de d como un valor estipulado por el usuario sin embargo al ser un parámetro que influyen considerablemente al algoritmo se debería enfatizar en el mejor valor para el caso de estudio.

Evaluar el comportamiento de la frecuencia de feromona de la mejor solución global, como este valor no fue mencionado en la literatura consultada es recomendable probar diferentes valores para estipular su influencia en el comportamiento del algoritmo.

Se necesita un ordenador con un procesador de alto rendimiento ya que en algunos computadores no se puede ejecutar, en este caso se utilizó uno con las siguientes características Intel® Core™ i5-3330 Processor (6M Cache, up to 3.20 GHz) y dos núcleos. El gasto computacional debe evaluarse en equipos con una mayor capacidad de procesador y memoria para comprobar si el tiempo de las iteraciones disminuye considerablemente.

En este trabajo se realizó únicamente la implementación de los algoritmos SH y MMSH al diseño de RDA's, se debería evaluar el desempeño de otros derivados de la OCH para comparar su rendimiento y la optimización en los diseños de la red.

En una futura investigación debería incluirse en el esquema del algoritmo OCH-EPANET el método de superficies optima que podría reducir el tiempo de búsqueda del algoritmo.

8. BIBLIOGRAFÍA

[1] Aaron C, Zecchin. Angus R, Simpson. Holger R, Maier. Michael Leonard. Andrew J, Roberts. Matthew J, Berrisford. Application of two ant colony optimization algorithms to water distribution system optimization. Australia Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, the University of Adelaide, Adelaide, S.A., 2005, Australia. Optimatics Pty. Ltd., Parkside, S.A., 5063.

[2] H, R, Maier. A, R, Simpson. A, C, Zecchin. W, K, Foong. K, Y, Phang. H, Y, Seah. C, L, Tan. Ant Colony Optimization for the design of water distribution systems. Journal of Water Resources Planning and Management, ASCE 129 (3) (2003) 200–209.

[3] Carlos A, Coy Calixto. Nelson O, Neira. Implementación en hidroinformática de un método de optimización matemática basada en la colonia de hormigas: Optimización Colonia de Hormigas. Bogotá, Colombia, 2005, 24 p. Trabajo de grado (Ingeniería civil). Pontificia Universidad Javeriana. Facultad de Ingeniería civil. Área de Hidrología.

[4] Evelyn Menéndez Alonso. Metaheurística de Optimización mediante Colonias de Hormigas y Aplicaciones. Madrid, España, evelynma@uclv.edu.cu.

[5] M, Dorigo. G, Di Caro. L, M, Gambardella. Ant algorithms for discrete optimization. Bruselas, Belgicas, 1999. Artificial Life 5 (2) 137–172.

[6] M,H, Afshar. Partially constrained ant colony optimization algorithm for the solution of constrained optimization problems: Application to storm water network

design. Tehran, Irán, 2006. Civil Engineering, Irán University of Science and Technology, Narmak, Tehran 16844, Irán Advances in Water Resources DOI:10.1016/j.advwatres.2006.08.004.

[7] T. Stützle. H, H, Hoos. MAX-MIN ant system. Future Generation Computer Systems. IRIDIA, Université Libre de Bruxelles, Avenue Franklin Roosevelt 50, CP 194/6, 1050 Brussels, Belgium Computer Science Department, University of British Columbia, 2366 Main Mall, Vancouver, BC, Canada V6T 1Z4 16 (2000) 889–914.

[8] A, C, Zecchin. A, R, Simpson. H, R, Maier. J, B, Nixon. Parametric study for an ant algorithm applied to water distribution system optimization. Australia, 2005. IEEE Transactions on Evolutionary Computation p 175–191.

[9] Yates, DF. Templeman, A.B, Boffey T.B. The computational complexity of the problem of determining least capital cost designs for water supply networks. Liverpool, UK.1984. Department of Statistics and Computational Mathematics, University of Liverpool 7(2), pp.142-155.

[10] G, V Fernández. J, G Saldarriaga. Algoritmos de optimización combinatoria para las redes de distribución de Agua potable. Bogotá: Revista de ingeniería Universidad de los Andes, 2005.

[11] J, G Saldarriaga. Hidráulica de tuberías Abastecimiento de agua redes y riego. Bogotá, Colombia: Universidad de Los Andes (Ediciones Uniandes) Alfaomega, 2007. P 415-443.

[12] D.A. Savic, G.A. Walters, Genetic algorithms for least-cost design of water distribution networks. UK: Journal of Water Resources Planning and Management, ASCE 123 (2) (1997) 67–77.

[13] Z. Y, Wu. P, F, Boulos. C, H, Orr. J, J, Ro. Using genetic algorithms to rehabilitate distribution system. Journal for American Water Works Association (2001) 74–85

[14] M. Cunha, J. Sousa, Water distribution network design optimization: Simulated Annealing Approach, Journal of Water Resources Planning and Management, ASCE 125 (4) (1999) 215–221.

[15] D.A. Savic, G.A. Walters, Genetic algorithms for least-cost design of water distribution networks, Journal of Water Resources Planning and Management, ASCE 123 (2) (1997) 67–77.

[16] Z.Y. Wu, P.F. Boulos, C.H. Orr, J.J. Ro, Using genetic algorithms to rehabilitate distribution system, Journal for American Water Works Association (2001) 74–85.

ANEXO A

CÓDIGO OCH-EPANET PARA EL PROBLEMA DEL AGENTE VIAJERO

El código del programa está codificado en el lenguaje MATLAB. Está compuesto por el archivo principal ACO y un archivo que contiene las coordenadas de las ciudades (ver tabla (1)).

FUNCIÓN VA (COORDENADAS DE LAS CIUDADES)

```
function T=VA;

Coor(1,1)=-80;
Coor(1,2)=60;
Coor(2,1)=-46;
Coor(2,2)=88.79;
Coor(3,1)=30;
Coor(3,2)=95.39;
Coor(4,1)=90;
Coor(4,2)=-43.6;
Coor(5,1)=-8;
Coor(5,2)=-99.68;
Coor(6,1)=-83;
Coor(6,2)=-55.78;
Coor;

for i=1:6
    for j=1:6
        dist(i,j)=sqrt(((Coor(i,1)-Coor(j,1))^2)+(Coor(i,2)-Coor(j,2))^2);
    end
end

T=dist;
```

ARCHIVO ACO

```
clc;
clear all;
format compact;
disp('*****')
disp('*****')
disp('***** ACO *****')
disp('*****')
disp('*****')
tic;

%Datos de entrada

Iter=100; %Iteraciones que realiza el algoritmo
m=6; %Número de hormigas
Alfa=1;
Beta=2;
Gamma=0.5;
Q=10;
Fi=.03; %Valor inicial de la feromona
Tr=VA;%Matriz Distancias recorridas por las hormigas
Tra=Tr;
for j=1:length(Tr)
for i=j:length(Tr)-1
    Tr(i,j)=Tr(i+1,j);
end
end
Tr(length(Tr),:)=[];
for i=1:length(Tr)-1
    for j=1:length(Tr)
        Vis(i,j)=1/(Tr(i,j));%Matriz de visibilidad
        MF(i,j)=Fi;%Matriz feromona inicial
    end
end
end

for kk=1:Iter

for i=1:length(Tr)
    for j=1:length(Tr)-1
        MM(j,i)=(MF(j,i)^Alfa)*((Vis(j,i))^Beta);
    end
    S=sum(MM);
    for k=1:length(Tr)-1
        Prob(k,i)= (MM(k,i))/(S(i));%Probabilidad de la ruta
    end
end

for i=1:m
    Pi(i,1)=fix(1+rand*(length(Tr)-1));%Posición inicial para cada
hormiga
    Ve=1:length(Tr);%Vector con número de ciudades a visitar
    LT(i,1)=Pi(i,1);%Primer término de la lista tabú
```

```

    Ve(LT(i,1))=[];%Elimina del vector de ciudades a visitar la que
agregó a la lista tabú
    Vec=Prob(:,Pi(i,1));%Vector de probabilidad de viajar las ciudades
desde la ciudad de origen
    for j=1:length(Tr)-1
        [xm,im]=max(Vec);% Escoge la ciudad con mayor probabilidad.
        LT(i,j+1)=Ve(im);%Agrega la ciudad escogida a la lista tabú
        Dt(i,j)=Tra(LT(i,j),LT(i,j+1));%Agrega la distancia entre la 2
ciudades
        Vec=Prob(:,Ve(im));%Nuevo vector de probabilidad (De la nueva ciudad
de partida)
        Ve=1:length(Tr);%Vector con número de ciudades a visitar
        Ve(LT(i,j+1))=[];%Elimina del vector de ciudades a visitar la que
agregó a la lista tabú
        for t=1:j
            for k=1:length(Ve)
                if Ve(k)==LT(i,t)%Compara las ciudades a visitar con las
ciudades en LT
                    Ve(k)=[];
                    Vec(k)=[];
                    break
                end
            end
        end
        end

        Dt(i,length(Tr))=Tra(LT(i,1),LT(i,length(Tr)));%Distancia del último
tramo
        Dtr=sum(Dt');%Distancia total recorrida por la hormiga
    end
MF=Gamma*MF;%Evaporación de la feromona
MFA=zeros(length(Tr));%Matriz de delta de feromona
for i=1:m
    for j=1:length(Tr)
        AFe(i,j)=Q/Dtr(i);
        if j==length(Tr)
            MFA(LT(i,j),LT(i,1))=MFA(LT(i,j),LT(i,1))+AFe(i,j);
            MFA(LT(i,1),LT(i,j))=MFA(LT(i,1),LT(1,j))+AFe(i,j);
        else
            MFA(LT(i,j),LT(i,j+1))=MFA(LT(i,j),LT(i,j+1))+AFe(i,j);
            MFA(LT(i,j),LT(i,j+1))=MFA(LT(i,j),LT(i,j+1))+AFe(i,j);
        end
    end
end
for j=1:length(MFA)
for i=j:length(MFA)-1
    MFA(i,j)=MFA(i+1,j);
end
end
MFA(length(MFA),:)=[];
MF=MF+MFA;%Matriz de feromona modificada

end
    LT

```

ANEXO B

CÓDIGO OCH-EPANET PARA LAS RDA's

El código del programa está codificado en el lenguaje MATLAB. Está compuesto por el archivo principal MN además de los archivos VA, DEMO y un archivo inp MybigFat que contiene las características de la red hidráulica (Todos los archivos incluyendo el Toolkit de EPANET 2.0 deben guardarse en una misma carpeta).

ARCHIVO MN

Contiene la estructura del algoritmo Máximos y Mínimos Sistema de Hormigas aplicado a la optimización de RDA's.

```
clc;
clear all;
format compact;
disp('*****')
disp('*****')
disp('***** OCH-EPANET *****')
disp('*****')
disp('*****')
tic;
NT=34; %Número de tramos de la red
%Datos de entrada
Iter=200; %Iteraciones que realiza el algoritmo
m=80; %Número de hormigas
Alfa=1.0;
Beta=0.5;
Gamma=0.98;
Q=1.1*10^7;
Fi=50; %Valor inicial de la feromona
Tr=VA;%Matriz Distancias recorridas por las hormigas
```

```

Tra=Tr;
Diam=[304.8 406.4 508 609.6 762 1016];
Vis=1./Tr;%Matriz de visibilidad
MF=ones(size(Tra,1),size(Tra,2))*Fi;%Matriz feromona inicial
Pbest=0.5;
n=34;
NOavg=6;
d=0.05;
%CMS=10^9999999999999999;
AA=1;
Sgb=1000000000000000000;
for kk=1:Iter
    MM=(MF.^Alfa).*Vis.^Beta;
    S=sum(MM);
    for i=1:size(Tr,2)
        for j=1:size(Tr,1)
            Prob(j,i)=MM(j,i)/S(i);
        end
    end
    for i=1:m
        for j=1:NT
            bb=Prob(1,j);
            pp=rand;
            for g=1:size(Prob,1)
                if pp<=bb
                    Pi(i,j)=g;
                    break
                else bb=bb+Prob(1+g,j);
                end
            end
            LT(i,j)=Pi(i,j);%Primer término de la lista tabú
            Dt(i,j)=Tra(Pi(i,j),j);
            Dia(i,j)=Diam(LT(i,j));
        end
    end
    Di=Dia(i,:);
    Di=Di';
    Hc=DEMO(Di); %llamada al motor hidráulico (Código fuente EPANET 2.0)
    Hcc(:,i)=Hc;
    cc(i)=0;
    for k=1:size(Hcc,1)-1
        if Hcc(k,i)<=30;
            cc(i)=cc(i)+1;
        end
    end
end
end
Dtr=sum(Dt');%Distancia total recorrida por la hormiga
Hcc(32,:)=[];
for tt=1:m
    if cc(tt)>=1
        Dtr(tt)=Dtr(tt)+abs(min(Hcc(:,tt))-30)*((sum(Tra(6,:))-sum(Tra(1,:)))/d); %Ecuación de costo de la red con penalización
    end
end
Hcc(32,:)=0;

```

```

Fmax=(1/(1-Gamma))*(Q/Sgb); %Feromona máxima
t=1;
[ix,mx]=min(Dtr);
Sib=Dtr(mx);
if Sib<=Sgb
    Sgb=Sib;
    MS=LT(mx,:);
    rrr=cc(mx);
    Hhc=Hcc(:,mx);
end
Fmin=(Fmax*(1-(Pbest)^(1/n)))/(((NOavg-1)*(Pbest)^(1/n))); %Feromona
mínima
MF=Gamma*MF;%Evaporación de la feromona
MFAi=0*Tr;%Matriz de delta de feromona
MFAg=MFAi;
AF(kk)=Q/Dtr(mx);
    for j=1:size(Tr,2)
        AFei(mx,j)=Q/Dtr(mx);
        MFAi(LT(mx,j),j)=MFAi(LT(mx,j),j)+AFei(mx,j);
    end

    u=kk/5;
    ub=fix(u);
    if u==ub
        for j=1:size(Tr,2)
            AFeg(j)=Q/Sgb;
            MFAg(MS(j),j)=MFAg(MS(j),j)+AFeg(j);
        end
    end
MFA=MFAi+MFAg; %Actualización de la Feromona
MF=MF+MFA;%Matriz de feromona modificada
for i=1:size(MF,1)
    for j=1:size(MF,2)
        if MF(i,j)>=Fmax
            MF(i,j)=Fmax;
        else
            if MF(i,j)<=Fmin
                MF(i,j)=Fmin;
            end
        end
    end
end
end
AFei=[];
AFeg=[];
msi(kk)=Sgb;
cm(kk)=ix;
rt(kk)=rrr;
Fm(kk)=Fmax;
Fn(kk)=Fmin;
Probkk(kk)=Prob(6,1);
end
MS %Mejor solución
plot(cm) %Grafica de Costo (US$) vs Numero de iteraciones
toc

```

ARCHIVO VA

Este archivo contiene los precios de cada diámetros de tubería y las longitudes de cada tramo de la red. Se utiliza como una función que se llama desde el archivo de MN.

```
function T=VA
```

```
Pre(1)=45.726;  
Pre(2)=70.400;  
Pre(3)=98.378;  
Pre(4)=129.333;  
Pre(5)=180.748;  
Pre(6)=278.28;
```

```
Long = [100 1350 900 1150 1450 450 850 850 800 950 1200 3500 800 500 550  
2730 1750 800 400 220 1500 500 2650 1230 1300 850 300 750 1500 2000 1600  
150 860 950];
```

```
for i=1:length(Pre)  
    for j=1:length(Long)  
        dist(i,j)=Pre(i)*Long(j);  
    end  
end
```

```
T=dist;
```

ARCHIVO DEMO

En este archivo se presenta el motor hidráulico (Código fuente de EPANET). Antes de utilizar el código se introducen los datos de la red. A demás se debe crear un archivo inp llamado “MybigFat inp” que contenga la información física de las tuberías y dinámicas de los nodos.

```
function H=DEMO(Di)
clc
% clear
delete('exptable.inp')

% open a file for writing
% en la primera segunda fila están el nodo inicial y el nodo final de
cada tramo, en la tercera se encuentran las longitudes en metros y en la
cuarta están los diámetros de las tuberías en milímetros.
RED=[1, 2, 100, 1016;
     2, 3, 1350, 1016;
     3, 4, 900, 1016,
     4, 5, 1150, 1016;
     5, 6, 1450, 1016;
     6, 7, 450, 1016;
     7, 8, 850, 1016;
     8, 9, 850, 1016;
     9, 10, 800, 1016;
     10, 11, 950, 762;
     11, 12, 1200, 609.6;
     12, 13, 3500, 609.6;
     10, 14, 800, 508;
     14, 15, 500, 406.4;
     15, 16, 550, 406.4;
     16, 17, 2730, 406.4;
     17, 18, 1750, 508;
     18, 19, 800, 609.6;
     19, 3, 400, 508;
     3, 20, 2200, 1016;
     20, 21, 1500, 508;
     21, 22, 500, 304.8;
     20, 23, 2650, 1016;
     23, 24, 1230, 762;
     24, 25, 1300, 762;
     25, 26, 850, 508;
     26, 27, 300, 304.8;
```

```

    27, 16, 750, 304.8;
    23, 28, 1500, 406.4;
    28, 29, 2000, 304.8;
    29, 30, 1600, 304.8;
    30, 31, 150, 304.8;
    31, 32, 860, 406.4;
    32, 25, 950, 508];

RED(:,4)=Di;
ID=(1:34);
Node1=RED(:,1)';
Node2=RED(:,2)';
Length =RED(:,3)';
Diameter=RED(:,4)';
MinorLoss=zeros(1,34);
Roughness=130.*(ones(1,34));

DataMa=[ID;Node1;Node2;Length;Diameter;Roughness;MinorLoss];
fid = fopen('exptable.inp', 'w');

ff= ';ID      Node1      Node2      Length      Diameter      Roughness
MinorLoss      Status \r\n';
% print a title, followed by a blank line
fprintf(fid,ff);

% print values in column order
% two values appear on each row of the file
myformat = ' %-6d %-6d  %-6d  %-9d  %-16g  %-9d  %-7d
Open      ; \r\n';
fprintf(fid, myformat, DataMa);
fclose(fid);

%To view the file, use the type function:
type exptable.inp;
system('copy 1.txt+exptable.inp+3.txt MyBigFat.inp');

% Example for EPANET_MATLAB_EXT
% Author: Demetrios G. Eliades, eldemet@gmail.com,
% University of Cyprus www.ucy.ac.cy
% KOIOS Centre for Intelligent Systems and Networks www.kios.org.cy
% August 2009

wdsfile='MyBigFat.inp';
pumps={'0'};
tanks={'1'};
epanetloadfile(wdsfile);
NNode=NodeInfoFromInp(wdsfile);
%find number node match with Presion and several variables
for i=1:length(NNode)
NodeNumber(i)=str2double(NNode{i});
end

```

```

TD          =0;    % duration in days
random      =0;    % random demands (1=yes, 0=no)

%Uncertainty
p_roughness =0;    % roughness in all pipes
p_bulk      =0;    % bulk coefficients in all pipes
p_basedemands =0;  % base demands
p_demandpattern =0; % demand pattern
p_wall      =0;    % wall coefficients

lentanks=length(tanks);
lenpumps=length(pumps);

%set hydraulic time step
TS=5; %in minutes
tstep=TS*60;
setdata('EN_HYDSTEP', tstep);

%set report time step
TR=5; %in minutes
treport=TR*60; %
setdata('EN_REPORTSTEP', treport);

%set duration
tduration=TD*24*60*60;
setdata('EN_DURATION', tduration);

%get number of patterns
patcount = getdata('EN_PATCOUNT');

if random==1
    %create set of random demands
    Demands_randomized=2.5*rand(patcount, (tduration/tstep));
    %set demands
    for i4=1:patcount
        setdata('SET_PATTERN', Demands_randomized(i4,:), i4);
    end %i4
end %if

%get base demands and add some white noise
basedemand=getdata('EN_BASEDEMAND');
basedemand_random=p_basedemands*2*(rand(length(basedemand),1) '-
0.5).*basedemand+basedemand;
setdata('EN_BASEDEMAND', basedemand_random);

```

```

%get roughness coefficients and add some white noise
roughness=getdata('EN_ROUGHNESS');
roughness_random=p_roughness*2*(rand(length(roughness),1) '-
0.5).*roughness+roughness;
setdata('EN_ROUGHNESS',roughness_random);

%get bulk coefficients (-0.5) and add some white noise
kbulk=getdata('EN_KBULK');
kbulk_random=p_bulk*2*(rand(length(kbulk),1) '-0.5).*kbulk+kbulk;
setdata('EN_KBULK',kbulk_random);

%get wall coefficients and add some white noise
kwall=getdata('EN_KWALL');
kwall_random=p_wall*2*(rand(length(kwall),1) '-0.5).*kwall+kwall;
setdata('EN_KWALL',kwall_random);

%printouts
fprintf('nodes = %d\n',getdata('EN_NODECOUNT'))
fprintf('links = %d\n',getdata('EN_LINKCOUNT'))
fprintf('tanks = %d\n',lentanks)
fprintf('pumps = %d\n',lenpumps)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% S O L V E   H Y D R A U L I C S
[d_N,tv] = getdata('EN_DEMAND');
[P] = getdata('EN_PRESSURE');
[Q] = getdata('EN_FLOW');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% F I G U R E S
figure(1);
plot(Q); %plot all flows
title('plot all flows');
ylabel('{Lt/s}');
figure(2);
plot(NodeNumber,P,'r+'); %plot all pressures
hold on;
plot(NodeNumber,P,'bo');

hold off;
title('Plot all Pressures');
xlabel('Node Number');
ylabel('{mca}');
grid on;

%close everything
epanetclose();
%clc

```

H=P '
 [TITLE]
 [JUNCTIONS]

ARCHIVO Mybigfit

Este archivo inp contiene las características físicas de la tuberías y dinámicas de cada nodo en la Red hidráulica, la columna de diámetros es la única que cambiara con cada iteración del algoritmo.

;ID	Elev	Demand	Pattern
2	0	247.22	:
3	0	236.11	:
4	0	36.11	:
5	0	201.39	:
6	0	279.17	:
7	0	375	:
8	0	152.78	:
9	0	145.83	:
10	0	145.83	:
11	0	138.89	:
12	0	155.56	:
13	0	261.11	:
14	0	170.83	:
15	0	77.78	:
16	0	86.11	:
17	0	240.28	:
18	0	373.61	:
19	0	16.67	:
20	0	354.17	:
21	0	258.33	:
22	0	134.72	:
23	0	290.28	:
24	0	227.78	:
25	0	47.22	:
26	0	250	:
27	0	102.78	:
28	0	80.56	:
29	0	100	:
30	0	100	:
31	0	29.17	:
32	0	223.61	:

[RESERVOIRS]
 ;ID Head Pattern ;
 1 100 ;

[TANKS]
 ;ID Elevation InitLevel MinLevel MaxLevel Diameter MinVol VolCurve

[PIPES]

;ID	Node1	Node2	Length	Diameter	Roughness	MinorLoss	Status
1	1	2	100	1016	130	0	Open ;
2	2	3	1350	304.8	130	0	Open ;
3	3	4	900	762	130	0	Open ;
4	4	5	1150	1016	130	0	Open ;
5	5	6	1450	1016	130	0	Open ;
6	6	7	450	1016	130	0	Open ;
7	7	8	850	304.8	130	0	Open ;
8	8	9	850	762	130	0	Open ;
9	9	10	800	762	130	0	Open ;
10	10	11	950	609.6	130	0	Open ;
11	11	12	1200	508	130	0	Open ;
12	12	13	3500	1016	130	0	Open ;
13	10	14	800	406.4	130	0	Open ;
14	14	15	500	609.6	130	0	Open ;
15	15	16	550	609.6	130	0	Open ;
16	16	17	2730	1016	130	0	Open ;
17	17	18	1750	1016	130	0	Open ;
18	18	19	800	1016	130	0	Open ;
19	19	3	400	1016	130	0	Open ;
20	3	20	2200	304.8	130	0	Open ;
21	20	21	1500	304.8	130	0	Open ;
22	21	22	500	406.4	130	0	Open ;
23	20	23	2650	762	130	0	Open ;
24	23	24	1230	508	130	0	Open ;
25	24	25	1300	508	130	0	Open ;
26	25	26	850	406.4	130	0	Open ;
27	26	27	300	762	130	0	Open ;
28	27	16	750	508	130	0	Open ;
29	23	28	1500	609.6	130	0	Open ;
30	28	29	2000	609.6	130	0	Open ;
31	29	30	1600	304.8	130	0	Open ;
32	30	31	150	304.8	130	0	Open ;
33	31	32	860	304.8	130	0	Open ;
34	32	25	950	508	130	0	Open ;

[PUMPS]

;ID	Node1	Node2	Parameters
-----	-------	-------	------------

[VALVES]

;ID	Node1	Node2	Diameter	Type	Setting	MinorLoss
-----	-------	-------	----------	------	---------	-----------

[TAGS]

[DEMANDS]

;Junction	Demand	Pattern	Category
-----------	--------	---------	----------

[STATUS]

;ID	Status/Setting
-----	----------------

[PATTERNS]

;ID	Multipliers
-----	-------------

[CURVES]

;ID	X-Value	Y-Value
-----	---------	---------

[CONTROLS]

[RULES]

[ENERGY]

Global Efficiency	75
Global Price	0
Demand Charge	0

```

[EMITTERS]
;Junction      Coefficient

[QUALITY]
;Node          InitQual

[SOURCES]
;Node          Type          Quality      Pattern

[REACTIONS]
;Type  Pipe/Tank      Coefficient

[REACTIONS]
Order Bulk      1
Order Tank      1
Order Wall      1
Global Bulk     0
Global Wall     0
Limiting Potential 0
Roughness Correlation 0

[MIXING]
;Tank          Model

[TIMES]
Duration        0
Hydraulic Timestep 1:00
Quality Timestep 0:05
Pattern Timestep 1:00
Pattern Start   0:00
Report Timestep 1:00
Report Start    0:00
Start ClockTime 12 am
Statistic       None

[REPORT]
Status          No
Summary         No
Page           0

[OPTIONS]
Units           LPS
Headloss        H-W
Specific Gravity 1
Viscosity       1
Trials          40
Accuracy        0.001
CHECKFREQ      2
MAXCHECK        10
DAMPLIMIT      0
Unbalanced     Continue 10
Pattern        1
Demand Multiplier 1.0
Emitter Exponent 0.5
Quality        None mg/L
Diffusivity    1
Tolerance      0.01

[COORDINATES]
;Node          X-Coord      Y-Coord
2              5228.81      1644.07

```

3	5228.81	3033.90
4	6737.29	3033.90
5	8245.76	3033.90
6	9703.39	3033.90
7	9703.39	4593.22
8	9703.39	6135.59
9	9703.39	7745.76
10	8652.54	7745.76
11	8652.54	9000.00
12	8652.54	9864.41
13	6533.90	9864.41
14	7500.00	7745.76
15	6415.25	7745.76
16	5228.81	7745.76
17	5228.81	6508.47
18	5228.81	5389.83
19	5228.81	4220.34
20	3737.29	3033.90
21	3737.29	1661.02
22	3737.29	169.49
23	2211.86	3033.90
24	2211.86	5389.83
25	2211.86	7745.76
26	3211.86	7745.76
27	4245.76	7745.76
28	1177.97	3033.90
29	110.17	3033.90
30	110.17	5389.83
31	127.12	7745.76
32	1245.76	7745.76
1	6415.25	152.54

[VERTICES]

;Link X-Coord Y-Coord

[LABELS]

;X-Coord Y-Coord Label & Anchor Node

[BACKDROP]

DIMENSIONS 0.00 0.00 10000.00 10000.00

UNITS Ninguno

FILE C:\Users\Usuario\Pictures\fondo\Universidad\hano3.jpg

OFFSET 0.00 0.00

[END]