

Trabajo Fin de Máster

Máster Universitario en Organización Industrial y
Gestión de Empresas

Diseño óptimo de redes de distribución de agua
mediante algoritmos de búsqueda tabú

Autor: Alicia Robles Velasco

Tutor: Pablo Cortés Achedad

Dep. Organización Industrial y Gestión de
Empresas II
Escuela Técnica Superior de Ingeniería

Sevilla, 2018



Trabajo Fin de Máster
Máster Universitario en Organización Industrial y Gestión de Empresas

Diseño óptimo de redes de distribución de agua mediante algoritmos de búsqueda tabú

Autor:

Alicia Robles Velasco

Tutor:

Pablo Cortés Achedad

Dep. Organización Industrial y Gestión de Empresas II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Máster: Diseño óptimo de redes de distribución de agua mediante algoritmos de búsqueda tabú

Autor: Alicia Robles Velasco

Tutor: Pablo Cortés Achedad

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

Agradecimientos

A mi tutor, Pablo, y a todos los que enseñando transmiten el gusto por aprender. A mis compañeros de desayuno, porque gracias a ellos he aprendido y crecido cada día. A mi familia, por su apoyo incondicional. Y en particular, a Rafa, por ofrecerme su tiempo y dedicación siempre.

*“El conocimiento es emocionante,
compartirlo y divulgarlo es maravilloso”*

Francisco Robles Maldonado

Resumen

En este Trabajo Fin de Máster se desarrolla un método de optimización que permite resolver el problema de diseño y planificación de redes hidráulicas de distribución.

Las redes de abastecimiento son infraestructuras de gran importancia que suponen grandes inversiones. Con el crecimiento demográfico de las ciudades, y el aumento de la contaminación, están apareciendo nuevos retos de diseño y gestión de las mismas.

Debido a las características intrínsecas de los sistemas hidráulicos y a la naturaleza discreta de algunas de sus variables, éste es un problema no-lineal mixto y entero. Ante este tipo de problemas de alta complejidad han surgido técnicas de resolución, como las metaheurísticas, que permiten obtener resultados satisfactorios en tiempos reducidos. En este trabajo se diseña un algoritmo de búsqueda tabú ajustando los parámetros a la problemática de estudio.

Para la validación del algoritmo se ha resuelto el problema de diseño de redes de distribución aplicado a la red de Alperovits. Esta red ha sido utilizada por muchos autores en la literatura, permitiendo la comparación de los resultados obtenidos en este trabajo con aquellos que obtuvieron otros autores aplicando otras técnicas de resolución.

La ejecución del algoritmo se realiza usando el lenguaje de programación Python y, en concreto, una de sus librerías, WNTR, que permite la generación y simulación de redes hidráulicas.

Abstract

This Final Master Project proposes an optimization method as a tool to the design of water distribution network.

Water distribution networks are very important infrastructures that involve huge investments. Due to the population growth of cities and the increase of pollution, new challenges are emerging in the design and manage of those networks.

The intrinsic characteristics of hydraulic systems and the discrete nature of some of its variables make this a nonlinear mixed integer problem. There are different techniques to face this kind of complex problems like the meta-heuristics which enables satisfactory solutions in a short period of time. In this work, a Tabu Search algorithm, meta-heuristic that simulates the human mind behaviour, is developed. The parameters are carefully selected for the subject of study.

In order to validate the method the design of water distribution network problem have been applied to a specific case, the Alperovits network. Many authors used this network in the past so it enables to compare the results obtained in this work and analyze its efficiency.

The algorithm is implemented in Python, a powerful programming language. It contains many libraries which execute different calculation. WNTR is one of the libraries used in the code that allows simulating hydraulic networks.

Índice

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	xii
Índice de tablas	xiv
Índice de figuras	xvi
1. Objeto del Trabajo	1
2. Introducción	3
3. Descripción del problema.....	7
3.1. Características de las redes de distribución	7
3.2. Componentes	10
3.3. Funcionamiento.....	12
4. Estado del arte.....	15
4.1. Redes modelo	15
4.1.1. La Red de Alperovits and Shamir	15
4.1.2. Red de Hanoi	17
4.1.3. Red de New York.....	20
4.2. Modelo de optimización de redes de distribución	22
4.3. Enfoques para la resolución del problema	28
5. Resolución del caso de estudio mediante Búsqueda Tabú.....	30
5.1. Particularización del modelo	30
5.2. Búsqueda Tabú	35
5.3. Python	41
5.3.1. Librerías.....	41
5.3.2. Paquete WNTR	37
5.3.3. Funciones desarrolladas en Python.....	39

5.3.4. Pseudocódigo	45
6. Resultados	48
6.1. Ajustes de los parámetros del algoritmo	48
6.2. Simulaciones	50
6.3. Evaluación del algoritmo.....	59
6.4. Evaluación hidráulica de las soluciones	65
6.5. Comparación de los resultados con otros enfoques previos	71
7. Conclusiones.....	74
8. Referencias	76
ANEXO	79

Índice de tablas

Tabla 1 - Datos referentes a los nodos - Red de Alperovits	16
Tabla 2 - Longitud de los arcos - Red de Alperovits.....	17
Tabla 3 - Gama de diámetros comerciales - Red de Alperovits.....	17
Tabla 4 - Datos referentes a los nodos - Red de Hanoi	19
Tabla 5 - Datos referentes a los arcos - Red de Hanoi	19
Tabla 6 - Gama de diámetros comerciales – Red de Hanoi	20
Tabla 7 - Datos referentes a los nodos - Red de New York.....	21
Tabla 8 - Datos referentes a los arcos - Red de New York	22
Tabla 9 - Datos y notación de los nodos - Red de Alperovits.....	31
Tabla 10 - Datos y notación de los arcos - Red de Alperovits	32
Tabla 11 - Datos diámetros comerciales - Red de Alperovits	32
Tabla 12 - Coeficientes de Hazen-Williams para la red de Alperovits	32
Tabla 13 - Unidades de medida - WNTR package	39
Tabla 14 - Soluciones de partida.....	48
Tabla 15 - Parámetros de las distintas simulaciones.....	50
Tabla 16 - Entradas simulación 1	50
Tabla 17 - Salidas simulación 1	51
Tabla 18 - Entradas simulación 2	52
Tabla 19 - Salidas simulación 2	52
Tabla 20 - Entradas simulación 3	53
Tabla 21 - Salidas simulación 3	53
Tabla 22 - Entradas simulación 4	53
Tabla 23 - Salidas simulación 4	53
Tabla 24 - Entradas simulación 5	53
Tabla 25 - Salidas simulación 5	54
Tabla 26 - Entradas simulación 6	54

Tabla 27 - Salidas simulación 6	54
Tabla 28 - Entradas simulación 7	54
Tabla 29 - Salidas simulación 7	55
Tabla 30 - Entradas simulación 8	55
Tabla 31 - Salidas simulación 8	55
Tabla 32 - Entradas simulación 9	55
Tabla 33 - Salidas simulación 9	56
Tabla 34 - Entradas simulación 10	56
Tabla 35 - Salidas simulación 10	56
Tabla 36 - Entradas simulación 11	56
Tabla 37 - Salidas simulación 11	57
Tabla 38 - Entradas simulación 12	57
Tabla 39 - Salidas simulación 12	57
Tabla 40 - Simulaciones agrupadas según mejor solución	68
Tabla 41 - Resultados obtenidos por otros autores para la red de Alperovits.....	71
Tabla 42 - Mejor solución encontrada y coste.....	73

Índice de figuras

Figura 1 - Distribución del agua del planeta	3
Figura 2 - Consumo de agua en España y Andalucía	4
Figura 3 - Canal escavado en la roca - Ciudad de Termes, Soria	5
Figura 4 - Red de distribución Dos Hermanas, Sevilla, 2018.....	5
Figura 5 - Red ramificada.....	8
Figura 6 - Red mallada	9
Figura 7 - Red mixta	9
Figura 8 - Tuberías de fundición dúctil	10
Figura 9 - Uniones	11
Figura 10 - Válvula reguladora	11
Figura 11 - Depósitos elevados	12
Figura 12 - Diseño de la Red de Alperovits	16
Figura 13 - Diseño de Red de Hanoi	18
Figura 14 - Diseño de red de New York.....	20
Figura 15 - Diagrama de Moody	27
Figura 16 - Variables secundarias– Red de Alperovits	31
Figura 17 - Representación de búsqueda del mínimo global	35
Figura 18 - Entradas y Salidas de la Búsqueda Tabú	36
Figura 19 - Diagrama de flujo - Búsqueda Tabú.....	37
Figura 20 - Mecanismo de generación de vecindario	39
Figura 21 - Mecanismo generación de nuevas soluciones	40
Figura 22 - Logos software Python - IDE Pycharm.....	41
Figura 23 - Estructura del paquete WNTR de Python	37
Figura 24 - Definición de funciones en Python.....	40
Figura 25 - Función Alperovits_Network	40
Figura 26 - Función Decrease_diameter	41

Figura 27 - Función Change_Network_diameter	41
Figura 28 - Comando para asignar nuevo diámetro.....	42
Figura 29 - Función Eval_Network_Technical.....	42
Figura 30 - Función Invalid	42
Figura 31 - Función Eval_Network_Cost	43
Figura 32 - Función Tabu_Search.....	43
Figura 33 - Gráficas generadas con la función Graph_solution.....	44
Figura 34 - Gráficas generadas con la función Graph_network	45
Figura 35 - Función principal – Python	45
Figura 36 - Coste mejor solución vs Número de iteraciones para Tabu-size=4	58
Figura 37 - Coste mejor solución vs Número de iteraciones para Tabu-size=2	58
Figura 38 - Evolución Solución actual – Simulaciones solución inicial A.....	60
Figura 39 - Evolución Mejor solución – Simulaciones solución inicial A	61
Figura 40 - Evolución Solución actual – Simulaciones solución inicial B	63
Figura 41 - Evolución Mejor solución – Simulaciones solución inicial B	64
Figura 42 - Demanda de los nodos - Simulación 1	66
Figura 43 - Presión en los nodos - Simulación 1	66
Figura 44 - Caudal en los arcos de la red - Simulación 1	67
Figura 45 - Diámetro de los arcos de la red	68
Figura 46 - Caudal de los arcos de la red.....	69
Figura 47 - Velocidad en los arcos de la red.....	70
Figura 48 - Presión en los nodos de la red	70
Figura 49 - Parámetros de red para la mejor solución encontrada	73

1. Objeto del Trabajo

El presente trabajo fin de máster tiene como objetivo la aplicación de algoritmos de búsqueda tabú al problema de diseño de redes hidráulicas de distribución y la comparación de los resultados con los obtenidos por otros autores usando otras técnicas de optimización.

El problema de diseño y planificación de redes hidráulicas de distribución ha sido abordado por diferentes autores en la literatura. No obstante, su estudio no ha sido tan amplio como el de otros problemas de redes, por ejemplo, los de redes de transporte o redes eléctricas. Por ello, parece interesante profundizar en la resolución de este tipo de redes, aportando diferentes estrategias de búsqueda de soluciones.

Debido a sus características, el problema de diseño de redes hidráulicas es un problema no lineal mixto y entero. La no linealidad se debe a las ecuaciones que rigen el comportamiento hidráulico del sistema. Además, los diámetros son variables discretas que solo pueden tomar ciertos valores pertenecientes a una gama de diámetros comerciales. Estas características matemáticas hacen que su resolución sea muy compleja, siendo las metaheurísticas la mejor opción para resolver este tipo de problemas encontrada hasta la actualidad.

Las metaheurísticas son técnicas capaces de resolver problemas de gran complejidad alcanzando soluciones aproximadas en tiempo reducido. Estos métodos imitan algún comportamiento de la naturaleza para explorar el espacio de soluciones, en general, a través de procedimientos iterativos. La escogida en este trabajo es la búsqueda tabú, la cual simula el comportamiento de la mente humana explorando las posibles soluciones del problema. Esta metaheurística ha sido utilizada para la resolución de otros problemas dando muy buenos resultados, por ello se espera un compartimiento similar para la problemática estudiada en este trabajo.

El software utilizado para resolver el problema es Python, un software libre de gran capacidad. Codificar el problema y su correspondiente algoritmo de resolución en lenguaje Python se plantea como una aportación valiosa a la literatura científica relacionada con las redes de distribución de agua.

Para el desarrollo de este trabajo fin de máster, en la sección dos se realiza una introducción al tema de estudio. A continuación, en la sección tres, se describe el problema, explicando los tipos de redes de distribución que existen, sus componentes principales y su funcionamiento. En el capítulo cuatro se hace un recorrido por la literatura científica que ha abordado este problema anteriormente para así conseguir una visión del estado actual del mismo, y de los aspectos que no han sido tratados con tanto detalle. Para ello, se analizan las redes modelos más usadas por los investigadores, los modelos matemáticos y sus características, y las técnicas de resolución aplicadas a esta

problemática.

En el quinto capítulo se presenta la particularización del caso de estudio resuelto en este trabajo, la red modelo simulada, el modelo matemático escogido, el algoritmo de resolución aplicado, y el lenguaje de codificación y software de resolución utilizado. En el capítulo seis se muestran los resultados obtenidos para las diferentes simulaciones realizadas con el fin de validar el modelo. Para ello, se comparan los resultados con los obtenidos por otros autores que resolvieron la misma red hidráulica con distintos métodos. En el capítulo siete se presentan las conclusiones obtenidas a partir de los resultados.

Por último, el octavo capítulo recoge las referencias bibliográficas consultadas. Al final del trabajo se añade un anexo con el código completo del algoritmo.

2. Introducción

El agua ocupa un 70% de la superficie total de la tierra, además es parte constituyente de todos los seres vivos. Del total de agua existente en el planeta se estima que solo el 2.5% se considera agua dulce, siendo el otro 97.5% agua salada. Además, la mayoría del agua dulce, un 69% aproximadamente, se encuentra en la Antártida en estado sólido y otro 30.1% son aguas subterráneas, humedales o acuíferos por lo que se consideran inaccesible para la población. Como consecuencia, el agua dulce, superficial y accesible a los seres humanos se reduce a tan solo el 0.007% del agua que forma el planeta y dicha cantidad se ve reducida cada año debido a la contaminación [1]. Garantizar el acceso a agua potable es uno de los desafíos más importantes del mundo en la actualidad.

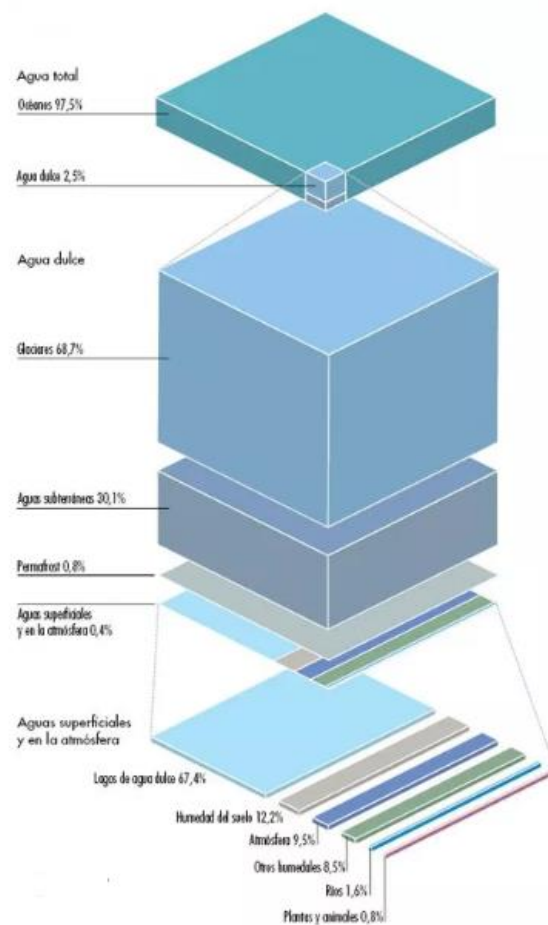


Figura 1 - Distribución del agua del planeta
Fuente: GEO4, 2006

En la figura 1 se muestra de manera gráfica la distribución de agua en el planeta en el año 2006.

El uso del agua es de lo más variado y dependiendo de este uso el agua requerirá una calidad menor o mayor. Este hecho es importante a la hora de planificar el ciclo del agua en una red, permitiendo el aprovechamiento de aguas para usos que requieran una calidad menor. Existen muchos factores que afectan al consumo de agua como el tamaño de la ciudad o núcleo urbano, el nivel de vida, el nivel de industrialización, el clima, el turismo, etc. En España el consumo de agua se distribuye en: 80% para agricultura o regadío, 14% para abastecimiento de núcleos urbanos y el otro 6% para uso industrial. El abastecimiento de núcleos urbanos incluye el agua apta para el consumo humano, conocida como agua potable, y el agua utilizada para servicios públicos como extinción de incendios. Cada cual cuenta con su red de distribución independiente. Al ser la calidad exigida menor para servicios públicos en muchas ocasiones se utiliza agua recuperada.

En las últimas estadísticas del Instituto Nacional de Estadística (INE, 2014) el consumo medio en España era de 132 L/hab · día, cantidad que engloba todos los usos derivados de la vida cotidiana como el aseo, la alimentación, o el uso de electrodomésticos. Estos datos se han utilizado para la confección del siguiente gráfico.

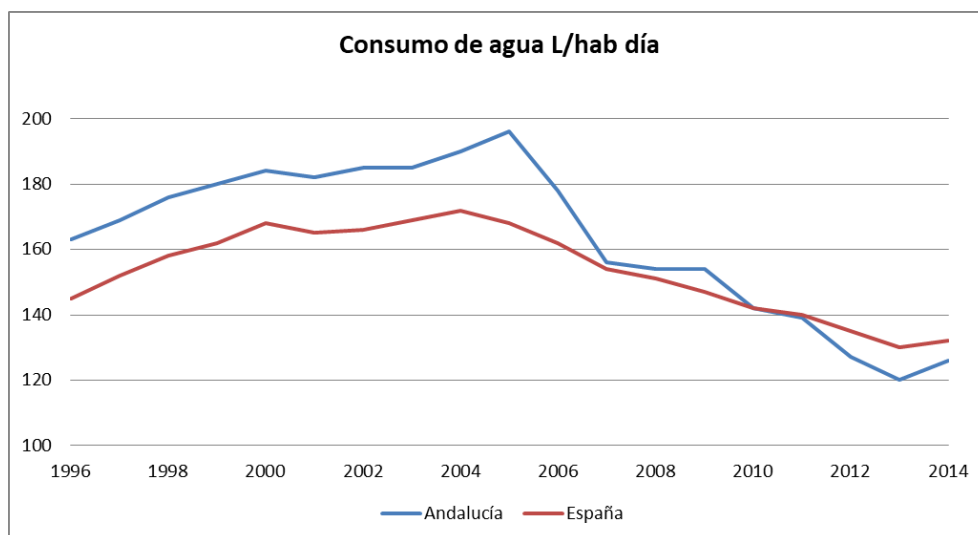


Figura 2 - Consumo de agua en España y Andalucía
Fuente: Elaboración propia

Como se observa en la figura 2, el consumo per cápita de agua en España ronda de los 110 a los 200 litros por habitante y día. Se observa que ha habido una reducción de dicho consumo desde los años 2004-2005, años con consumos más elevados, hasta el 2014, último año del que se tienen datos fiables. Se muestran los datos de Andalucía y España donde puede apreciarse que el consumo de agua en Andalucía se ha reducido bastante con respecto al resto de España en los últimos 20 años.

Las redes de abastecimiento surgen ante la necesidad del transporte del agua desde su fuente hasta la población. El origen de las redes de distribución de agua en España se remonta a la civilización romana, ya que está desarrolló infraestructuras de captación, transporte y distribución. Antiguamente las grandes civilizaciones se concentraban cerca

de grandes ríos para aprovechar el fácil acceso al agua que este recurso natural ofrece. El desarrollo y la evolución de los métodos de captación, tratamiento y transporte de agua han hecho posible la aparición de grandes centros urbanos en las condiciones más inhóspitas como es el caso de Las Vegas, ciudad localizada en el medio de un desierto.



Figura 3 - Canal escavado en la roca - Ciudad de Termes, Soria
Fuente: <https://www.romanorumvita.com/>

La figura 3 se muestra un canal de agua romano de la antigua ciudad de Termes, Soria. Se observa como la estructura estaba preparada para estar cubierta en su totalidad.



Figura 4 - Red de distribución Dos Hermanas, Sevilla, 2018
Fuente: EMASESA

Como se observa en la figura 4, obra de acondicionamiento y mejora de las redes de abastecimiento en Sevilla, actualmente las redes de distribución también están, generalmente, enterradas.

No se podría imaginar la vida moderna como la conocemos si no fuera por el estado actual de las redes hidráulicas tanto de abastecimiento como de saneamiento. El impacto generado por estos sistemas abarca más que la facilidad de obtener el agua potable para el uso diario. El mantenimiento de los índices de calidad y la abundante oferta de este recurso garantizado por las redes de abastecimiento tienen un profundo impacto en la salud de la población. El acceso a agua limpia se relaciona directamente con la reducción de la mortalidad infantil y la prevención de graves enfermedades, entre otras cosas.

Hoy en día la población se concentra en núcleos urbanos cada vez mayores, este crecimiento de las ciudades genera la necesidad de ampliar constantemente las redes de abastecimiento. El reto del diseño y la gestión de dichos sistemas es cada día más difícil.

El diseño y la construcción de una red hidráulica es un trabajo complejo y ostentoso. Una planificación ineficiente puede suponer grandes gastos públicos o de las empresas encargadas de la construcción y mantenimiento de la red. Por ello, este campo tiene una gran proyección de futuro, cobrando mayor importancia a medida que aparecen nuevos avances tecnológicos que permiten la mejora de estos sistemas.

3. Descripción del problema

La red de abastecimiento es la encargada de suministrar agua a los consumidores en condiciones adecuadas dependiendo del uso al que esté destinada. El abastecimiento de agua consta de varias fases: Captación, transporte, tratamiento, almacenamiento y distribución.

La captación es la fase inicial y consiste en la toma de agua de las diferentes fuentes que existen en la naturaleza. A continuación, el agua es transportada desde las fuentes de abastecimiento hasta los núcleos urbanos donde es tratada y almacenada para su posterior distribución. El tratamiento del agua tiene como objetivo principal conseguir la calidad adecuada del agua dependiendo del uso al que esté destinada, además, se busca mantener las instalaciones de almacenamiento y distribución posteriores en buen estado. El almacenamiento sirve para asegurar el suministro de agua a los consumidores ante diferentes situaciones y regular caudales y presiones en la red de distribución.

La última etapa de una red de abastecimiento es la distribución. Este trabajo se centra en dicha etapa que comienza a la salida de la planta de tratamiento de agua o depósitos de almacenamiento, y llega a su fin en el punto de conexión con la instalación de suministro del consumidor.

3.1. Características de las redes de distribución

Una red de distribución se define como el conjunto de tuberías y elementos de control que permite el acceso al agua potable a los consumidores.

En general, la red de distribución de una ciudad está constituida por multitud de tuberías enlazadas entre sí. Una clasificación de los distintos tramos de la red según su rango podría ser [5]:

- Red de transporte: tramo constituido por las tuberías que conectan la planta de tratamiento, depósitos o estación de bombeo con la red arterial. Se caracteriza por contar con tuberías de diámetros mayores.
- Red arterial: constituida por los tramos que enlazan los diferentes sectores del área a abastecer.
- Red secundaria: tuberías que conectan la red arterial con las acometidas de suministros, tomas contra incendios y bocas de riego.
- Acometida: tramo que conecta las instalaciones de suministro directo al consumidor con la red secundaria.

Existen muchas clasificaciones de las redes atendiendo a diferentes criterios. Según el tipo

de consumo se pueden diferenciar redes para consumo humano, para servicios públicos como extinción de incendios o riego de parques y zonas comunes, y redes para usos industriales. Según el número de depósitos de la red, éstas pueden ser redes de un solo depósito, sin depósitos o con varios depósitos. Según la presión existen redes de alta presión y redes de baja presión. Otra clasificación es entre redes que cuente o no con sistemas de impulsión. Dependiendo de la topología del terreno existen redes en un piso o redes escalonadas. A continuación, se muestran los tipos de redes según su estructura en planta:

Ramificada

Son redes que cuentan con una tubería principal o primaria de donde parten otras tuberías secundarias y de éstas otras, etc. Normalmente cuentan con un solo nodo fuente situado al comienzo de la tubería principal. El agua circula por la red en un único sentido y se localizan finales de red.

Su diseño es sencillo y la regulación de presiones en la misma también, no obstante, en el caso de surgir ampliaciones de la red pueden aparecer presiones ineficientes. En este tipo de redes hay que asegurar la calidad del agua al final del ramal, pues existe mayor tendencia al estancamiento si se produce un bajo uso de algunos ramales.

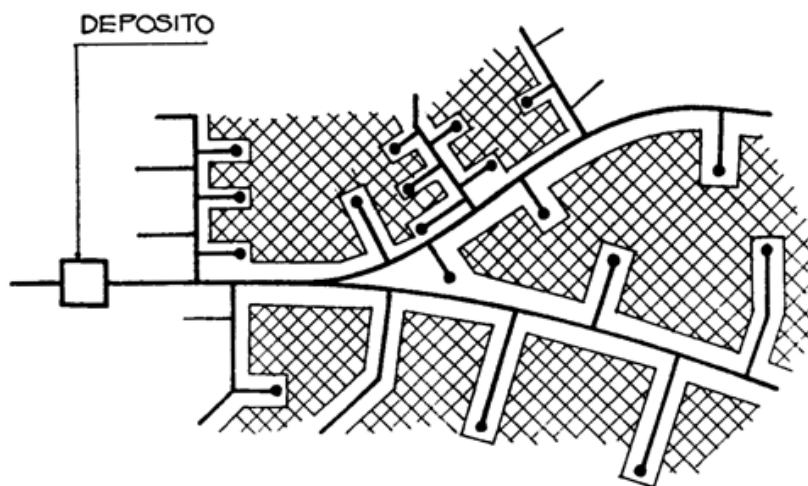


Figura 5 - Red ramificada
Fuente: Navarro & Prieto, 2011

En la figura 5 se muestra el diseño de una red ramificada donde puede observarse la rama principal alimentada por un solo depósito.

Mallada

En este tipo de redes cada punto de demanda puede estar alimentado por más de una tubería. Generalmente las redes malladas cuentan con más de un depósito, al menos uno de cabecera, en el comienzo de la red y otro en el fin para mantener el equilibrio de presiones en los nodos. A diferencia de las redes ramificadas, en cada tramo el agua puede circular en ambos sentidos. La pérdida de calidad del agua es más improbable, ya que en general, el agua está en circulación. Sin embargo, el diseño de estas redes es más

laborioso.

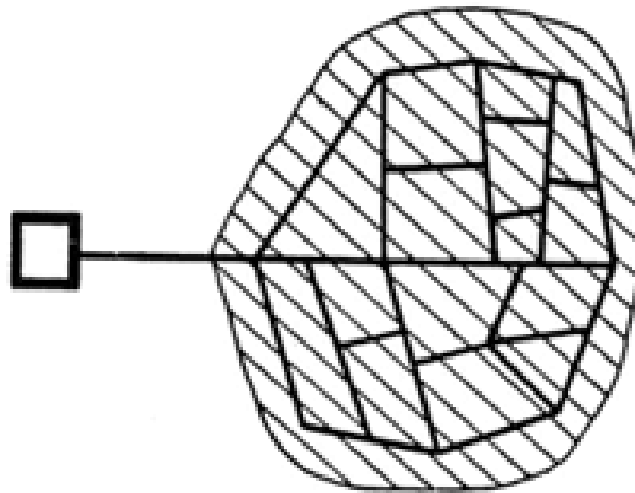


Figura 6 - Red mallada
Fuente: Navarro & Prieto, 2011

En la figura 6 se observa un ejemplo de red mallada con un solo depósito de cabecera. Cada anillo o recorrido cerrado que puede seguir el agua se denomina malla o bucle.

Mixtas

Es una combinación de los dos tipos anteriores, ya que cuenta con mallas y con ramales. Son las más utilizadas en las instalaciones reales.

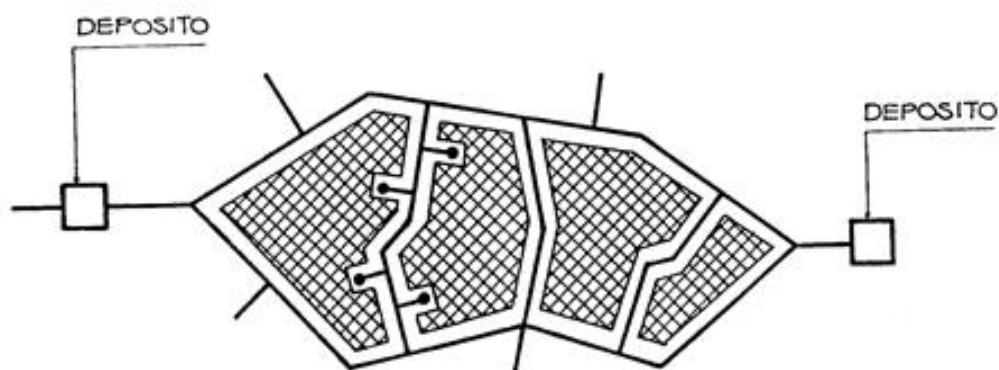


Figura 7 - Red mixta
Fuente: Navarro & Prieto, 2011

En la figura 7 puede observarse una red mixta genérica. Este tipo de redes cuenta con una o varias tuberías principales, circulares o lineales, que conectan la mayoría de mallas o ramales de la red.

Las diferencias principales entre los tipos de redes expuestos recaen en el coste y la calidad, siendo las redes malladas considerablemente más caras, pero, asegurando una

mayor calidad y fiabilidad a la red. El coste de inversión inicial es mayor ya que este tipo de redes supone una mayor infraestructura, más kilómetros de tuberías. Además, el diseño de la red es mucho más complejo. No obstante, en el caso de existir una avería se mantiene el suministro a los consumidores ya que éstos están conectados a la red por más de un camino.

Por el contrario, las redes ramificadas conllevan una inversión inicial menor pero la fiabilidad de la red es también menor. En caso de avería en algún punto de la red todo nodo consumidor de la red posterior al punto de avería quedará desabastecido.

El tipo elegido dependerá de muchos factores como la disponibilidad financiera, la densidad demográfica, las características de uso de la red, las distancias entre los nodos consumidores, entre otros. Como en la mayoría de las redes de abastecimiento existe gran diversidad de condiciones en sus diversos puntos, la red mixta suele ser una opción más viable y por ello la más utilizada.

3.2. Componentes

Una de las premisas de todos los componentes de la red es que no deben causar alteración alguna en la calidad del agua.

Los elementos básicos de la red pueden clasificarse en elementos lineales y elementos puntuales.

Tuberías

Son elementos lineales y esenciales para el transporte de fluidos. Constan de una sección transversal uniforme, generalmente circular, y en sentido longitudinal son rectos. Están definidos por su diámetro, material y longitud. Los materiales más comunes utilizados para la construcción de tuberías de uso hidráulico son:

- Tuberías metálicas: de acero o FD (fundición dúctil).
- Tuberías de hormigón armado y pretensado.
- Tuberías de materiales plásticos: PVC, PE (Polietileno) o PRFV (Poliéster reforzado con fibra de vidrio).
- Tuberías de materiales cerámicos: GRES.

La elección del material de la tubería debe estar condicionada al uso al que está destinada.



Figura 8 - Tuberías de fundición dúctil
Fuente: EMASESA

En la figura 8 se observa un grupo de tuberías de fundición dúctil.

Uniones

Permiten enlazar dos elementos consecutivos de la red. Se distinguen sistemas de unión flexibles y rígidos. Los primeros permiten la desviación angular durante y después de la instalación mientras que los segundos no permiten el desvío angular ni durante ni después de la instalación.



Figura 9 - Uniones
Fuente: Navarro and Prieto, 2011

En la figura 9 se observa, a la izquierda una unión flexible, y a la derecha una unión de junta fija.

Válvulas

Elementos puntuales instalados entre las tuberías que permiten regular o cortar el caudal y la presión. Estos elementos se clasifican según su función en la red, existiendo: válvulas de toma, válvulas de corte, válvulas reguladoras de presión y caudal o válvulas de retención.



Figura 10 - Válvula reguladora
Fuente: Navarro and Prieto, 2011

En la figura 10 se observa una válvula reguladora de presión que reduce la presión a un valor fijo e independiente del caudal.

Depósitos

Estructura acta para el almacenamiento de un cierto volumen de agua. Sus funciones principales son el almacenamiento de agua y la regulación de caudales. Calcular el volumen óptimo de los depósitos es importante, hay que hacer un balance entre el tiempo de abastecimiento que se desea cubrir y el tiempo máximo de permanencia del agua en el depósito. Los depósitos de cabecera de la red, generalmente, se dimensionan ajustándose a la demanda de 1 día de consumo.



Figura 11 - Depósitos elevados
Fuente: Navarro and Prieto, 2011

En la figura 11 se observan tres depósitos elevados. Según su posición en el terreno éstos pueden ser también superficiales o semienterrados.

Como se explicó en el apartado anterior, las redes de distribución pueden contar o no con estos sistemas.

Bombas

Sistemas de impulsión que aportan la energía necesaria al sistema. En general se utilizan para mover el agua de una zona de menor presión o altitud a otra de mayor presión o altitud. Las pérdidas de carga hacen que muchas veces este instrumento sea necesario para garantizar la presión adecuada en los nodos consumidores.

Además de los elementos definidos, existen muchos otros que pueden formar parte de la red, como ventosas, codos o desagües. Usualmente, las redes cuentan con dispositivos de medida que permiten el control del estado de la red, los más habituales son los manómetros, miden la presión, y los caudalímetros, encargados de medir el caudal.

3.3. Funcionamiento

Para el diseño de una red hidráulica de distribución se parte del conocimiento del diseño y trazado de la red, así como de la distribución de la demanda o nodos consumidores. Los parámetros que caracterizan el funcionamiento de una red son: la presión, la pérdida de

carga, la velocidad del agua, el diámetro de las tuberías y caudales que recorren las mismas. A continuación, se explica cada uno de estos parámetros, así como las diferentes unidades de medida de los mismos.

Presión

Es la magnitud física que expresa la fuerza ejercida por un cuerpo por unidad de superficie. En hidráulica esta magnitud se mide en metros por columna de agua (m.c.a.), unidades que expresan la presión ejercida por una columna de agua de 1 metro de altura. La normativa establece la presión de servicio como la presión mínima en el punto de conexión con el consumidor con caudal nulo en la acometida [5]. Cuando se aborde el problema de diseño hay que tener en cuenta esta restricción a la hora de elegir los diámetros de las tuberías.

La presión máxima de diseño es la máxima presión de funcionamiento que puede alcanzarse en una sección de la tubería.

Pérdida de carga

Es un parámetro fundamental en el diseño de la red. Las pérdidas de carga representan las pérdidas de energía del agua a medida que ésta atraviesa la red, lo que resulta en suministros con presiones más bajas. Las pérdidas principales se deben al rozamiento del agua con las paredes de las tuberías, existiendo también pérdidas ocasionadas por elementos puntuales de la red. Alguno de los factores que determinan las pérdidas lineales son la rugosidad y el diámetro de la tubería.

En cualquier red se busca minimizar las pérdidas de carga, reduciéndose así los aportes de energía necesarios a la red y por ello los costes.

Velocidad del agua

La velocidad es un parámetro que debe ser controlado en la red. Velocidades demasiado altas pueden conllevar a desgaste excesivo o incluso rotura de tuberías, mientras que velocidades muy bajas ocasionan el estancamiento del agua y, por ello, la pérdida de calidad de la misma. En general y con carácter meramente orientativo, se deberá procurar que la velocidad de circulación del agua dentro de las tuberías alcance un valor comprendido entre 0,3 y 2,0 m/seg [5]. Estos límites hacen que el sistema de abastecimiento de agua se comporte como un sistema hidráulico de conductos forzados (donde el fluido ejerce presión en todos los puntos de la tubería). Esta característica es diferente para los sistemas de transporte de aguas residuales, sistemas hidráulicos de conductos libres, por sus menores exigencias de mantenimiento y de calidad. Un sistema hidráulico de conductos forzados resulta en una red con mayores exigencias y, por tanto, más caras.

Diámetros

Los diámetros de las tuberías están acotados a ciertos valores de una gama comercial que depende del material. La elección de los diámetros depende de diversos factores una vez que éstos influyen en los costes de instalación y utilización del sistema, así como en las presiones de los nodos consumidores.

Caudal

Es la cantidad de agua que atraviesa una sección por unidad de tiempo. El caudal en los diferentes tramos de la red está condicionado a la cantidad de demanda de los nodos consumidores, siendo en general una premisa satisfacer dicha demanda.

4. Estado del arte

En este apartado se realiza un análisis del estado actual del diseño de redes hidráulicas de distribución. Se comienza explicando las redes modelos más utilizadas en la literatura. A continuación, se resumen los modelos matemáticos que representan el funcionamiento de una red hidráulica. Y por último, los enfoques de los diferentes autores para resolver este problema.

4.1. Redes modelo

Las siguientes redes que se muestran son las tres más utilizadas en la literatura para simular el comportamiento de algoritmos de resolución aplicados al problema de diseño de redes de distribución de agua potable. Los distintos autores que se enfrentan a este problema y lo hacen con diferentes métodos comparan los resultados obtenidos con los que previamente obtuvieron otros autores. De esta forma es fácil determinar cuan eficaz es un nuevo método de resolución.

4.1.1. La Red de Alperovits and Shamir

La primera red que se expone es la red de Alperovits and Shamir (1977). Ésta es una red mallada constituida por 2 mallas, 7 nodos y 8 arcos. El suministro de agua es realizado en totalidad por un único nodo fuente. Es una red de circulación por gravedad, es decir, la red no requiere impulsión.

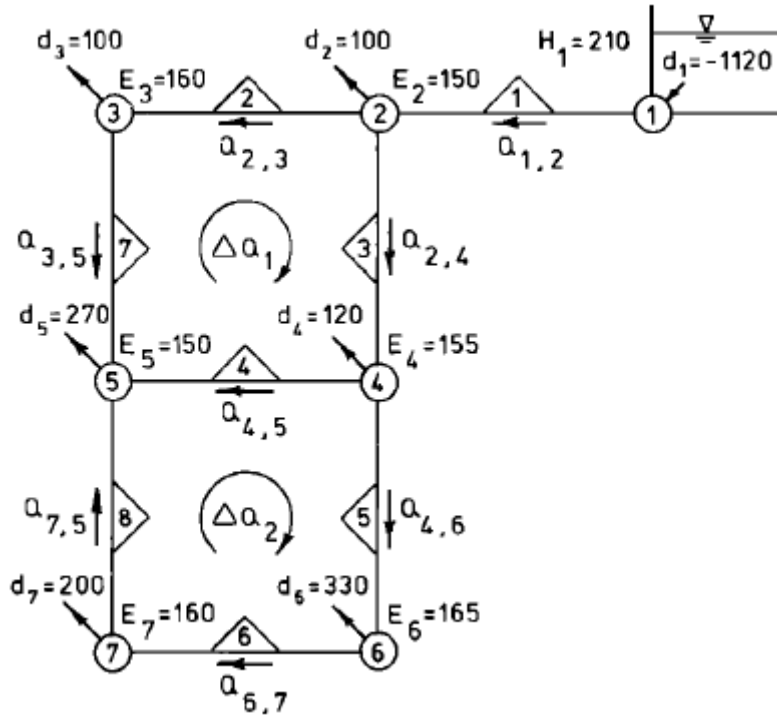


Figura 12 - Diseño de la Red de Alperovits
Fuente: Alperovits and Shamir, 1997

Como se observa en la figura 12 la red está definida por la altura de sus nodos (E_i, H_i) y la demanda de los mismos (d_i). Al ser una red mallada hay que definir hipotéticamente el sentido del flujo de agua ($Q_{i,j}$) y si finalmente, este tiene signo negativo, el sentido será el opuesto. El nodo fuente, nodo 1, es un embalse con una cota constante de 210 metros sobre el nivel del mar.

Tabla 1 - Datos referentes a los nodos - Red de Alperovits
Fuente: Alperovits and Shamir, 1997

Nodo	Altura E_i [m]	Mínima presión permitida [m.c.a]	Demanda d_i [m³/h]
1	210,0	0,0	-1120,0
2	150,0	30,0	100,0
3	160,0	30,0	100,0
4	155,0	30,0	120,0
5	150,0	30,0	270,0
6	165,0	30,0	330,0
7	160,0	30,0	200,0

La tabla 1 recoge la altura de los nodos consumidores de la red y la mínima presión de servicio que se debe satisfacer. Además, la demanda en cada nodo es conocida.

Tabla 2 - Longitud de los arcos - Red de Alperovits
Fuente: Alperovits and Shamir, 1997

Arco	1	2	3	4	5	6	7	8
Longitud [m]	1000	1000	1000	1000	1000	1000	1000	1000

En la tabla 2 se observa que la longitud de todos los arcos de la red es igual a 1000 metros.

Existe un total de 14 diámetros comerciales, cada uno con un coste unitario por unidad de longitud. En la siguiente tabla se recogen dichos diámetros medidos en pulgadas (in) unidades originales, su equivalencia en metros (m), unidades del sistema internacional, y por último el coste unitario asignado a cada diámetro.

$$1m = 39,3701in$$

Tabla 3 - Gama de diámetros comerciales - Red de Alperovits
Fuente: Alperovits and Shamir, 1997

Diámetros [in]	1	2	3	4	6	8	10
Diámetros [m]	0,0254	0,0508	0,0762	0,1016	0,1524	0,2032	0,254
Coste unitario (u.m./m.L.)	2	5	8	11	16	23	32
Diámetros [in]	12	14	16	18	20	22	24
Diámetros [m]	0,3048	0,3556	0,4064	0,4572	0,508	0,5588	0,6096
Coste unitario (u.m./m.L.)	50	60	90	130	170	300	550

En la tabla 3 se muestra el valor de cada diámetro existente y su coste unitario por unidad de longitud asociado.

El número de posibles combinaciones para una solución que represente el diámetro a instalar en cada tubería es $14^8 = 1,475,789,056$, sin tener en cuenta requisitos técnicos de la red.

4.1.2. Red de Hanoi

La red de Fujiwara y Khang (1990), más conocida como red de Hanoi, representa la red de abastecimiento de la ciudad de Hanoi, Vietnam. Es una red mixta formada por 32 nodos y 34 arcos, la distribución de los mismos se observa en la siguiente figura, observándose tres mallas y dos ramas independientes. Al igual que en la red anterior, existe un único nodo fuente, el nodo 1, y es una red de circulación por gravedad, ya que este nodo fuente está a

una cota de 100 metros sobre el nivel del mar.

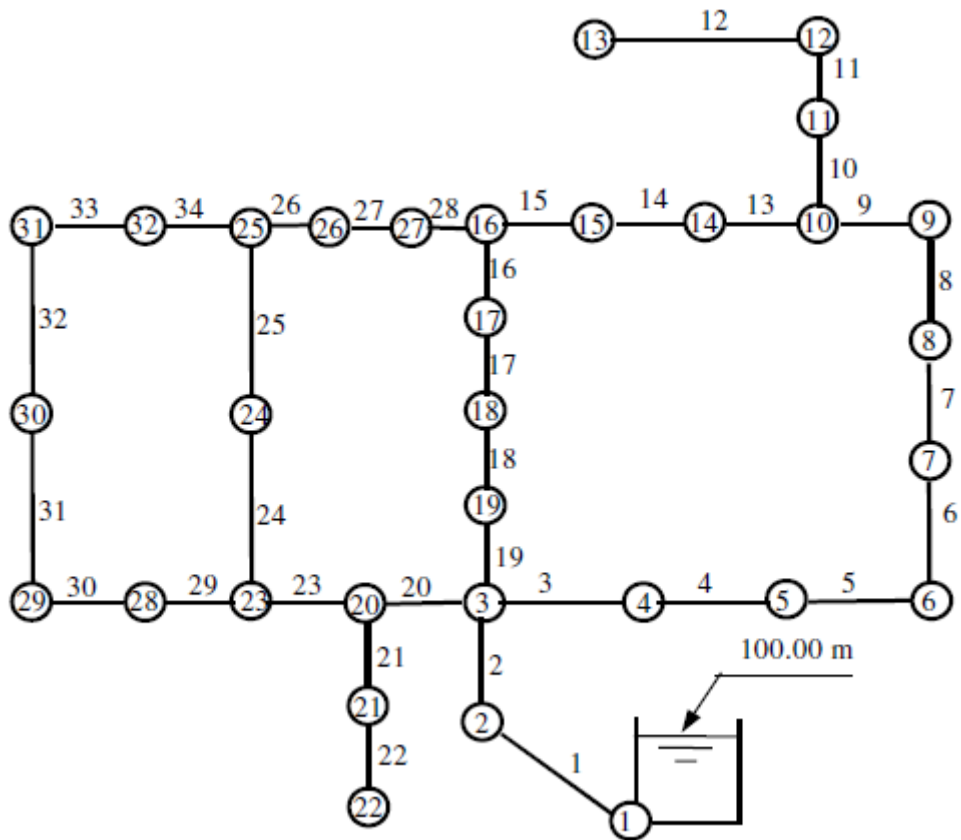


Figura 13 - Diseño de Red de Hanoi
Fuente: Fujiwara y Khang, 1990

La figura 13 muestra el diseño de la red de Hanoi donde puede verse la distribución de los nodos consumidores y las tuberías que los conectan.

La red está definida por los caudales que atraviesan cada arco, para mantener el formato seguido al mostrar los datos de la red de Alperovits, se han calculado los caudales de demanda en cada nodo y son los que se muestran en la tabla 4.

Tabla 4 - Datos referentes a los nodos - Red de Hanoi

Nodo	Altura [m]	Mínima presión permitida [m.c.a.]	Demanda [m3/h]	Nodo	Altura [m]	Mínima presión permitida [m.c.a.]	Demanda [m3/h]
1	100	0	0,0	17	0	30	865,0
2	0	30	890,0	18	0	30	1345,0
3	0	30	850,0	19	0	30	60,0
4	0	30	130,0	20	0	30	1275,0
5	0	30	725,0	21	0	30	930,0
6	0	30	1005,0	22	0	30	485,0
7	0	30	1350,0	23	0	30	1045,0
8	0	30	550,0	24	0	30	820,0
9	0	30	525,0	25	0	30	170,0
10	0	30	525,0	26	0	30	900,0
11	0	30	500,0	27	0	30	370,0
12	0	30	560,0	28	0	30	290,0
13	0	30	940,0	29	0	30	360,0
14	0	30	615,0	30	0	30	360,0
15	0	30	280,0	31	0	30	105,0
16	0	30	310,0	32	0	30	805,0

La altura del nodo fuente es de 100 metros sobre el nivel del mar, estando el resto de nodos a nivel del mar. La presión mínima exigida para cada nodo consumidor es, al igual que en la red anterior, de 30 m.c.a, excepto para el nodo fuente que no requiere ninguna presión específica.

Tabla 5 - Datos referentes a los arcos - Red de Hanoi

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

En la tabla 5 se observan la longitud en metros de todos los tramos representados en la figura 13.

Por último, el número de diámetros disponibles en esta red es considerablemente menor al caso anterior, con un total de seis diámetros, lo que conlleva a un total de $6^{34} = 2.865118 \cdot 10^{26}$ posibles configuraciones.

Tabla 6 - Gama de diámetros comerciales – Red de Hanoi

Diámetros [in]	12	16	20	24	30	40
Diámetros [m]	0,3048	0,4064	0,508	0,6096	0,762	1,016
Coste unitario [u.m/m.L]	45,7	70,4	98,4	129,3	180,7	278,3

En la tabla 6 se muestran las medidas de los seis diámetros disponibles en pulgadas (in) y su conversión a metros (m) para mantener el formato seguido en la presentación de la red anterior. El coste unitario por unidad de longitud se presenta en la última fila de la tabla 6.

4.1.3. Red de New York

La red de Quindry et al. (1981), más conocida como red de New York, simula la distribución de tuberías en la ciudad de Nueva York (E.E.U.U.).

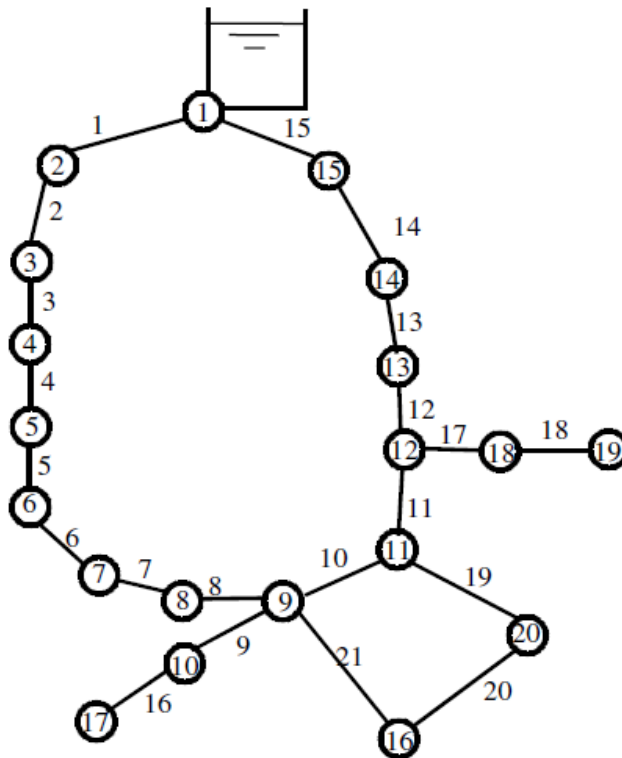


Figura 14 - Diseño de red de New York
Fuente: Quindry et al., 1981

En la figura 14 se muestra la estructura en plante de la red. Es una red usada por un varios autores para comparar sus métodos de resolución. No obstante, se observa que en la mayoría de los trabajos, los autores han utilizado distintas gamas comerciales de diámetros, por lo que resulta más difícil la comparación cuantitativa de los resultados.

Tabla 7 - Datos referentes a los nodos - Red de New York

Nodo	Mínima altura [ft]	Mínima altura [m]	Demanda [ft³/s]	Demanda [m³/s]
1	300,0	91,4	-2017,5	-57,1
2	255,0	77,7	92,4	2,6
3	255,0	77,7	92,4	2,6
4	255,0	77,7	88,2	2,5
5	255,0	77,7	88,2	2,5
6	255,0	77,7	88,2	2,5
7	255,0	77,7	88,2	2,5
8	255,0	77,7	88,2	2,5
9	255,0	77,7	170,0	4,8
10	255,0	77,7	1,0	0,0
11	255,0	77,7	170,0	4,8
12	255,0	77,7	117,1	3,3
13	255,0	77,7	117,1	3,3
14	255,0	77,7	92,4	2,6
15	255,0	77,7	92,4	2,6
16	255,0	77,7	170,0	4,8
17	255,0	77,7	57,5	1,6
18	255,0	77,7	117,1	3,3
19	255,0	77,7	117,1	3,3
20	255,0	77,7	170,0	4,8

La red está caracterizada por contar con veinte nodos, la mínima altura y la demanda de cada uno de ellos se recoge en la tabla 7.

Tabla 8 - Datos referentes a los arcos - Red de New York

Arco	Longitud [ft]	Longitud [m]	Arco	Longitud [ft]	Longitud [m]
1	11600	3535,68	12	12200	3718,56
2	19800	6035,04	13	24100	7345,68
3	7300	2225,04	14	21100	6431,28
4	8300	2529,84	15	15500	4724,4
5	8600	2621,28	16	26400	8046,72
6	19100	5821,68	17	31200	9509,76
7	9600	2926,08	18	24000	7315,2
8	12500	3810	19	14400	4389,12
9	9600	2926,08	20	38400	11704,32
10	11200	3413,76	21	26400	8046,72

La tabla 8 recoge los datos referentes a los arcos de la red, un total de 21 arcos. La red original utiliza los pies (ft) como unidad de medida de longitud. Para mantener el formato de las dos redes modelo mostradas anteriormente, estas medidas se transforman a metros (m), como puede apreciarse en las tablas 7 y 8.

4.2. Modelo de optimización de redes de distribución

El diseño óptimo de redes hidráulicas de distribución es un problema en el cual se pretende satisfacer la demanda de agua, con caudal y presión específica, de una serie de nodos consumidores. A lo largo de la historia han aparecido diferentes modelos que simulan el comportamiento de una red hidráulica de distribución, pero todos ellos incluyen las principales características del problema.

El punto de partida es la correcta definición de la topología de la red, tanto su estructura como su demanda. La estructura de la red se define por la localización de los nodos, coordenadas y elevación, y la longitud de los arcos que representan las tuberías. La demanda de los nodos consumidores se estima a través de la predicción del crecimiento de la población y el consumo per cápita, datos en general de fácil acceso.

El comportamiento de la red se rige por leyes de conservación de masa y energía, ambas se introducen en el modelo en forma de restricción. El objetivo es la elección de un diámetro específico para cada tramo de la red minimizando el coste total a la vez que se satisfacen una serie de restricciones técnicas [10]. Los diámetros asignados a cada tramo de tubería son, en general, las variables principales del modelo, estas variables son discretas ya que deben ser escogidas de entre una serie de diámetros comerciales previamente establecidos.

Actualmente los modelos más utilizados son los modelos de programación discreta ya que son simples y concisos favoreciendo el alcance de soluciones en relativamente poco tiempo.

Un criterio de clasificación de los diferentes modelos matemáticos que simulan el comportamiento de una red hidráulica de distribución puede ser según su función objetivo. Los modelos de redes, en general, buscan minimizar costes, sin embargo, una

estrategia común es añadir términos de penalización en la función objetivo por incumplimiento de ciertos requisitos de la red. Por lo que el objetivo del modelo pasa de ser la minimización de los costes de red a incluir la minimización del incumplimiento de las especificaciones. Según este criterio una clasificación de los modelos de redes de abastecimiento podría ser la siguiente [11]:

- Modelo sin costes de penalización.
- Modelo con costes de penalización por incumplimiento de caída de presiones.
- Modelo con costes de penalización por incumplimiento de caída de presiones y velocidad.
- Modelo con costes de penalización por incumplimiento de caída de presiones, velocidad y fiabilidad.

Cada modelo añade una penalización diferente a la función objetivo. El uso de un modelo u otro dependerá de las características de red y las prioridades de la misma.

El modelo general sin costes de penalización es el más usado por los autores que han abordado el problema. Este modelo no contempla términos de penalización en la función objetivo si se incumplen alguna de los requisitos establecidos.

Se procede a desarrollar el modelo explicando los conjuntos, los datos y las variables que lo forman. A continuación, se muestra el modelo completo y se realiza una explicación detallada de la función objetivo y de cada una de las restricciones.

Conjuntos

N : Conjunto de nodos (tanto consumidores, como nodo fuente)

E_i : Conjunto de arcos que llegan al nodo i . $\forall i = 1, \dots, N$

S_i : Conjunto de arcos que salen del nodo i . $\forall i = 1, \dots, N$

A : Conjunto total de arcos

$$A = E_i \cup S_i \quad \forall i = 1 \dots N$$

M : Conjunto de mallas

K : Conjunto de diámetros de la gama comercial

R : Conjunto de números de reales

Datos

d_i : Demanda del nodo i . $\forall i = 1, \dots, N$

$Cota_i$: Cota del nodo i . $\forall i = 1, \dots, N$

L_j : Longitud del tramo o arco j . $\forall j = 1, \dots, A$

$C[\emptyset_k]$: Coste unitario de tubería para el diámetro \emptyset_k . $\forall k = 1, \dots, K$

Variables

ϕ_{kj} : Diámetro $k \in K$, gama de diámetros comerciales, asignado al tramo j .

q_j : Caudal en el tramo j .

h_i : Presión en el nodo i .

H_j : Caída de presión en el arco j .

Se define el conjunto total de nodos y arcos que forman la red (N y A), dos subconjuntos de arcos por cada nodo (E_i y S_i) que representan los arcos que llegan y salen del nodo i . También el conjunto de mallas (M), entendiéndose por malla cualquier conjunto de arcos que forman un recorrido cerrado o bucle en la red. Por último, es necesario definir el conjunto de diámetros posibles a asignar a cada tubería (K).

Como se ha comentado, el punto de partida es definir la topología de la red por sus nodos y sus arcos. La posición de los nodos, coordenadas (x,y) , es un dato aunque no aparezca en el modelo. $Cota_i$ representa la altura sobre el nivel del mar de cada nodo i , y d_i su demanda expresada en *volumen/tiempo*. L_j es la longitud de cada tramo donde debe instalarse una tubería, además se conocen los nodos que une cada uno de los arcos. Cada diámetro de la gama comercial tiene asociado un coste unitario por unidad de longitud $C(\phi_k)$.

Las variables principales del modelo son los diámetros de las tuberías en cada arco de la red, ϕ_{kj} donde k representa el diámetro de la gama comercial K a instalar en el arco j . Dichos diámetros son variables discretas ya que solo pueden tomar ciertos valores predefinidos. La naturaleza de estas variables es una de las causas de la complejidad de la resolución del modelo, explicada en más detalle en apartados posteriores. El resto de variables, o variables secundarias, son los caudales en los arcos, q_j , y las presiones en los nodos, h_i . Estas variables se calculan a través de un proceso iterativo ya que las ecuaciones que las rigen no son lineales.

Existen modelos que limitan los valores de velocidad en las tuberías, apareciendo v_j como variable asociada a cada arco de la red, sin embargo, el modelo tradicional no incluye esta restricción.

El modelo es el siguiente:

$$\text{Min Coste}_{Total} = f(\phi_1, \phi_2, \dots, \phi_A) = \sum_{j \in A} C(\phi_{kj}) \cdot L_j \quad (\text{ec. 1})$$

sa:

$$\sum_{\forall j \in E_i} q_j - \sum_{\forall j \in S_i} q_j = d_i \quad \forall i = 1 \dots N \quad (\text{ec. 2})$$

$$\sum_{\forall j \in M} H_j = 0 \quad \forall m = 1 \dots M \quad (\text{ec. 3})$$

$$h_i = h_{i-1} - H_j \quad \forall i = 2 \dots N, j \in E_i \quad (\text{ec. 4})$$

$$h_i \geq h_i^{min} \quad \forall i = 2 \dots N \quad (\text{ec. 5})$$

$$\phi_{kj} \in K ; q_j \in R ; h_i \in R \quad (\text{ec. 6})$$

La función objetivo depende los diámetros asignados a las distintas secciones o tuberías. Como se muestra en la ecuación 1, esta función es el coste total de red definido como la suma del coste unitario del tipo de diámetro k asignado a cada tramo j, $c_k(\phi_{kj})$, multiplicado por la longitud de dicho tramo, L_j .

El primer conjunto de restricciones, ecuación 2, asegura el cumplimiento de la ley de conservación de masa en el sistema. La cantidad de agua que llega a un nodo menos la que lo abandona debe ser igual a la demanda de dicho nodo.

El segundo grupo de restricciones, ecuación 3, representa la conservación de energía del sistema forzando que la caída de presión en cada malla sea igual a 0. Además, la presión en cada nodo de la red debe ser igual o mayor a la mínima presión de servicio establecida, h_i^{min} , ecuación 5. Para ello se calcula la presión en cada nodo como la presión en el nodo anterior menos la caída de presión en el arco que une dichos nodos, ecuación 4. Cabe destacar que no existe una única forma de calcular la presión en un nodo dado la tipología mallada de la red.

Por último, se define la naturaleza de las variables en la ecuación 6, siendo los diámetros variables discretas y pertenecientes al conjunto de diámetros comerciales K. Tanto los caudales en los arcos q_j , como las presiones en los nodos h_i , son variables que toman valores del conjunto de números reales.

Para calcular las caídas de presión H_j es necesario establecer un criterio o método de cálculo. Existen diferentes expresiones utilizadas para el cálculo de estas caídas de presión pero sin duda la expresión empírica más utilizada en la literatura es la expresión de Hazen-Williams.

$$H_j = K_j \cdot q_j^p \quad (\text{ec. 7})$$

Dónde:

H_j : Representa la caída de presión en el arco j.

K_j : Resistencia hidráulica de la tubería j.

q_j : Caudal que atraviesa el arco j.

$$K_j = \alpha \frac{L_j}{C^p \cdot \phi_j^\gamma} \quad (ec. 8)$$

α : Coeficiente que depende de las unidades utilizadas.

p, γ : Exponentes experimentales que toman los siguientes valores:

$$p = 1,852 \quad (ec. 9)$$

$$\gamma = 4,87 \quad (ec. 10)$$

C : Coeficiente de Hazen-Williams, es adimensional y depende de la rugosidad del material de fabricación de la tubería.

ϕ_j : Diámetro de la sección j.

L_j : Longitud de la sección j.

Por lo que la pérdida de presión en una sección o tubería se calculará a través de la expresión de Hazen-Williams de la siguiente forma:

$$H_j = \alpha \cdot \frac{L_j \cdot q_j^{1,852}}{C^{1,852} \cdot \phi_j^{4,87}} \quad (ec. 11)$$

Al introducir esta definición de la caída de presión en la ecuación 3 del modelo éste pasa a contener un sistema de ecuaciones no lineales. Esto añadido a la naturaleza discreta de las variables 'diámetros' hace que el problema de diseño de redes hidráulicas de distribución sea un problema NP-duro.

Existen otras expresiones utilizadas para el cálculo de la caída de presión en redes hidráulicas de distribución. Una muy famosa es la ecuación empírica de Darcy-Weisbach:

$$H_j = f \cdot \frac{L_j \cdot v_j^2}{2g \cdot \phi_j} \quad (ec. 12)$$

Dónde:

H_j : Caída de presión entre los nodos i y j.

f : Factor adimensional de fricción de Darcy.

\varnothing_j : Diámetro de la sección j .

L_j : Longitud de la sección j .

v_j : Velocidad media del agua en la sección j .

g : Aceleración debida a la gravedad, con un valor de $9,81m/s^2$.

El factor de fricción varía según la rugosidad y el diámetro de la tubería, además del tipo de fluido (Número de Reynold). Se puede calcular analíticamente, o gráficamente a través del diagrama de Moody que se muestra en la siguiente figura.

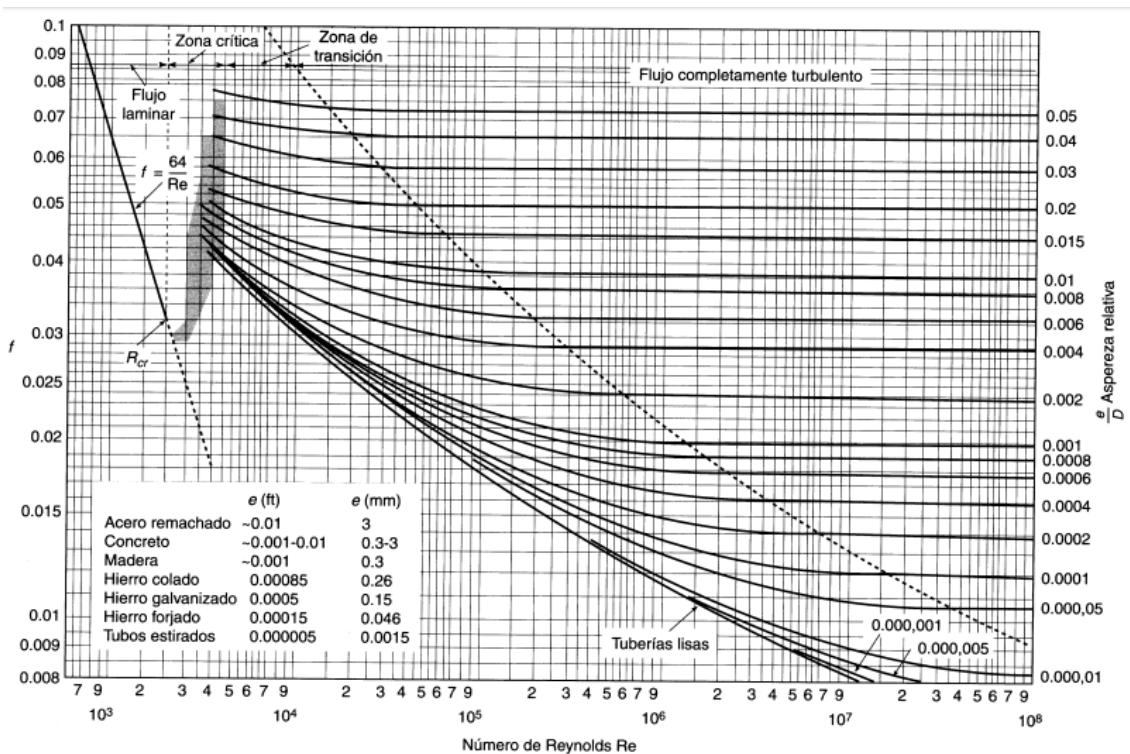


Figura 15 - Diagrama de Moody
Fuente: Moody, 1944

En la gráfica de la figura 15 se relaciona el coeficiente de fricción con el número de Reynolds y con la aspereza relativa. El número de Reynolds es un factor adimensional que caracteriza el tipo de fluido según su densidad, velocidad y la viscosidad, y del diámetro de la tubería por la que éste circula.

$$Re = \frac{\rho \cdot v \cdot \varnothing}{\mu} \quad (ec. 13)$$

Dónde:

ρ : Densidad del fluido

v : Velocidad media del fluido

\emptyset : Diámetro de la tubería

μ : Viscosidad del fluido.

El cálculo del número de Reynolds se muestra en la ecuación 13, y éste se caracteriza por ser directamente proporcional a la densidad y velocidad del fluido, además, de al diámetro de la tubería por la que éste circula. E inversamente proporcional a la viscosidad del fluido.

Para una tubería con diámetro \emptyset , el flujo es laminar si $Re < 2300$, turbulento si $Re > 4000$ y si $2300 < Re < 4000$ existe una zona de transición.

4.3. Enfoques para la resolución del problema

La resolución de un problema de optimización no es más que encontrar el valor que deben tomar las variables para hacer óptima la función objetivo a la vez que se satisfacen las restricciones. Dependiendo de la naturaleza de la función objetivo, las variables y las restricciones de un problema, éste se resolverá de una manera u otra.

Según la complejidad del modelo éste se resolverá mediante los denominados “métodos clásicos” o “metaheurísticas”. Los métodos clásicos abarcan la programación lineal, la no lineal, la programación entera mixta entre otros. Las metaheurísticas surgen para resolver problemas de un grado de complejidad alto a través de procedimientos sencillos que proveen soluciones satisfactorias pero no necesariamente la óptima en relativamente poco tiempo.

Unos de los primeros autores en abordar la resolución del problema de diseño de redes de distribución de agua fueron Alperovits and Shamir (1977) mediante el método del gradiente de programación lineal (LPG). Este método descompone el problema no lineal original en dos subproblemas, uno lineal y otro no lineal. Es un método iterativo donde en cada iteración se fijan los caudales y las tuberías son divididas en segmentos, a cada segmento se le asignará un diámetro diferente. Las variables de este subproblema son la longitud de dichos segmentos y se resuelve mediante programación lineal. Posteriormente en cada iteración los flujos o caudales son heurísticamente ajustados de acuerdo al gradiente de la función objetivo, subproblema no lineal. El procedimiento se repite hasta encontrar un óptimo local. Otros autores que investigaron en la misma línea fueron Kessler, A. and Shamir (1989).

Las soluciones alcanzadas mediante este procedimiento no son realistas ya que algunos segmentos toman un diámetro para el 99% de su longitud y otro diferente para el resto.

Otros autores aplicaron la programación lineal tomando los diámetros como variables continuas que, una vez resuelto el problema, se transforman en los diámetros de la gama comercial más cercanos a los valores de dichas variables continuas [8][14]. El problema surge al transformar los diámetros ya que puede perderse la optimalidad.

Ante esta problemática las metaheurísticas aplicadas al diseño de redes hidráulicas de distribución surgieron como métodos capaces de tratar los diámetros como variables

discretas y por ello resolver de forma más realista este problema no lineal mixto y entero.

En general, una metaheurística imita un comportamiento sencillo de la naturaleza. Estas técnicas se pueden clasificar según estén basadas en poblaciones, como son los algoritmos genéticos, la optimización por enjambre de partículas o los algoritmos meméticos, o basadas en soluciones, como el recocido simulado o la búsqueda tabú. Los algoritmos genéticos están inspirados en el proceso de evolución natural de los seres vivos. El método de optimización por enjambre de partículas imita el comportamiento de ciertas especies en las que existe un líder que marca los movimientos del grupo. Los algoritmos meméticos añaden la búsqueda local a los algoritmos basados en la evolución. Dentro de las metaheurísticas basadas en las soluciones, el recocido simulado se basa en el enfriamiento de los materiales o recocido, la convergencia al óptimo global de este método ha sido teóricamente demostrada. Y, por último, la utilizada en este trabajo, búsqueda tabú, se inspira en la tendencia inconsciente humana a desviarse del curso trazado.

En los últimos años varios autores han aplicado dichas técnicas al problema de diseño y planificación de redes de distribución. Cunha and Sousa (1999, 2001) aplicaron el algoritmo de recocido simulado y algoritmos genéticos alcanzando mejoras significativas en la calidad de la solución y en los tiempos de resolución. Suribabu and Neelakantan (2006) abordaron el problema aplicando el algoritmo de optimización por enjambre de partículas y utilizando un modelo con costes de penalización. Meirelles et al. (2018) aplica este mismo algoritmo pero añade, al objetivo general de minimizar los costes de instalación, los beneficios de la producción energética que supondría instalar recuperadores en la red. Esto cobra sentido al tratarse de un terreno con grandes desniveles.

Matías (2003) utilizó el modelo con costes de penalización por fiabilidad y aplicó algoritmos genéticos para resolver el problema. Este autor alcanzó resultados más precisos utilizando la herramienta EPANET 2, debido a los avances del programa y la precisión de los coeficientes de Hazen-Williams utilizados. Más tarde, Afshar (2006) aplicó el algoritmo basado en el comportamiento de colonias de hormigas para resolver el diseño de una red de distribución con el mismo modelo que incluye la fiabilidad de la misma.

En este trabajo el problema se resuelve aplicando la búsqueda tabú. La resolución del problema mediante esta metaheurística se ha tratado en la literatura científica únicamente por Cunha and Ribeiro (2004). Esta metaheurística permite aplicar diferentes criterios que se explican en la sección 5.2. En el presente trabajo se añaden nuevas características a la búsqueda tabú aplicada por estos autores.

Una vez descritos las distintas técnicas aplicadas a la problemática de estudio, cabe destacar que no existen pruebas de que ninguna de las soluciones alcanzadas aplicando estas técnicas proporcione el óptimo global del problema. Por lo que, la aplicación de nuevos métodos para resolver el problema de diseño de redes de distribución cobra un importante valor.

5. Resolución del caso de estudio mediante Búsqueda Tabú

En este trabajo se ha resuelto el problema de diseño de redes hidráulicas de distribución mediante el algoritmo de búsqueda tabú y usando el modelo sin costes de penalización. Como ya se ha comentado el problema es de naturaleza NP-duro, por ello el uso de metaheurísticas es el único camino para hallar soluciones realistas.

El problema se resuelve usando el software libre Python en la versión 3.6 [21]. Para validar el algoritmo se utiliza la red de Alperovits lo cual permite comparar los resultados obtenidos con los de otros autores y así comprobar la eficacia del algoritmo.

En la siguiente sección 5.1 se particulariza el modelo sin costes de penalización para la red de Alperovits, los datos, variables y restricciones, y las unidades de medida utilizadas. A continuación, en la sección 5.2, se explica detalladamente las características de la metaheurística desarrollada. Los módulos de Python utilizados y las funciones creadas para resolver de manera eficiente y clara el problema, se definen en la sección 5.3. Por último se presenta el pseudocódigo que contiene las diferentes etapas del código.

5.1. Particularización del modelo

En este apartado se muestra la adaptación del modelo general, desarrollado en la sección 4.2, a la red de Alperovits. Para ello, se sustituyen los datos propios de la red con las unidades requeridas por el algoritmo de resolución codificado en Python. Se muestra de nuevo la figura de la red para facilitar la comprensión de los datos y las restricciones del modelo adaptado.

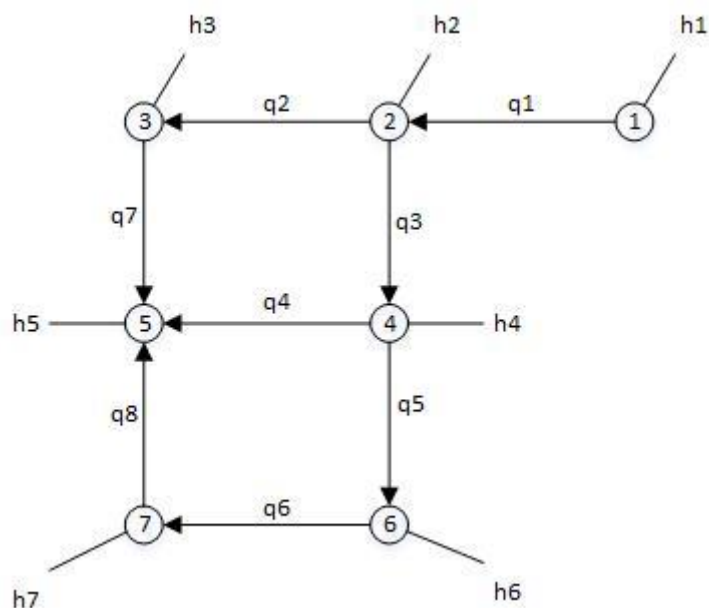


Figura 16 - Variables secundarias– Red de Alperovits
Fuente: Elaboración propia

En la figura 16 se han representado los nodos y los arcos que forman la red junto con las variables secundarias del modelo, caudales de agua por los arcos y presiones en los nodos.

La tabla 9 recoge todos los datos propios de la red que sirven para caracterizar su estructura y las ecuaciones del modelo, además las unidades de medida de cada dato. La tabla 10 recoge la longitud de los arcos en metros.

Tabla 9 - Datos y notación de los nodos - Red de Alperovits

Nodo	Coordenadas	Valor [m]	Elevación	Valor [m]	Demanda	Valor [m^3/s]	Presión mínima	Valor [m.c.a.]
1	(x_1, y_1)	(2000,2000)	E_1	210	d_1	-0,311	h_1^{min}	-
2	(x_2, y_2)	(1000,2000)	E_2	150	d_2	0,028	h_2^{min}	30
3	(x_3, y_3)	(0,2000)	E_3	160	d_3	0,028	h_3^{min}	30
4	(x_4, y_4)	(1000,1000)	E_4	165	d_4	0,033	h_4^{min}	30
5	(x_5, y_5)	(0,1000)	E_5	155	d_5	0,075	h_5^{min}	30
6	(x_6, y_6)	(1000,0)	E_6	165	d_6	0,092	h_6^{min}	30
7	(x_7, y_7)	(0,0)	E_7	160	d_7	0,056	h_7^{min}	30

Tabla 10 - Datos y notación de los arcos - Red de Alperovits

Arco	Longitud	Valor
		[m]
1	L_1	1000
2	L_2	1000
3	L_3	1000
4	L_4	1000
5	L_5	1000
6	L_6	1000
7	L_7	1000
8	L_8	1000

Alperovits and Shamir (1997) en su trabajo exponen una gama comercial de 14 de diámetro medidos en pulgadas (in) y su coste unitario por unidad de longitud correspondiente. Se muestra la conversión a unidades del sistema internacional (m) ya que el simulador utilizado para resolver el modelo requiere dichas unidades.

Tabla 11 - Datos diámetros comerciales - Red de Alperovits

Diámetro	Valor [m]	Coste(\emptyset_k)	Diámetro	Valor [m]	Coste(\emptyset_k)
		$[\frac{u.m.}{u.L.}]$			$[\frac{u.m.}{u.L.}]$
\emptyset_1	0,0254	2	\emptyset_8	0,3048	50
\emptyset_2	0,0508	5	\emptyset_9	0,3556	60
\emptyset_3	0,0762	8	\emptyset_{10}	0,4064	90
\emptyset_4	0,1016	11	\emptyset_{11}	0,4572	130
\emptyset_5	0,1524	16	\emptyset_{12}	0,5080	170
\emptyset_6	0,2032	23	\emptyset_{13}	0,5588	300
\emptyset_7	0,2540	32	\emptyset_{14}	0,6096	550

En la tabla 11 se recogen el valor y los costes unitarios asociados a cada diámetro comercial. La gama de diámetros obliga a que ningún tramo pueda ser eliminado, como mínimo el diámetro de un tramo será de 0,0254m o lo que es lo mismo 2,54cm.

Antes de formular el modelo hay que definir los valores del coeficiente, α y la rugosidad C usados para el cálculo de las caídas de presiones mediante la ecuación de Hazen-Williams. Estos valores se recogen en la tabla 12.

Tabla 12 - Coeficientes de Hazen-Williams para la red de Alperovits

Coeficiente	Valor
α	10,667
C	130

El valor del coeficiente α es definido en el paquete de simulación hidráulica de Python debido a las unidades que éste utiliza. El coeficiente de rugosidad de Hazen-Williams se fija en el trabajo de Alperovits en 130 [7]. Ambos coeficientes son adimensionales.

Modelo

Función objetivo:

$$\text{Min } [1000 \cdot C(\phi_{k1}) + C(\phi_{k2}) + C(\phi_{k3}) + C(\phi_{k4}) + C(\phi_{k5}) + C(\phi_{k6}) + C(\phi_{k7}) + C(\phi_{k8})] \quad (\text{ec. 14})$$

sa:

- *Ecuaciones de conservación de masa:*

$$q_1 = -0,311 \quad (\text{ec. 15})$$

$$q_1 - q_2 - q_3 = 0,028 \quad (\text{ec. 16})$$

$$q_2 - q_7 = 0,028 \quad (\text{ec. 17})$$

$$q_3 - q_4 - q_5 = 0,033 \quad (\text{ec. 18})$$

$$q_7 + q_4 - q_8 = 0,075 \quad (\text{ec. 19})$$

$$q_5 - q_6 = 0,092 \quad (\text{ec. 20})$$

$$q_6 - q_8 = 0,056 \quad (\text{ec. 21})$$

- *Ecuaciones de conservación de energía:*

$$\frac{10,667 \cdot 1000}{130^{1,852}} \cdot \left[\frac{q_3^{1,852}}{\phi_{k3}^{4,87}} + \frac{q_4^{1,852}}{\phi_{k4}^{4,87}} - \frac{q_7^{1,852}}{\phi_{k7}^{4,87}} - \frac{q_2^{1,852}}{\phi_{k2}^{4,87}} \right] = 0 \quad (\text{ec. 22})$$

$$\frac{10,667 \cdot 1000}{130^{1,852}} \cdot \left[\frac{q_5^{1,852}}{\phi_{k5}^{4,87}} + \frac{q_6^{1,852}}{\phi_{k6}^{4,87}} + \frac{q_8^{1,852}}{\phi_{k8}^{4,87}} - \frac{q_4^{1,852}}{\phi_{k4}^{4,87}} \right] = 0 \quad (\text{ec. 23})$$

- *Presiones en los nodos:*

$$h_1 = 210 \quad (\text{ec. 24})$$

$$h_2 = 210 - 10,667 \cdot \frac{1000 \cdot q_1^{1,852}}{130^{1,852} \cdot \phi_1^{4,87}} \quad (\text{ec. 25})$$

$$h_3 = h_2 - 10,667 \cdot \frac{1000 \cdot q_2^{1,852}}{130^{1,852} \cdot \phi_2^{4,87}} \quad (\text{ec. 26})$$

$$h_4 = h_2 - 10,667 \cdot \frac{1000 \cdot q_3^{1,852}}{130^{1,852} \cdot \phi_3^{4,87}} \quad (\text{ec. 27})$$

$$h_5 = h_4 - 10,667 \cdot \frac{1000 \cdot q_4^{1,852}}{130^{1,852} \cdot \emptyset_4^{4,87}} \quad (ec. 28)$$

$$h_6 = h_4 - 10,667 \cdot \frac{1000 \cdot q_5^{1,852}}{130^{1,852} \cdot \emptyset_5^{4,87}} \quad (ec. 29)$$

$$h_7 = h_6 - 10,667 \cdot \frac{1000 \cdot q_6^{1,852}}{130^{1,852} \cdot \emptyset_6^{4,87}} \quad (ec. 30)$$

- *Requisitos de presiones en los nodos:*

$$h_2 \geq 30 \quad (ec. 31)$$

$$h_2 \geq 30 \quad (ec. 32)$$

$$h_2 \geq 30 \quad (ec. 33)$$

$$h_2 \geq 30 \quad (ec. 34)$$

$$h_2 \geq 30 \quad (ec. 35)$$

$$h_2 \geq 30 \quad (ec. 36)$$

- *Variables:*

$$\emptyset_{k1}, \emptyset_{k2}, \emptyset_{k3}, \emptyset_{k4}, \emptyset_{k5}, \emptyset_{k6}, \emptyset_{k7}, \emptyset_{k8} \in K$$

$$q_2, q_3, q_4, q_5, q_6, q_7, q_8 \in R$$

$$h_2, h_3, h_4, h_5, h_6, h_7 \in R$$

Hay un total de 21 variables, los 8 diámetros que son variables discretas que deben tomar valores de entre un conjunto fijo. Las variables continuas referentes a los caudales en los arcos, en total 7 variables, pues el caudal del arco 1 es conocido e igual al sumatorio de la demanda de los nodos consumidores. Y 6 variables que representan las presiones en los nodos, siendo también variables continuas. La presión en el nodo fuente, h_1 , es conocida.

La función objetivo depende los diámetros asignados a las distintas secciones o tuberías. Como se muestra en la ecuación 1, esta función es el coste total de la red definido como el sumatorio del coste unitario del tipo de diámetro k asignado al tramo j , $C(\emptyset_{kj})$ multiplicado por la longitud de dicho tramo, L_j . Dado, que todos los tramos tienen una longitud de 1000m, este valor sale como factor común en la ecuación.

La conservación de masa se impone en las ecuaciones 15-21, existiendo una restricción para cada nodo de la red. Al existir un solo nodo fuente, el caudal del mismo será la suma de las demandas de todos los nodos consumidores, ecuación 15.

Para la conservación de energía existe una restricción para cada malla, ecuaciones 22 y 23. Por último, se impone que la presión en cada nodo sea igual o mayor a la mínima presión de servicio establecida, 30 m.c.a., por ello es necesario calcular la presión en cada nodo como la presión en el nodo anterior menos la caída de presión del arco que une dichos nodos. En las ecuaciones 24-30 se calculan las presiones en cada nodo a partir del nodo anterior, no obstante, al ser una red mallada, cada nodo está conectado a más de un arco,

por ello, no existe una única forma de calcular dicha presión. En las ecuaciones 24-30 se ha desarrollado el cálculo para uno de los posibles caminos.

5.2. Búsqueda Tabú

La Búsqueda Tabú, Tabu Search (TS) en inglés, es una metaheurística que surgió en 1986 por el científico Fred Glover debido a la necesidad de procedimientos rápidos e iterativos para la resolución de problemas complejos de programación entera. El mayor reto al que se enfrentan las metaheurísticas son los óptimos locales, por lo que generar un método capaz de escapar de los mismos representó un gran avance.

El algoritmo se basa en el comportamiento de la memoria del ser humano, y especialmente, en su tendencia a desviarse de un curso trazado. Esta tendencia puede verse como un error, pero en muchos casos conlleva a una mejoría. De hecho, es un mecanismo útil, incluso fundamental para el ingenio humano.

Partiendo de una solución inicial se construye un vecindario de soluciones adyacentes que pueden ser alcanzadas desde la solución actual. La característica principal de esta metaheurística es la prohibición o penalización de ciertos movimientos, lo que reduce el vecindario en cada iteración. Dichos movimientos son marcados como movimientos “tabú” y permanecerán durante un número de iteraciones con dicho atributo. Una vez ese número de iteraciones, inicialmente fijado, ha transcurrido, el movimiento se desprende del atributo “tabú” y puede ser ejecutado de nuevo sin penalización alguna. Este mecanismo recibe el nombre de memoria a corto plazo y su objetivo es evitar los ciclos en la búsqueda de soluciones. [Glover, 1986]

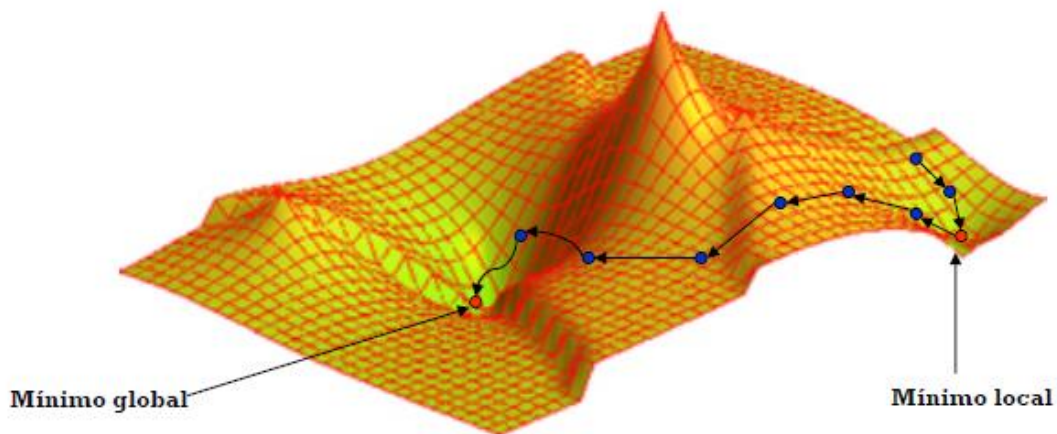


Figura 17 - Representación de búsqueda del mínimo global
Fuente: A. Escudero y P. Aparicio, 2018

En la figura 17 se observa el recorrido que debe seguir la búsqueda tabú para hallar la mejor solución, u óptimo global, del problema.

La búsqueda tabú explora el espacio de búsqueda con mecanismos como la memoria flexible, restricciones estratégicas o niveles de aspiración con el fin de escapar de los óptimos locales y alcanzar el óptimo global. Para ello se permite el empeoramiento de la solución como mecanismo para alcanzar una solución final mejor, en la figura 17 se muestra gráficamente este mecanismo.

Se definen las principales características del método:

- Entorno reducido: al prohibirse ciertos movimientos etiquetados como movimientos tabú, el entorno o vecindario de cada solución se ve reducido y por lo tanto el tiempo total de resolución disminuirá. Es importante fijar el número de iteraciones que un movimiento permanece con el atributo tabú ya que esto influye bastante en los resultados.
- Nivel de aspiración: condición que al satisfacerse permite alcanzar una solución, aunque el movimiento que lleva a dicha solución tenga el atributo tabú.
- Memoria a corto plazo: almacena características de las soluciones recientes, por ejemplo, el movimiento en la solución anterior que llevó a la solución actual, para asignarle el atributo tabú y así explorar a fondo una sección del espacio de soluciones.
- Memoria a largo plazo: el objetivo es intensificar y diversificar la búsqueda almacenando frecuencia de ocurrencias de ciertas características de las soluciones y aplicando criterios para acotar estas ocurrencias.

Estas características son las más diferenciadoras de la búsqueda tabú con respecto a otras metaheurísticas. No obstante, el algoritmo puede implementarse incluyendo todas o bien solo alguna de las características anteriores, cada problema tendrá unos requisitos diferentes.

A continuación, se procede a explicar mediante un diagrama de flujo el algoritmo de búsqueda tabú diseñado para la resolución de la problemática de estudio.

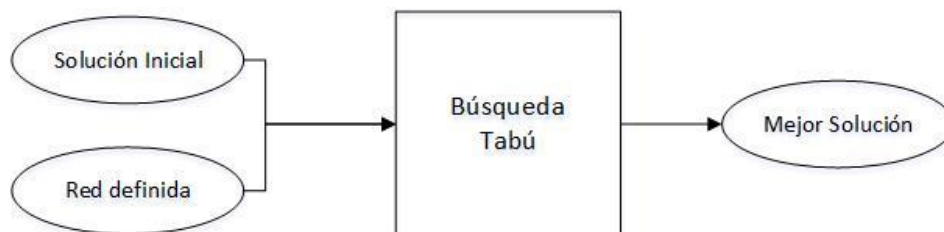


Figura 18 - Entradas y Salidas de la Búsqueda Tabú
Fuente: Elaboración propia

En primer lugar, el algoritmo requiere dos entradas: la solución inicial y la red de estudio definida, como muestra la figura 18. La solución inicial queda definida por el conjunto de diámetros asignados a cada arco de la red. La red de estudio, en este caso red de Alperovits, se define mediante un paquete de Python, *WNRT*, explicado con más detalle en la siguiente sección. Una vez aplicada la búsqueda tabú se obtiene la mejor solución, conjunto de diámetros que minimizan los costes de la red, y el coste de dicha mejor solución. A través de este coste se comparan el resto de métodos con el aquí mostrado.

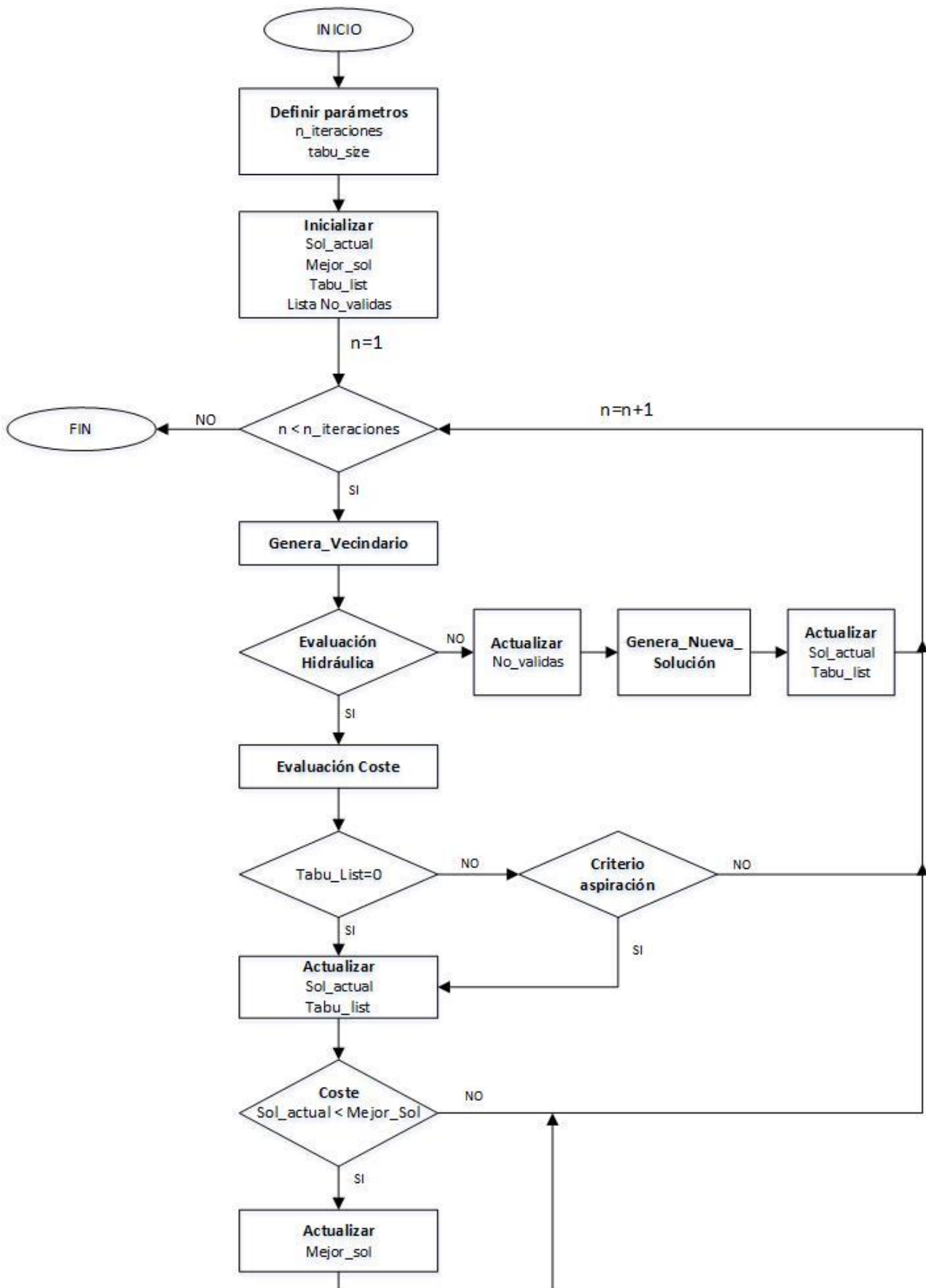


Figura 19 - Diagrama de flujo - Búsqueda Tabú
Fuente: Elaboración propia

En el diagrama de flujo de la figura 19 se muestran los pasos que sigue el algoritmo de búsqueda tabú aplicado al problema en cuestión. Se procede a describir con más detalles dichos pasos:

- **Definir parámetros**

- `n_iteraciones`: es el número de veces que se genera un vecindario.
- `Tabu_size`: es el número de iteraciones que un movimiento permanece en la lista tabú, es decir, con el atributo tabú explicado anteriormente.

Ambos parámetros tomarán distintos valores en las diferentes simulaciones con objeto de encontrar una combinación mediante la cual se alcance una solución mejor en menos tiempo.

- **Inicializar**

- `Sol_actual`: la solución actual se inicializa como la solución inicial de entrada al algoritmo. Este parámetro guarda en todo momento la solución actual del algoritmo. Además, en cada iteración se guarda el coste de dicha solución actual en una lista para, una vez realizadas todas las iteraciones, representar la evolución que experimenta.
- `Mejor_sol`: la mejor solución se inicializa como la solución inicial y en cada iteración se actualizará siempre y cuando la solución actual tenga un coste menor que ésta. Por lo que el parámetro `Mejor_sol` guarda la mejor solución encontrada hasta el momento. Al igual que para la solución actual, el coste de la mejor solución también se almacena en una lista para representar la evolución que experimenta.
- `Tabu_list`: tiene la misma forma que la solución, es decir, si la solución es una lista de 4 elementos la lista tabú será una lista de 4 elementos. Inicialmente, todos los valores de la lista tabú toman el valor de 0, y en cada iteración se incrementará en `Tabu_size` el elemento que ha sido modificado de la solución en dicha iteración. La lista tabú es el mecanismo de memoria a corto plazo, ya que guarda durante cierto número de iteraciones información de los movimientos anteriores para evitar repetirlos y así no caer en bucles.

- **Genera_Vecindario**

La estructura del vecindario es uno de los aspectos más importantes del algoritmo. La generación del vecindario se lleva a cabo a través una función llamada "Reduce_Diámetro" la cual, teniendo como entrada la solución actual, devuelve

todas las soluciones originadas al reducir al siguiente diámetro menor de la gama comercial cada uno de los diámetros que forman dicha solución.

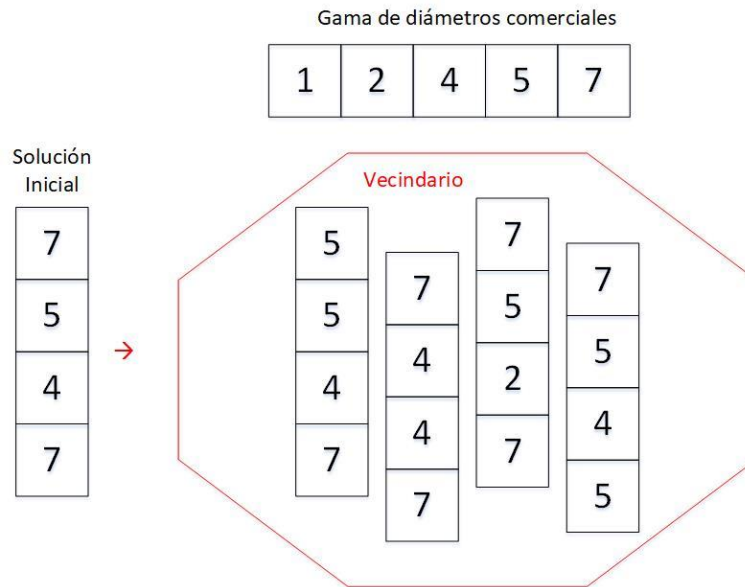


Figura 20 - Mecanismo de generación de vecindario
Fuente: Elaboración propia

En la figura 20 se representa a través del ejemplo de una red con cuatro arcos o tuberías, el mecanismo de generación de vecindario. En este ejemplo la gama de diámetros comerciales cuenta con cinco diámetros de dimensiones: 1, 2, 4, 5 y 7 u.L. Se observa que para una solución inicial el vecindario son cuatro soluciones vecinas, en cada una de ellas se ha disminuido un diámetro.

Para el caso concreto de la red de Alperovits, al contar ésta con ocho arcos, una solución está constituida por una lista de ocho diámetros. El vecindario de dicha solución será un conjunto de ocho soluciones diferentes en las que en cada una se ha reducido uno de los ocho diámetros, y el resto se ha mantenido constante.

- **Evaluación hidráulica**

Cada solución vecina se analiza hidráulicamente a través de un simulador del paquete WNTR que reproduce las funciones del software EPANET. Este simulador calcula los caudales en los arcos y las presiones en los nodos aplicando el método de Newton Rapson al sistema de ecuaciones de conservación de masa y energía que rigen el funcionamiento de la red.

El requisito para que una solución apruebe la validez hidráulica es que las presiones en todos los nodos, excepto en el nodo fuente, sean mayores a 30 m.c.a.

- **Genera_Nueva_Solución**

Este mecanismo sólo se lleva a cabo en el caso de que ninguno de los vecinos de una solución sea hidráulicamente válido. Para escapar de un bucle en el que el algoritmo no avanzaría, se decide aumentar un diámetro, aunque esto suponga un aumento del coste de la solución actual. Para la elección del diámetro a aumentar

se escoge al azar entre el conjunto de diámetros cuyo atributo Tabú sea nulo en dicha iteración.

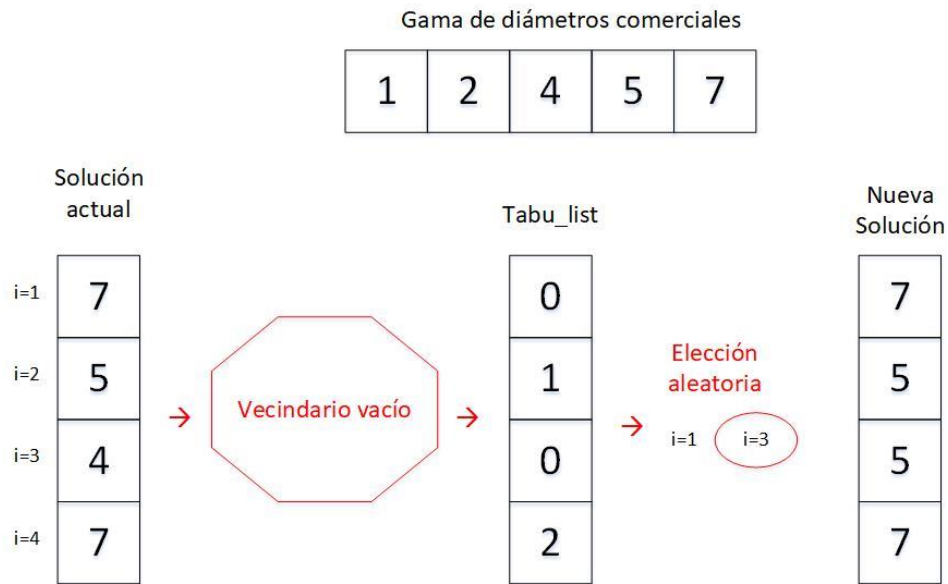


Figura 21 - Mecanismo generación de nuevas soluciones
Fuente: Elaboración propia

En la figura 21 se presenta el mecanismo de generación de nuevas soluciones, para el supuesto de que el vecindario de una solución actual esté vacío. En el ejemplo de la figura, los únicos diámetros con estado tabú igual a 0 son los diámetros de los arcos i igual a 1 e i igual a 3, se escoge aleatoriamente uno de ellos y se aumenta su diámetro.

Una vez generada esta nueva solución se actualiza la solución actual, Sol_actual , y la lista tabú evitando que en las próximas iteraciones se vuelva a disminuir el mismo diámetro que se ha aumentado.

• Evaluación Coste

Se evalúa el coste de las soluciones vecinas que son válidas hidráulicamente. Este coste se calcula como el coste unitario del diámetro escogido para cada arco por su longitud. Una vez calculado dicho valor se pueden comparar soluciones y así poder escoger la de menor coste total.

A continuación, se comprueba si la solución con menor coste del vecindario tiene estado tabú. En el caso de que el estado tabú sea 0, se actualiza la Sol_actual , el coste de la solución actual y la lista tabú, aumentando en $Tabu_size$ el elemento que se modifica para alcanzar dicha solución. Si la solución actual tuviese un coste menor al coste de la mejor solución, ésta también se actualizaría. Si por el contrario, la mejor solución del vecindario tiene estado tabú, se aplica el criterio de aspiración.

- **Criterio de aspiración**

El criterio de aspiración permite asignar como solución actual una solución del vecindario con estado tabú distinto de 0, siempre y cuando el coste de ésta sea menor al coste de la mejor solución encontrada hasta el momento.

5.3. Python

Python es un lenguaje de programación de gran capacidad, su mayor interés recae en la gran variedad de librerías que contiene y las funciones incorporadas en el propio lenguaje, además su filosofía provee una sintaxis de fácil lectura. Posee licencia de código abierto, es decir, es gratuito. Es un lenguaje interpretado, no necesita compilar el código fuente para poder interpretarlo. En este trabajo se utiliza la última versión 3.6.4 disponible en la web [21].

Python puede trabajar sin entorno de desarrollo, no obstante, es más fácil utilizar un entorno que analiza el código advirtiéndolo de posibles fallos en el mismo, además, permite localizar errores fácilmente. El entorno de desarrollo integrado (IDE) utilizado es Pycharm desarrollado por la compañía JetBrains, es un entorno diseñado específicamente para la programación en lenguaje Python [24].



Figura 22 - Logos software Python - IDE Pycharm

Fuente: <https://www.python.org/>

En las figuras 22 se muestran los logos tanto de Python, como del entorno de desarrollo utilizado, Pycharm.

Además de la librería estándar de Python, existen muchas otras librerías que pueden ser importadas en el programa y permiten ejecutar tareas específicas. En el siguiente apartado se explican todas las librerías utilizadas en el código y sus características principales.

5.3.1. Librerías

Una librería en Python no es más que un código que almacena variables, métodos, clases y atributos y realiza funciones específicas. Por sencillez, se llamará librería a la librería estándar de Python y paquete al resto de librerías.

```

import wntr
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from time import time
import random

```

Las líneas de código muestran los seis paquetes importados en Python para la resolución del problema. Se definen las funciones principales de cada paquete. El primer paquete, WNTR, se explica con más detalle en el siguiente apartado.

- **WNTR** (Water Network Tool for Resilience): es un paquete de Python que reproduce las funciones del software EPANET e incluye otras opciones para simular y analizar redes hidráulicas de distribución [25][26]. Este paquete se ha utilizado para realizar la comprobación técnica de cada solución, evaluación hidráulica. Se explica con mayor detalle más adelante.
- **Panda**: permite el análisis de datos de forma rápida y eficaz, es utilizado dentro del paquete WNTR.
- **Numpy**: permite trabajar con vectores y matrices de manera fácil y rápida.
- **Matplotlib**: permite la generación de gráficos a través de datos contenidos en listas o matrices.
- **Time**: pertenece a la librería estándar de Python y permite trabajar con periodos de tiempo, fechas, etc. Se ha utilizado para controlar el tiempo de ejecución del código.
- **Random**: proporciona un generador de números aleatorios, también pertenece a la librería estándar de Python.

5.3.2. Paquete WNTR

En este apartado se describe la estructura del paquete WNTR y el uso que se ha hecho de éste para lograr una mejor comprensión del código.

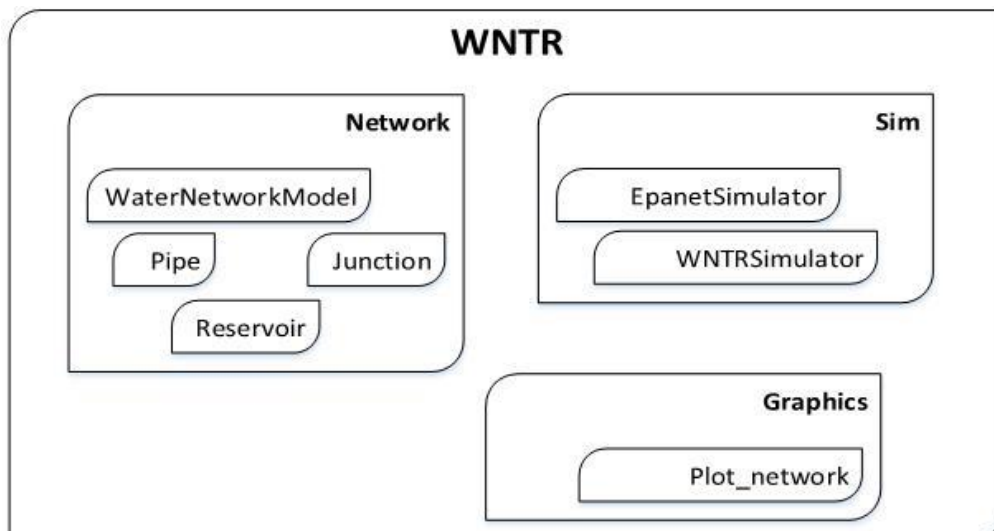


Figura 23 - Estructura del paquete WNTR de Python
Fuente: Elaboración propia

El paquete contiene varios módulos, y esto a su vez distintas clases. En la figura 23 se representa la estructura del paquete WNTR con alguno de sus módulos utilizados en este trabajo.

- **Módulo *Network***

Contiene clases o métodos que permiten definir una red hidráulica. En la figura 23 solo se muestran alguno de los métodos utilizados en el código, pero existen muchos más. Por ejemplo, *Junction*, permite la definición de los nodos de la red a través de su demanda, elevación y coordenadas. Con la clase *Pipe* se definen los arcos o tuberías mediante su longitud, diámetro, rugosidad, pérdidas y estado (abierto o cerrado), permitiendo modificar los atributos de la red a lo largo del código. *Reservoir* se utiliza para generar los depósitos o embalses que forman parte de la red, esta clase tan solo requiere la altura y coordenadas del elemento.

- **Módulo *Sim***

Este módulo contiene únicamente dos métodos, representados en la figura 23, ambos para ejecutar simulaciones hidráulicas usando modelos de redes. El método escogido, *EpanetSimulator*, permite ejecutar simulaciones hidráulicas impulsadas por la demanda, es decir, la presión en el sistema depende de la demanda de los nodos consumidores. *EpanetSimulator* utiliza las mismas ecuaciones que el software EPANET para resolver el sistema de ecuaciones.

EPANET es un software libre desarrollado por la agencia de protección medioambiental de los Estados Unidos (USEPA) para la simulación de redes hidráulicas de distribución [27]. Con el fin de hacer accesible las funciones del software a diversos usos y aplicaciones la agencia ha creado una librería llamada EPANET Toolkit que contiene funciones que permiten simular el comportamiento de EPANET y servir como herramienta de programación para distintos lenguajes.

El método *EpanetSimulator* incluye funciones de dicha librería permitiendo la resolución de redes hidráulicas. La herramienta EPANET Toolkit resuelve el sistema de ecuaciones no lineales formado por las ecuaciones de conservación de masa y energía usando el método de Newton-Rapson. Éste utiliza la interpretación geométrica de la derivada de la función como mecanismo para hallar la solución simultánea del sistema de ecuaciones de conservación de masa y energía, en la parte lineal y no-lineal. Finalmente se obtienen los valores de presión en los nodos y de caudales en las tuberías de la red.

Cabe destacar, que las caídas de presión son calculadas mediante la expresión de Hazen-Williams.

- **Módulo *Graphics***

Este módulo, con el soporte del paquete Matplotlib, permite crear gráficas para visualizar la red y sus características una vez resuelta.

El paquete WNTR utiliza las unidades de medida del sistema internacional (SI). Es importante tenerlo presente para la correcta definición de la red y la posterior interpretación de los resultados.

Tabla 13 - Unidades de medida - WNTR package

Magnitud	Unidades
Longitud	m
Diámetro	m
Presión del agua*	m. c. a.
Altura	m
Demanda	m ³ /s
Tiempo	s
Concentración	kg/m ³

*asumiendo un fluido de 1000kg/m³

En la tabla 13 se resumen las unidades de medida del paquete WNTR para las magnitudes utilizadas en este trabajo.

5.3.3. Funciones desarrolladas en Python

Una función en Python es un código reutilizable que toma argumentos de entrada, realiza algún cálculo, y devuelve uno o varios resultados. Hay dos tipos de funciones en Python, funciones integradas que proporciona el lenguaje, y funciones definidas por el usuario.

Son una buena herramienta para reducir el tamaño de un código cuando hay tareas repetitivas. Además, permiten una organización visual del mismo.

En este trabajo se ha definido una serie de funciones cada cual realiza una tarea diferente.

```

def Alperovits_Network(D):...
def Decrease_diameter(D,i):...
def Increase_diameter(D,i):...
def Change_Network_Diameters(wn,D):...
def Eval_Network_Technical(wn,D):...
def Invalid(Invalid_solution,D):...
def Eval_Network_Cost(D):...
def Tabu_Search(D):...
def Graph_solution(Graph_BS,Graph_CS):...
def Graph_network(wn):...

```

Figura 24 - Definición de funciones en Python
Fuente: Elaboración propia

La figura 24 muestra el nombre de todas las funciones definidas en el algoritmo para la resolución del problema. Se intenta asignar un nombre representativo del cálculo realizado por cada función.

A continuación, se presenta una figura ilustrativa de cada función generada con sus entradas y sus salidas, y se definen los cálculos que éstas realizan.



Figura 25 - Función Alperovits_Network
Fuente: Elaboración propia

La figura 25 muestra la entrada, lista de diámetros asignados a cada arco de la red, y la salida, red definida, de la función Alperovits_Network. Esta función, usando el módulo *Network* del paquete WNTR, crea la red de Alperovits para un conjunto de diámetros definido. Para ello, la función contiene todos los datos referentes a la red, posición y demanda de los nodos, y longitud de los arcos. El nombre “wn” viene de las siglas del inglés “water network”, a partir de ahora se usará esta abreviatura para referirse a la red definida.

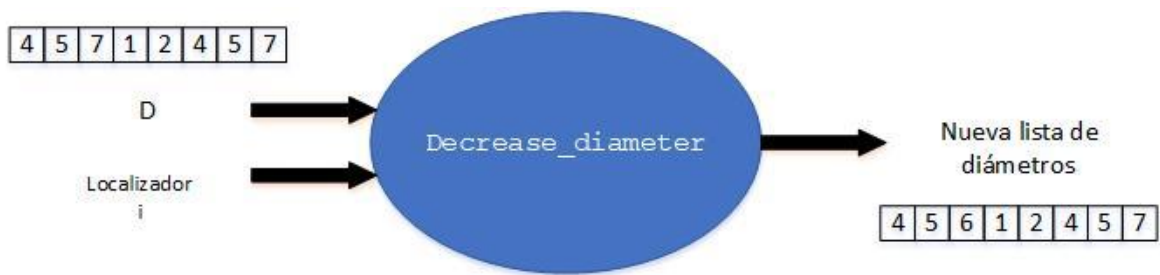


Figura 26 - Función Decrease_diameter
Fuente: Elaboración propia

La figura 26 muestra las entradas, lista de diámetros correspondiente a cada arco de la red, e *i*, localizador de un diámetro de la lista, y la salida, nueva lista de diámetros, de la función Decrease_diameter. Esta función asigna, al diámetro de la posición *i* de la lista D, el diámetro inmediatamente menor de la gama comercial. En la figura se puede apreciar un ejemplo donde *i* es igual a 3, por lo que, el tercer diámetro de la lista ha disminuido pasando de valer 7 a 6, como puede observarse en la nueva lista de diámetros, o salida de la función.

La función Increase_diameter consta de las mismas entradas y salidas, sin embargo, con dicha función el diámetro *i* de la lista D tomará el valor inmediatamente superior de la gama comercial de diámetros.



Figura 27 - Función Change_Network_diameter
Fuente: Elaboración propia

La función representada en la figura 27 cambia todos los diámetros de una red previamente definida. Esta función devuelve una red con las mismas características que la red de entrada, pero con los nuevos diámetros asignados a cada uno de los arcos que la forman. Se requieren dos entradas, la red "wn" definida y la lista con los nuevos diámetros. Para la implementación de esta función se utiliza el módulo *Network* del paquete WNTR, el cual permite modificar parámetros de una red sin necesidad de crearla de nuevo, reduciéndose así los tiempos de cálculo.

```
pipe=wn.get_link('pipe1')
pipe.diameter=D[i]
```

Figura 28 - Comando para asignar nuevo diámetro
Fuente: Elaboración propia

En la figura 28 se presentan los comandos utilizados para asignar un nuevo diámetro a una tubería de la red “wn”. Con la sentencia `get_link` se elige un arco, llamado en este ejemplo `pipe1`. En la siguiente línea se extrae el parámetro “diámetro” de dicho arco y se le asigna el valor de la lista de diámetros `D` que le corresponda.



Figura 29 - Función `Eval_Network_Technical`
Fuente: Elaboración propia

La función `Eval_Network_Technical`, figura 29, simula la red usando el módulo `EpanetSimulator` del paquete `WNTR`. En primer lugar, se realiza la simulación hidráulica de la red verificando que existe una solución definida por los caudales en los arcos y las presiones en los nodos. Una vez se alcanza dicha solución se comprueba si la presión en todos los nodos de la red supera la presión mínima de servicio.

Las entradas de la función son la red previamente definida, `wn`, y la lista de diámetros asignados a cada tubería, `D`. La salida será `True` si dicho conjunto de diámetros supone una solución válida y la presión en todos los nodos supera la mínima. Si se incumple alguno de estos requisitos la salida será `False`.

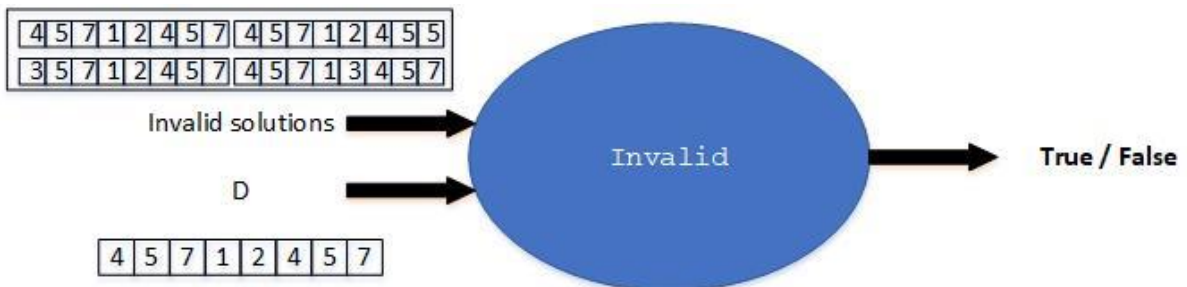


Figura 30 - Función `Invalid`
Fuente: Elaboración propia

La figura 30 presenta las entradas y salidas de la función `Invalid`. Esta función se crea para evitar realizar comprobaciones hidráulicas de soluciones ya estudiadas. Es decir,

ahorrar tiempo de resolución.

La función comprueba si una configuración determinada de diámetros, lista D, ya ha sido catalogada como solución inválida hidráulicamente. Para ello, cada vez que se encuentra una solución inválida, ésta se añade a la lista Invalid_solutions. Invalid_solutions es, por tanto, una lista de listas. En la función se recorre dicha lista comparando las soluciones inválidas almacenadas con la solución de entrada, lista de diámetros D.

La salida será True si la lista de diámetros D se corresponde con alguna de las listas contenidas en Invalid_solution, y False, en caso contrario.



Figura 31 - Función Eval_Network_Cost
Fuente: Elaboración propia

La función Eval_Network_Cost, figura 31, calcula el coste de la red para unos diámetros específicos en cada arco. Esta función tiene una sola entrada, la lista de diámetros D, y una sola salida, el coste, en unidades monetarias, de dicha asignación de diámetros.

Para realizar este cálculo la función contiene los costes asociados a cada diámetro de la gama comercial, establecidos por Alperovits and Shamir en su trabajo. Se recorre la lista de diámetros D, comparando cada elemento de la lista con los valores de los diámetros de la gama comercial, si coinciden se añade el coste de dicho diámetro multiplicado por la longitud del arco en el que está instalado.



Figura 32 - Función Tabu_Search
Fuente: Elaboración propia

La función `Tabu_Search`, figura 32, es la principal del código, conteniendo los mecanismos referentes a la metaheurística presentados en la sección 5.2. Dentro de esta función se llama a las otras funciones explicadas, ya que, en general, son parte de los procedimientos incluidos en la búsqueda tabú.

A partir de la solución inicial, lista de diámetros `D`, la función devuelve la mejor solución encontrada tras aplicar la metaheurística, `best_solution`, y el coste, calculado con la función `Eval_Network_Cost`, de dicha solución.

Por último, se definen dos funciones encargadas de generar distintas gráficas de los resultados obtenidos. Éstas son `Graph_solution` y `Graph_network`.

A través de la función `Graph_solution` se representan dos gráficas que muestran la evolución de la mejor solución y la solución actual durante la ejecución de la búsqueda tabú. Las entradas de esta función son dos listas, una que contiene los costes de las mejores soluciones encontradas durante la implementación del algoritmo, y la otra contiene todas las soluciones que toman el estatus de solución actual.

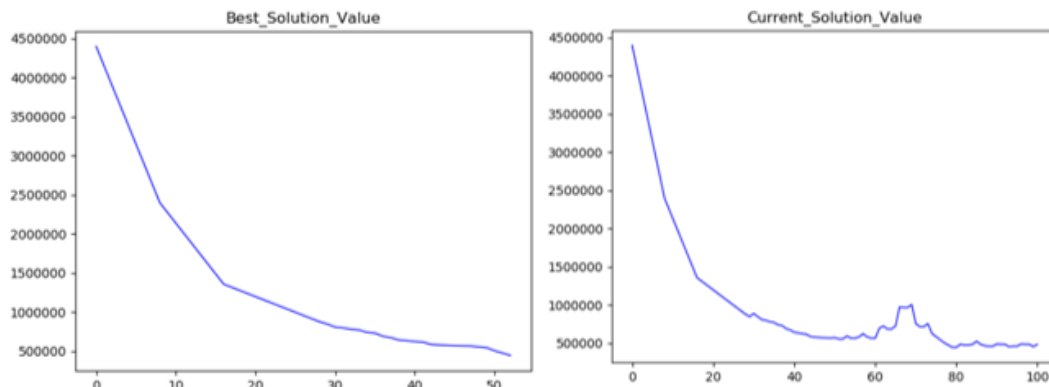


Figura 33 - Gráficas generadas con la función `Graph_solution`
Fuente: Elaboración propia

En la figura 33 se exhiben las dos gráficas que se crean al utilizar la función explicada. A la izquierda la evolución de las mejor solución, y a la derecha la evolución de la solución actual.

La función `Graph_network` simula la red generando tres gráficas: la demanda y los caudales en los arcos, y la presión de los nodos. Esto se realiza utilizando el módulo `Graphics` del paquete `WNTR`.

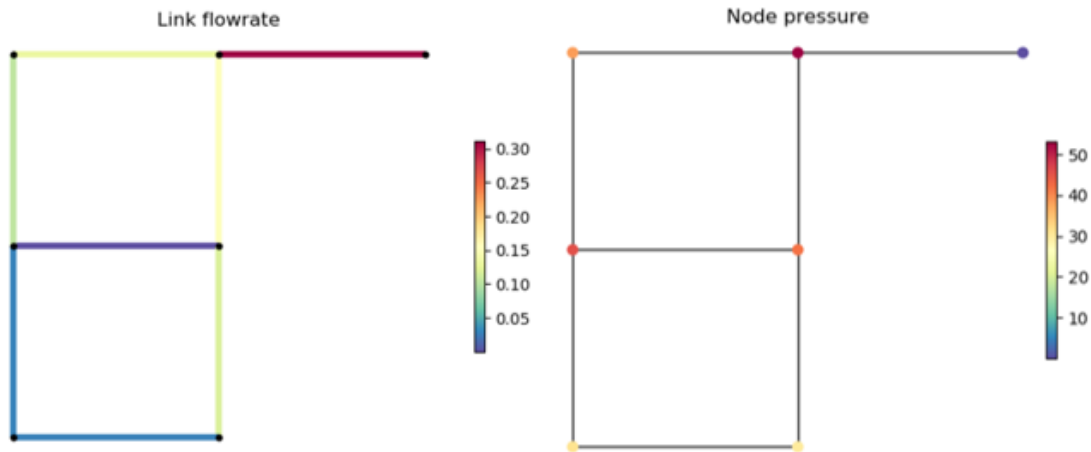


Figura 34 - Gráficas generadas con la función Graph_network
Fuente: Elaboración propia

En la figura 34 se observan dos de las gráficas que se generan al aplicar la función Graph_network. Estas gráficas permiten comprobar el rango en el que se mueven los parámetros principales de la red.

Por último, en la siguiente figura se muestra la función main(), función principal del código.

```
def main():
    D=[0.508,0.4064,0.4064,0.254,0.0508,0.3048,0.3048,0.1016]

    t1=time()
    print(Tabu_Search(D))
    t2=time()
    print('time execution:',t2-t1)
```

Figura 35 - Función principal – Python
Fuente: Elaboración propia

Las líneas de código que forman la función están recogidas en la figura 35. Una de las características de la función main() es que no tiene entradas ni salidas, al ejecutar el código ésta es la función que se ejecuta por defecto. Se comienza estableciendo la solución de partida, o solución inicial, lista de diámetros asignados a cada tramo de la red y se llama a la función Tabu_Search, cuya entrada es dicha lista. Además, se mide el tiempo de ejecución del algoritmo de búsqueda tabú.

5.3.4. Pseudocódigo

“Pseudo” deriva del griego “seudo” y su traducción sería falso, se utiliza para referirse a algo que no es original o que es una imitación. “Código” proviene del latín, representando el documento donde se establecían las leyes. En la actualidad, un código es un conjunto de símbolos con un determinado valor, previamente asignado, que se combinan respetando ciertas reglas y permiten transmitir un mensaje. En programación, código se utiliza para referirse al lenguaje que interpretan los procesadores o softwares. Mientras que los

pseudocódigos describen dichos algoritmos para una fácil interpretación humana.

Para concluir el capítulo 5 se presenta el pseudocódigo del algoritmo implementado. El código completo se añade en un anexo al final del trabajo.

```
----- INICIALIZACIÓN -----
Sol_Inicial=Definir_Solucion_Inicial()
Wn=Definir_red()

Inicializar n_iteraciones, Tabu_size, Tabu_list
Inicializar Lista Soluciones NO_validas
Inicializar Lista_Mejor_solucion, Lista_Solucion_Actual

Sol_actual=Sol_Inicial
Mejor_solución=Sol_actual

Mientras (n<n_iteraciones):

    ----- VECINDARIO -----
    Vecindario=Reduce_diametro(Sol_actual,wn)
    Quitar del vecindario vecinos ∈ NO_validas
    Evaluación Hidráulica(Vecindario)
    Actualizar lista NO_validas

    si Vecindario Vacío:
        i =Indice_aleatorio ∈ Lista Tabu=0
        Sol_actual=Incrementa_diametro(Sol_actual,i)
    sino:
        CosteVecindario=Eval_Coste(Vecindario)
        vecino=vecino ∈ Vecindario con menor CosteVecindario
        Coste_vecino=Eval_Coste(vecino)

    ----- Criterio Aspiración y Memoria corto plazo-----
    si vecino∉Lista_tabu or Coste_vecino<Coste_Mejor_sol:
        Actualiza Sol_actual
        Coste_Sol_actual=Coste_vecino
        Añadir Sol_actual a Lista_Solucion_Actual
```

```

        Actualiza Tabu_list incrementando en Tabu_size+1

    si Coste_Sol_actual<Coste_Mejor_sol:
        Actualiza Mejor_sol
        Coste_Mejor_sol=Coste_Sol_actual
        Añadir Mejor_sol a Lista_Mejor_Solucion

    Tabu_list=Tabu_list-1
    n=n+1

----- Gráficas -----
Genera grafica Lista_Mejor_Solucion
Genera grafica Lista_Solucion_Actual
Genera graficas parámetros Red

```

FIN

El pseudocódigo anterior se divide en inicialización, generación del vecindario, criterio de aspiración, memoria a corto plazo, y gráficas. En la inicialización se define la solución inicial y la red, además, se inicializan los parámetros característicos de la búsqueda tabú. La generación del vecindario se realiza a través de los mecanismos explicados en los apartados anteriores. Una vez generado el vecindario, éste se evalúa y se analiza la mejor solución del mismo, si esta solución tiene estatus tabú se aplica el criterio de aspiración. En el caso de que no existe ninguna solución válida en un vecindario, se aplicaría el procedimiento de aumentar uno de los diámetros de la solución actual. Una vez se ha implementado el algoritmo tantas veces como iteraciones se fijen al inicio, se generan las gráficas.

Con este pseudocódigo queda explicado el algoritmo de resolución.

6. Resultados

En esta sección se muestran las simulaciones realizadas y los resultados obtenidos con el fin de validar el algoritmo. En cada simulación se modifican ciertos parámetros que afectan al método de resolución. En primer lugar se explica la elección de dichos parámetros, a continuación se muestra los resultados obtenidos en las simulaciones realizadas. En los siguientes apartados se evalúa el algoritmo y las características hidráulicas de las soluciones alcanzadas. Por último, se comparan los resultados obtenidos con los de otros autores recogidos en la literatura.

6.1. Ajustes de los parámetros del algoritmo

En esta sección se estudian los distintos parámetros que afectan al método de resolución con el fin de ajustar sus valores y así evitar ineficiencias computacionales. Los parámetros que más influyen en la ejecución del algoritmo de búsqueda tabú son: la solución inicial, el criterio de parada, y el tabu_size, o número de iteraciones que un movimiento permanece con el estatus “tabú”.

- **Solución inicial**

En primer lugar, se ha decidido ejecutar el algoritmo para dos soluciones iniciales diferentes. Esto permitirá comprobar la validez del algoritmo sea cual sea la solución de partida.

Las soluciones se caracterizan por una lista con ocho elementos que representan el diámetro asignado a cada arco de la red, por ejemplo, el diámetro del arco 1 de la red es el primer elemento de la lista, y así sucesivamente. La solución inicial comienza tomando el papel de solución actual y , entonces, el algoritmo generará el vecindario de dicha solución.

Tabla 14 - Soluciones de partida

Solución inicial A	[0.6096,0.6096,0.6096,0.6096,0.6096,0.6096,0.6096,0.6096]
Solución inicial B	[0.508,0.4064,0.4064,0.254,0.0508,0.3048,0.3048,0.1016]

La primera fila de la tabla 14 muestra la primera solución inicial utilizada, de aquí en adelante nos referiremos a ella como solución inicial A. Ésta es una solución en la cual todas las tuberías de la red tienen el mayor diámetro de la gama comercial. El coste asociado a esta solución es de 4,400,000 u.m. Esta solución es inadmisiblemente

ya que las presiones en los nodos no estarían dentro de los límites establecidos, además, la velocidad del agua en las tuberías sería excesivamente baja debido a los grandes diámetros de las mismas. No obstante, parece lógico como norma general comenzar por la peor solución existe para así explorar un espacio de búsqueda mayor.

La segunda solución de partida, solución inicial B, es una solución válida hidráulicamente y con un coste de 498,000 u.m. Se pueden observar los diámetros, en metros, asignados a cada arco de la red en la segunda fila de la tabla 14. Esta solución ya reduce en un 88.68% el coste de la solución inicial A. No obstante, dado que se conocen soluciones aún mejores para este problema, se sabe que ésta no es una solución óptima y por tanto, plantea un escenario diferente al anterior.

- **Tabu_size**

Otro de los parámetros que ha sido modificado para las distintas simulaciones es el `tabu_size`, número de iteraciones que un movimiento permanece en la lista de movimientos prohibidos o lista tabú. Generalmente este parámetro se fija en torno a la mitad del total de movimientos posibles partiendo de una solución, es decir, si el vecindario de una solución son 10 soluciones vecinas, el `tabu_size` más común es 5. No hay reglas establecidas, pero un número demasiado grande de este parámetro puede generar vecindarios vacíos, lo que haría que el tiempo de resolución aumentase sin avances en el estudio del espacio de soluciones. En cambio, si el `tabu_size` es demasiado pequeño la metaheurística no cumpliría su función ya que no prohibiría dichos movimientos en un número significativo de iteraciones.

Dado que el vecindario de una solución está formado por ocho soluciones vecinas, se elige un `tabu_size` igual a 4. Hay que tener en cuenta que por las características del problema, existirán soluciones que no sean válidas hidráulicamente y, por ello, no se añadirán al vecindario, apareciendo en ciertas ocasiones vecindarios de menor tamaño. En consecuencia, se decide implementar el algoritmo también utilizando un `tabu_size` igual a 2.

En resumen, se realizaran simulaciones con estos dos valores para el parámetro `tabu_size`.

- **Criterio de parada**

El criterio de parada influye en el tiempo de resolución del problema, tiempo de cálculo del software. Es conveniente comenzar con un número de iteraciones reducido e ir aumentándolo, para comprobar el correcto funcionamiento del algoritmo. Existen dos criterios generales para fijar este parámetro. El primero, y más simple, es fijar un número máximo de iteraciones. En ocasiones se observa que a partir de un cierto número de iteraciones la mejor solución del problema no se reduce. Por ello, el segundo criterio es crear un mecanismo de control del número de iteraciones en las que la mejor solución no se actualiza, y detener el algoritmo cuando el número máximo fijado se alcance.

En este trabajo se escoge el primer criterio, estableciendo un número de iteraciones para las diferentes simulaciones de 100, 1000 y 2000 iteraciones. Cabe destacar el algoritmo se testó para 5000 y hasta 10000 iteraciones, no observándose mejora a partir de las 2000 iteraciones. Por ello, este número se establece como el máximo.

Un criterio establecido por Baños et al. (2010) en su trabajo, en el cual se resuelven redes de distintos tamaños, es el siguiente:

$$n_i = K * n_a * \log(n_\phi) \quad (\text{ec. 36})$$

Dónde:

n_i : Número de iteraciones

K : Valor constante

n_a : Número de arcos de la red

n_ϕ : Número de diámetros de la gama comercial

En este trabajo al resolver una sola red se decide utilizar un número fijo de iteraciones. No obstante, este criterio tiene utilidad para futuras pruebas del algoritmo en otras redes modelo.

Una vez definido el valor de los parámetros que interesa modificar, se obtiene un total de doce combinaciones, por lo que se realizarán doce simulaciones con las siguientes características.

Tabla 15 - Parámetros de las distintas simulaciones

		Tabu_size	
		4	2
Número de iteraciones	100	Solución inicial A	Solución inicial A
		Solución inicial B	Solución inicial B
	1000	Solución inicial A	Solución inicial A
		Solución inicial B	Solución inicial B
	2000	Solución inicial A	Solución inicial A
		Solución inicial B	Solución inicial B

En la tabla 15 se observa el valor de los parámetros de cada una de las doce simulaciones realizadas.

6.2. Simulaciones

Se procede a mostrar en detalle los datos de entrada y los resultados obtenidos en cada una de las doce simulaciones realizadas.

Tabla 16 - Entradas simulación 1

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0,6096	0,6096	0,6096	0,6096	0,6096	0,6096	0,6096	0,6096
Coste de la solución [u.m.]				Tabu_size	Número de iteraciones			
	4.400.000			4	100			

Tabla 17 - Salidas simulación 1

Coste de la solución [u.m.]	Tiempo de resolución [s]		Nº soluciones no válidas					
420.000	35		336					
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0,5080	0,2540	0,4064	0,0254	0,3556	0,2540	0,2540	0,0254
Caudal [m3/s]	0,311	0,102	0,181	0,000	0,147	0,056	0,075	0,000
Velocidad [m/s]	1,535	2,019	1,395	0,541	1,484	1,100	1,471	-0,367
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0,00	55,96	30,87	46,56	32,48	30,80	30,90	

Las entradas de la primera simulación, recogidas en la tabla 16, son la solución inicial, diámetros asignados a cada arco de la red, el coste de dicha solución en unidades monetarias, el valor del tabu_size y el número de iteraciones.

Las salidas o resultados de la primera simulación, expuestas en la tabla 17, son la mejor solución encontrada y el coste que conlleva. El tiempo de resolución, parámetro que no aporta mucha información ya que depende del procesador y la memoria del ordenador en el que se realizan las simulaciones. No obstante, ya que todas las simulaciones se realizan con el mismo ordenador, Intel Core i5-5200U con 4GB de memoria, el tiempo de resolución sirve para comparar unas simulaciones con otras.

Otra de las salidas almacenadas es el número de soluciones no válidas analizadas, este parámetro sólo almacena cada solución no válida hidráulicamente una vez. Permite valorar el entorno analizado en cada simulación. Si el número de soluciones no válidas analizadas es muy alto, se ha dedicado mucho tiempo de cálculo a explorar un espacio de soluciones sin validez hidráulica, lo que no conviene si el objetivo es reducir el tiempo total de resolución.

Los parámetros característicos de la red cuya distribución de diámetros es la solución final se obtienen usando el paquete de Python WNTR. Como se observa en la tabla 17, el algoritmo facilita el caudal en cada arco y la presión en cada nodo de la red. Además, la velocidad media del agua en cada arco se calcula a partir del caudal y del diámetro de cada tubería usando la siguiente fórmula.

$$v_j = \frac{Q_j}{S_j} \quad (ec. 37)$$

Dónde:

v : Velocidad media del agua en el arco j

Q_j : Caudal que circula a través del arco j

S_j : Sección transversal de la tubería en el arco j

En este trabajo la sección transversal de las tuberías es circular, por ello se calcula siguiendo la siguiente ecuación.

$$S_j = \pi \left(\frac{\phi_j}{2} \right)^2 \quad (ec. 38)$$

Dónde:

ϕ_j : Diámetro de la tubería j.

Sustituyendo el término anterior en la ecuación 37 se obtiene la siguiente relación entre velocidad, caudal y diámetro de una tubería.

$$v_j = \frac{4 \cdot Q}{\pi \cdot \phi^2} \quad (ec. 39)$$

Cabe destacar que no se han fijado límites para la velocidad del agua en las tuberías, no obstante, se conocen los valores mínimos y máximos aconsejados para mantener la calidad del agua y el buen estado de la red. Estos valores son 0,3 y 2,0 m/s.

A continuación, se muestran las entradas y las salidas obtenidas en el resto de simulaciones. El color azul representa las simulaciones cuya solución de partida es la solución inicial A, y el color rojo las simulaciones cuya solución de partida es la solución inicial B.

Tabla 18 - Entradas simulación 2

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096
Coste de la solución [u.m]				Tabu_size	Número de iteraciones			
	4400000			2	100			

Tabla 19 - Salidas simulación 2

Coste de la solución [u.m]	Tiempo de resolución [s]							Nº soluciones no válidas
448000	23							312
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.4572	0.3556	0.3556	0.0254	0.4064	0.2032	0.3556	0.2032
Caudal [m3/s]	0.311	0.131	0.152	0.000	0.119	0.027	0.104	0.029
Velocidad [m/s]	1.895	1.322	1.530	0.138	0.914	0.830	1.043	0.883
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	53.25	38.60	42.15	45.60	30.14	31.37	

Tabla 20 - Entradas simulación 3

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096
Coste de la solución [u.m]				Tabu_size	Número de iteraciones			
	4400000			4	1000			

Tabla 21 - Salidas simulación 3

Coste de la solución [u.m]					Tiempo de resolución [s]	Nº soluciones no válidas		
	424000				102	733		
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.4572	0.3556	0.3556	0.0254	0.3556	0.0254	0.3556	0.3048
Caudal [m3/s]	0.311	0.158	0.125	0.000	0.092	0.000	0.130	0.055
Velocidad [m/s]	1.895	1.591	1.262	0.330	0.925	0.316	1.311	0.759
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	53.25	36.70	43.98	42.12	31.59	30.13	

Tabla 22 - Entradas simulación 4

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096
Coste de la solución [u.m]				Tabu_size	Número de iteraciones			
	4400000			2	1000			

Tabla 23 - Salidas simulación 4

Coste de la solución [u.m]					Tiempo de resolución [s]	Nº soluciones no válidas		
	424000				80	545		
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.4572	0.3556	0.3556	0.0254	0.3556	0.0254	0.3556	0.3048
Caudal [m3/s]	0.311	0.158	0.125	0.000	0.092	0.000	0.130	0.055
Velocidad [m/s]	1.895	1.591	1.262	0.330	0.925	0.316	1.311	0.759
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	53.25	36.70	43.98	42.12	31.59	30.13	

Tabla 24 - Entradas simulación 5

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096
Coste de la solución [u.m]				Tabu_size	Número de iteraciones			
	4400000			4	2000			

Tabla 25 - Salidas simulación 5

Coste de la solución [u.m]		Tiempo de resolución [s]				Nº soluciones no válidas			
420000		195				747			
Arco	1	2	3	4	5	6	7	8	
Diámetro [m]	0.5080	0.2540	0.4064	0.0254	0.3556	0.2540	0.2540	0.0254	
Caudal [m3/s]	0.311	0.102	0.181	0.000	0.147	0.056	0.075	0.000	
Velocidad [m/s]	1.535	2.019	1.395	0.541	1.484	1.100	1.471	-0.367	
Nodo	1	2	3	4	5	6	7		
Presión[m.c.a.]	0.00	55.96	30.87	46.56	32.48	30.80	30.90		

Tabla 26 - Entradas simulación 6

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096	0.6096
Coste de la solución [u.m]		Tabu_size			Número de iteraciones			
4400000		2			2000			

Tabla 27 - Salidas simulación 6

Coste de la solución [u.m]		Tiempo de resolución [s]				Nº soluciones no válidas			
424000		154				460			
Arco	1	2	3	4	5	6	7	8	
Diámetro [m]	0.4572	0.3556	0.3556	0.0254	0.3556	0.0254	0.3556	0.3048	
Caudal [m3/s]	0.311	0.158	0.125	0.000	0.092	0.000	0.130	0.055	
Velocidad [m/s]	1.895	1.591	1.262	0.330	0.925	0.316	1.311	0.759	
Nodo	1	2	3	4	5	6	7		
Presión[m.c.a.]	0.00	53.25	36.70	43.98	42.12	31.59	30.13		

Tabla 28 - Entradas simulación 7

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.4064	0.4064	0.2540	0.0508	0.3048	0.3048	0.1016
Coste de la solución [u.m]		Tabu_size			Número de iteraciones			
498000		4			100			

Tabla 29 - Salidas simulación 7

Coste de la solución [u.m]			Tiempo de resolución [s]			Nº soluciones no válidas		
482000			17			518		
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.2032	0.4572	0.3048	0.3048	0.1016	0.1524	0.2450
Caudal [m3/s]	0.311	0.040	0.243	0.113	0.097	0.005	0.013	0.050
Velocidad [m/s]	1.535	1.243	1.480	1.543	1.331	0.676	0.687	1.062
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	55.96	37.99	46.68	44.28	31.05	30.26	

Tabla 30 - Entradas simulación 8

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.4064	0.4064	0.2540	0.0508	0.3048	0.3048	0.1016
Coste de la solución [u.m]			Tabu_size			Número de iteraciones		
498000			2			100		

Tabla 31 - Salidas simulación 8

Coste de la solución [u.m]			Tiempo de resolución [s]			Nº soluciones no válidas		
442000			18			533		
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.4572	0.3048	0.4064	0.2540	0.3556	0.1524	0.2540	0.2540
Caudal [m3/s]	0.311	0.087	0.196	0.057	0.106	0.014	0.060	0.041
Velocidad [m/s]	1.895	1.196	1.511	1.120	1.067	0.784	1.175	0.814
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	53.25	38.62	43.15	43.09	30.03	30.28	

Tabla 32 - Entradas simulación 9

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.4064	0.4064	0.2540	0.0508	0.3048	0.3048	0.1016
Coste de la solución [u.m]			Tabu_size			Número de iteraciones		
498000			4			1000		

Tabla 33 - Salidas simulación 9

Coste de la solución [u.m]	Tiempo de resolución [s]					Nº soluciones no válidas		
424000	95					1206		
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.4572	0.3556	0.3556	0.0254	0.3556	0.0254	0.3556	0.3048
Caudal [m3/s]	0.311	0.158	0.125	0.000	0.092	0.000	0.130	0.055
Velocidad [m/s]	1.895	1.591	1.262	0.330	0.925	0.316	1.311	0.759
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	53.25	36.70	43.98	42.12	31.59	30.13	

Tabla 34 - Entradas simulación 10

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.4064	0.4064	0.2540	0.0508	0.3048	0.3048	0.1016
Coste de la solución [u.m]	Tabu_size			Número de iteraciones				
	498000			2			1000	

Tabla 35 - Salidas simulación 10

Coste de la solución [u.m]	Tiempo de resolución [s]					Nº soluciones no válidas		
424000	65					535		
Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.4572	0.3556	0.3556	0.0254	0.3556	0.0254	0.3556	0.3048
Caudal [m3/s]	0.311	0.158	0.125	0.000	0.092	0.000	0.130	0.055
Velocidad [m/s]	1.895	1.591	1.262	0.330	0.925	0.316	1.311	0.759
Nodo	1	2	3	4	5	6	7	
Presión[m.c.a.]	0.00	53.25	36.70	43.98	42.12	31.59	30.13	

Tabla 36 - Entradas simulación 11

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.4064	0.4064	0.2540	0.0508	0.3048	0.3048	0.1016
Coste de la solución [u.m]	Tabu_size			Número de iteraciones				
	498000			4			2000	

Tabla 37 - Salidas simulación 11

Coste de la solución [u.m]		Tiempo de resolución [s]				Nº soluciones no válidas			
420000		178				1084			
Arco	1	2	3	4	5	6	7	8	
Diámetro [m]	0.5080	0.2540	0.4064	0.0254	0.3556	0.2540	0.2540	0.0254	
Caudal [m3/s]	0.311	0.102	0.181	0.000	0.147	0.056	0.075	0.000	
Velocidad [m/s]	1.535	2.019	1.395	0.541	1.484	1.100	1.471	-0.367	
Nodo	1	2	3	4	5	6	7		
Presión[m.c.a.]	0.00	55.96	30.87	46.56	32.48	30.80	30.90		

Tabla 38 - Entradas simulación 12

Arco	1	2	3	4	5	6	7	8
Diámetro [m]	0.5080	0.4064	0.4064	0.2540	0.0508	0.3048	0.3048	0.1016
Coste de la solución [u.m]		Tabu_size			Número de iteraciones			
498000		2			2000			

Tabla 39 - Salidas simulación 12

Coste de la solución [u.m]		Tiempo de resolución [s]				Nº soluciones no válidas			
420000		128				630			
Arco	1	2	3	4	5	6	7	8	
Diámetro [m]	0.5080	0.2540	0.4064	0.0254	0.3556	0.2540	0.2540	0.0254	
Caudal [m3/s]	0.311	0.102	0.181	0.000	0.147	0.056	0.075	0.000	
Velocidad [m/s]	1.535	2.019	1.395	0.541	1.484	1.100	1.471	-0.367	
Nodo	1	2	3	4	5	6	7		
Presión[m.c.a.]	0.00	55.96	30.87	46.56	32.48	30.80	30.90		

Las tablas anteriores recogen los resultados de todas las simulaciones. Se observa que en las simulaciones 1, 5, 11 y 12 se obtiene la misma solución cuyo coste es 420,000 u.m, esto se recoge en las tablas 17, 25, 37 y 39. Por otro lado, en las simulaciones 3, 4, 6, 9 y 10 se llega a una solución de 424,000 u.m., tablas 21, 23, 27, 33 y 35. En las otras tres simulaciones restantes, simulación 2, 7 y 8, recogidas en las tablas 19, 29 y 31, se alcanzan soluciones diferentes y de peor calidad.

La mejor solución encontrada tiene un coste de 420,000 u.m. y se llega a esta solución en cuatro de las simulaciones realizadas, tres de ellas para 2000 iteraciones. En cinco casos la mejor solución encontrada tiene un coste de 424,000 u.m., parece que una vez alcanzada esta solución el algoritmo tiene dificultad para escapar del óptimo local. Finalmente, las tres soluciones encontradas con peor coste coinciden con simulaciones de 100 iteraciones.

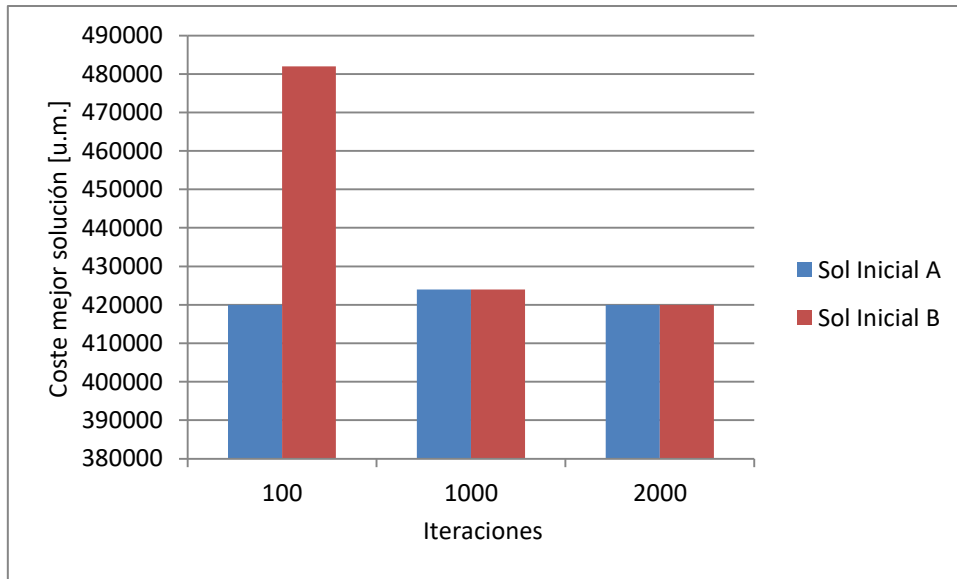


Figura 36 - Coste mejor solución vs Número de iteraciones para Tabu-size=4
Fuente: Elaboración propia

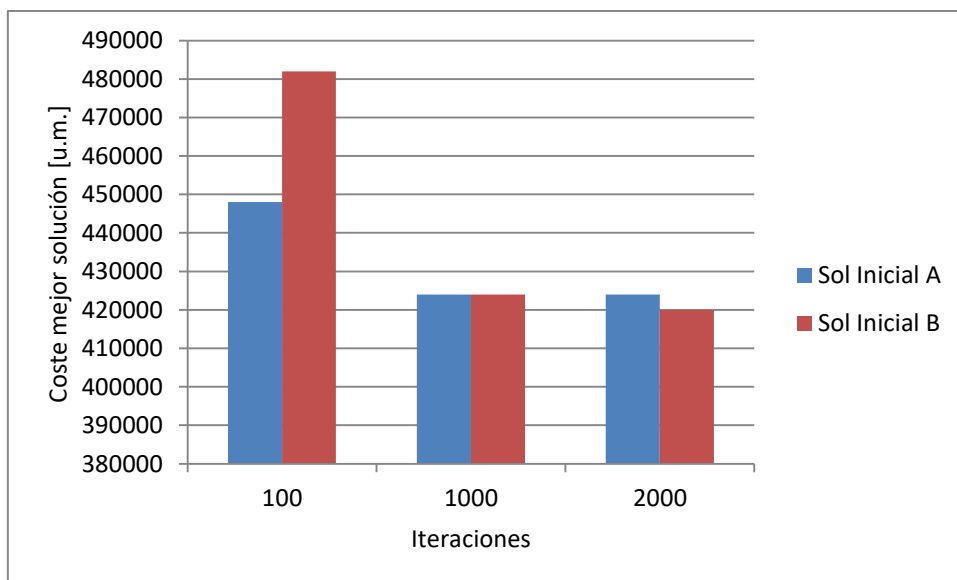


Figura 37 - Coste mejor solución vs Número de iteraciones para Tabu-size=2
Fuente: Elaboración propia

Las gráficas de las figuras 36 y 37 muestran el coste de la mejor solución encontrada para las doce simulaciones. En azul las simulaciones cuya solución de partida era la solución inicial A, y en rojo las simulaciones cuya solución de partida en la solución inicial B. En la figura 36 se observan los resultados obtenidos para un `tabu_size` igual a 4, mientras que la figura 37 presenta los resultados para un valor del `tabu_size` igual a 2.

Se observa que tanto para un `tabu_size` igual a 4 como a 2 se obtienen mejores resultados cuando se parte de la solución inicial A, es decir, la peor solución posible del problema. Esto puede ocurrir debido a que la probabilidad de escapar de un óptimo local en pocas iteraciones es menor cuando se parte de una solución buena. Al incluir el algoritmo una generación aleatoria de soluciones cuando no se encuentran soluciones válidas, parece

que esta aleatoriedad funciona mejor si partimos de una solución de mala calidad. La función `Increase_diameter` aumenta un diámetro escogido al azar cuando en un vecindario no se encuentra ninguna solución que cumpla los requisitos marcados.

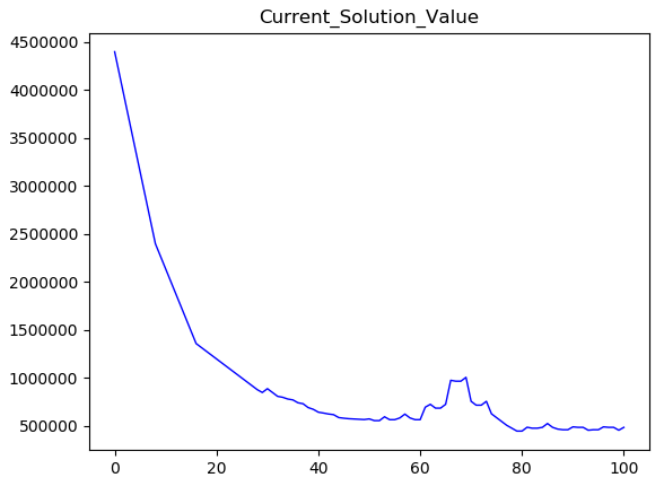
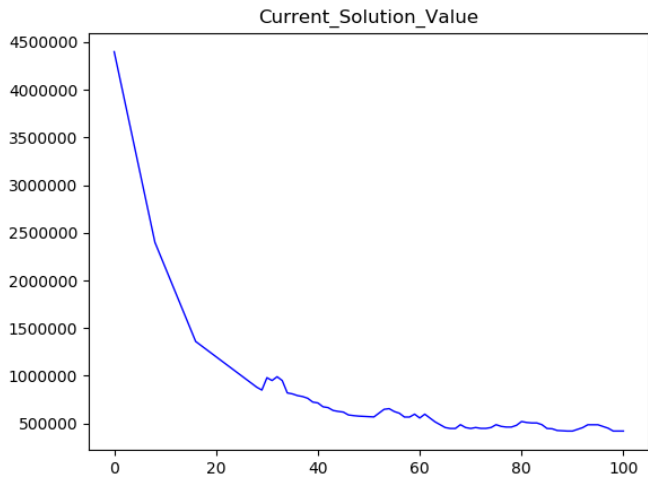
En ambas gráficas se observa que cuando el número de iteraciones aumenta las soluciones encontradas son, en general, mejores. Como se comentó en la anterior sección, se ejecuta el algoritmo para un número mayor de iteraciones sin observarse mejora en la solución, por ello, se decide parar en 2000 iteraciones.

6.3. Evaluación del algoritmo

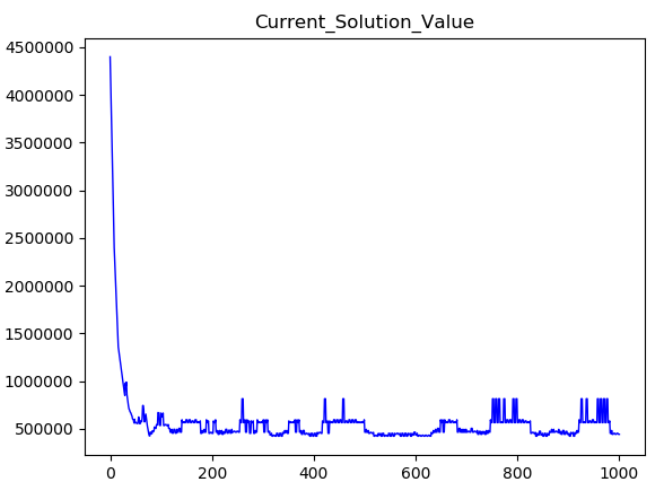
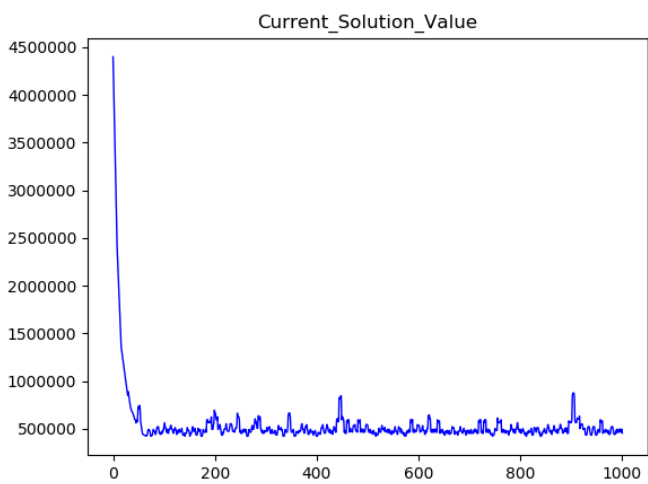
Una forma útil de verificar si el algoritmo de búsqueda tabú está funcionando correctamente es representando las gráficas de evolución, tanto de la mejor solución, como de la solución actual. Una característica distintiva de esta metaheurística es que permite el empeoramiento de la solución con el fin de escapar de los óptimos locales. Se recuerda que la solución actual se actualiza en cada iteración, mientras que la mejor solución solo se actualiza si la solución actual tiene un coste menor al de la mejor solución encontrada.

A continuación, se muestran las gráficas de evolución, tanto de la mejor solución, como de la solución actual para las doce simulaciones realizadas. De nuevo el color azul representa las simulaciones cuya solución de partida es la solución inicial A, y el color rojo las que utiliza la solución inicial B.

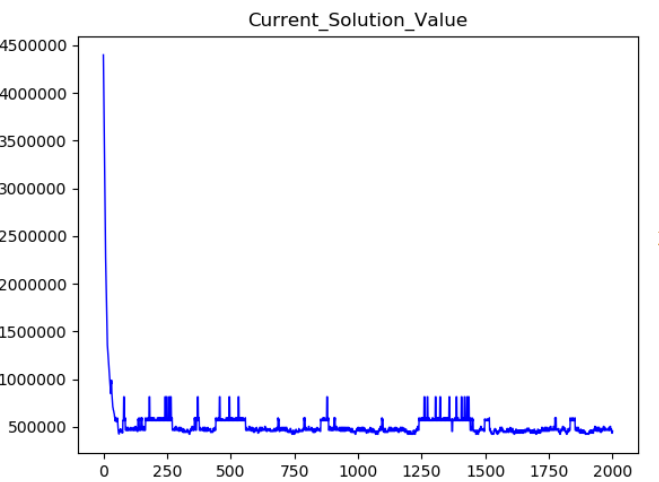
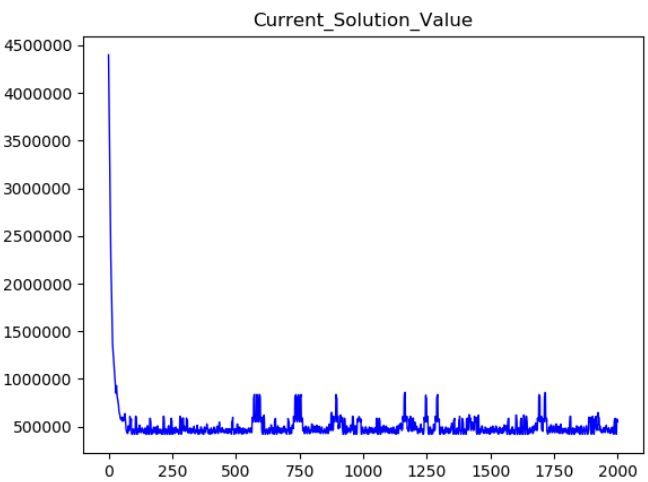
El eje X representa la iteración y el eje Y el coste de la solución para dicha iteración en unidades monetarias. Las gráficas se han organizado en dos columnas, la primera correspondiente a aquellas simulaciones cuyo `tabu_size` es igual a 4, mientras en la segunda columna el `tabu_size` es igual a 2. Cada fila de figuras representa un número de iteraciones, mostrado en la parte derecha.



100



1000



2000

Tabu_size = 4

Tabu_size = 2

Figura 38 - Evolución Solución actual – Simulaciones solución inicial A
Fuente: Elaboración propia

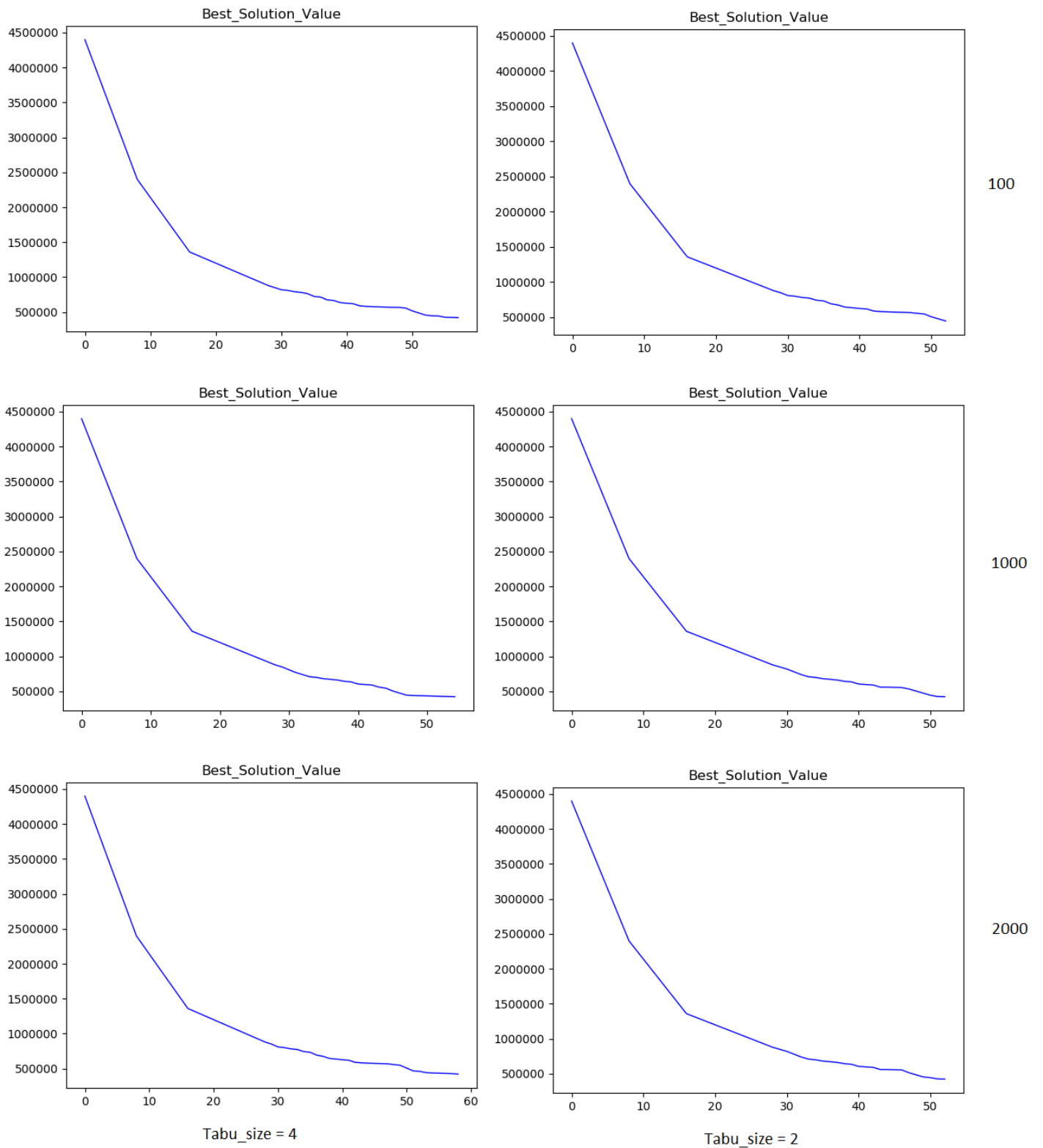


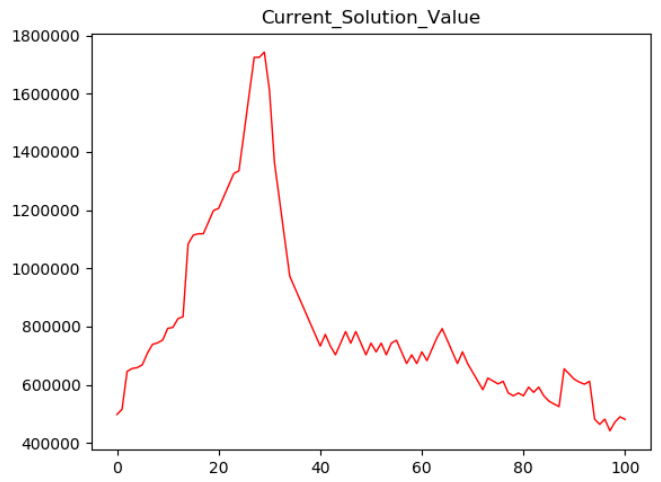
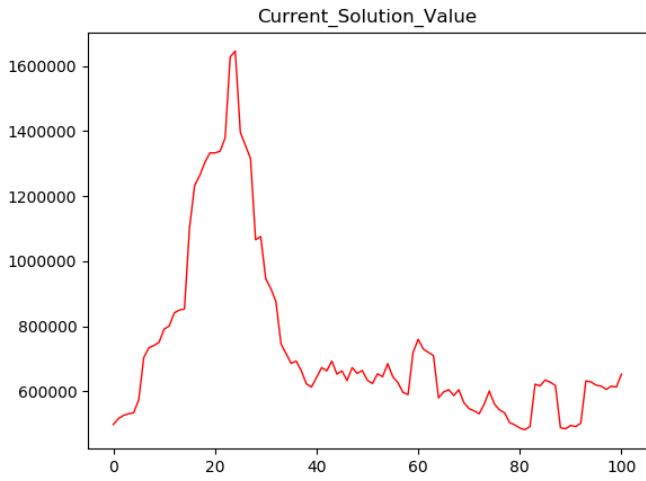
Figura 39 - Evolución Mejor solución – Simulaciones solución inicial A
Fuente: Elaboración propia

Las figuras 38 y 39 presentan la evolución de las soluciones para las seis simulaciones cuya solución de partida es la solución inicial A, esta solución se caracteriza por ser la peor del

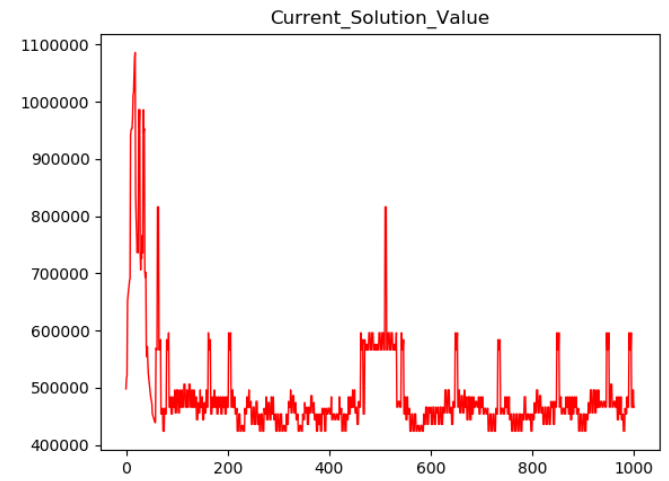
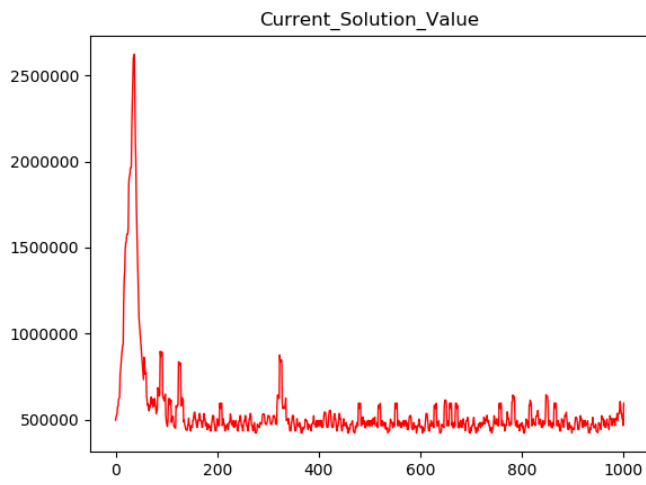
problema. Se observa cómo tanto la solución actual como las mejores soluciones tienen un coste inicial muy alto 4,400,000 u.m., y por lo tanto, se aprecia un descenso del mismo muy acentuado en las primeras iteraciones.

El número total de iteraciones de cada simulación puede observarse en el eje X de las gráficas de solución actual, con título *Current_Solution*. El funcionamiento del algoritmo queda validado al observar que la solución actual empeora y consigue mejorar después, fenómeno que ve más fácilmente en las dos primeras simulaciones de la figura 38. Con este mecanismo se consigue analizar un espacio de soluciones mayor.

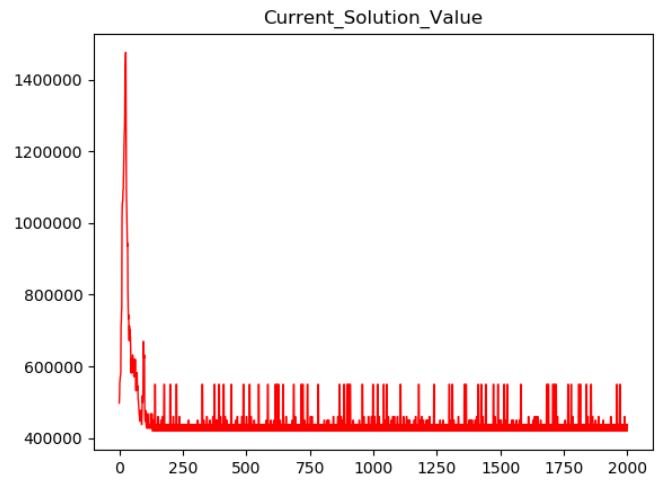
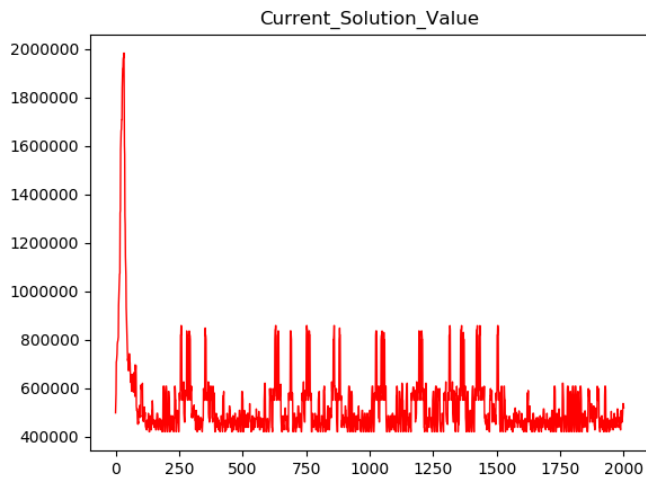
En las gráficas de evolución de la mejor solución el eje X representa el número de veces que la mejor solución se actualizó, es decir, que se encontró una solución mejor. Se observa que, independientemente del número de iteraciones, 100, 1000 o 2000, la mejor solución se actualiza alrededor de 50 veces. En las primeras iteraciones el algoritmo encuentra mejores soluciones con facilidad, no obstante, cuando el problema alcanza una solución con un coste relativamente bajo es cuando le cuesta más escapar de los óptimos locales.



100



1000



2000

Tabu_size = 4

Tabu_size=2

Figura 40 - Evolución Solución actual – Simulaciones solución inicial B
Fuente: Elaboración propia

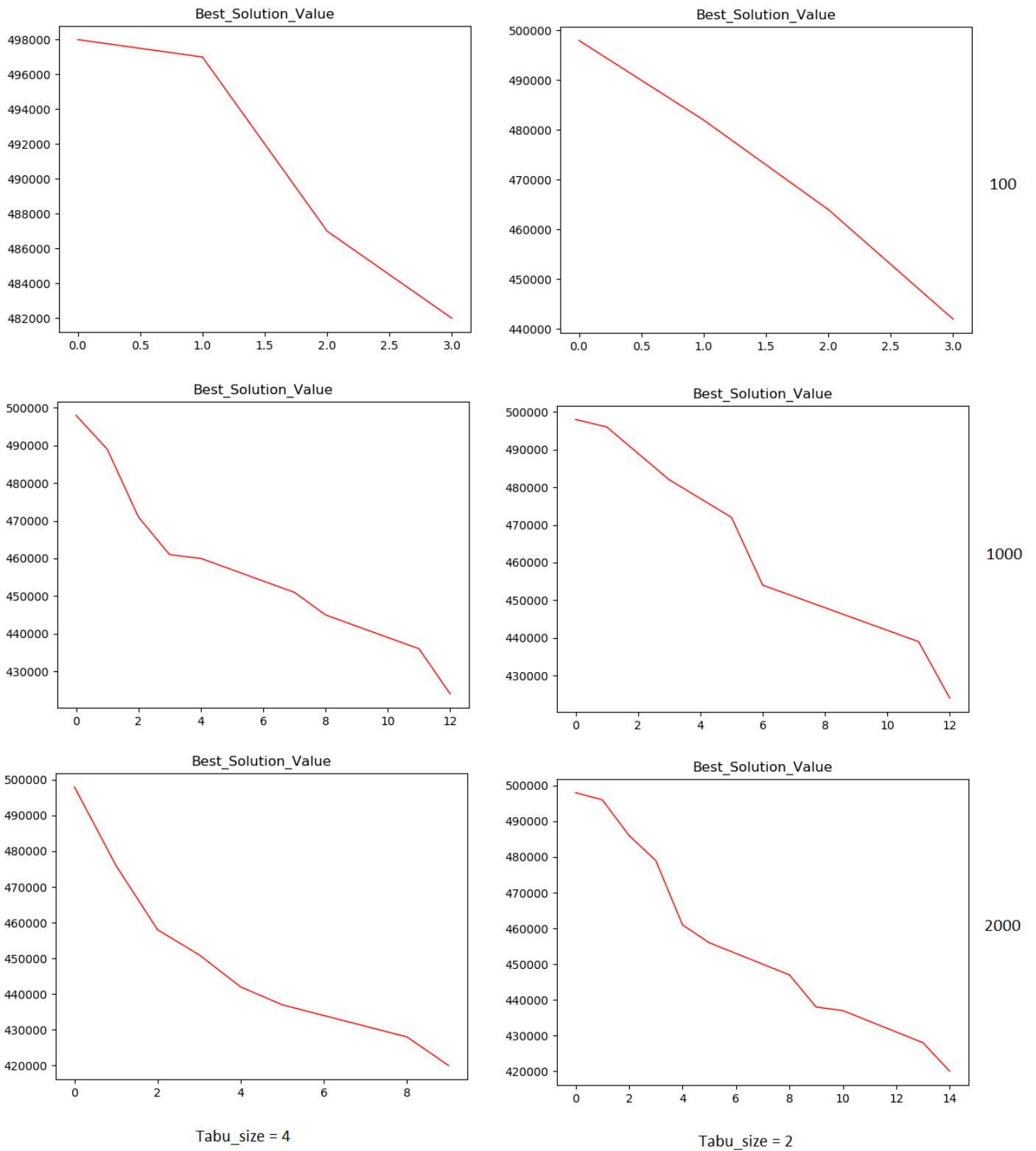


Figura 41 - Evolución Mejor solución – Simulaciones solución inicial B
Fuente: Elaboración propia

En las figuras 40 y 41 se muestran las gráficas de evolución para las simulaciones cuya solución de partida es la solución inicial B, solución cuyo coste es 498,000 u.m. En todas las

simulaciones de este conjunto se observa como el coste de la solución actual, *Current_solution*, experimenta un ascenso considerable en las primeras iteraciones, figura 40. Al partir de una solución ya optimizada, el algoritmo encuentra dificultades para hallar una mejor solución por lo que comienza aumentando ciertos diámetros, incrementando así el coste total de la solución en las primeras iteraciones. Se observa como gracias a este empeoramiento se consigue explorar nuevos espacios de búsqueda alcanzando finalmente mejores soluciones.

A diferencia de la evolución de las mejores soluciones para el conjunto de simulaciones anteriores, en estas seis simulaciones la mejor solución se actualiza entre 3 y 14 veces, como muestra la figura 41. Al partir de una solución mejor, el número de veces que esta puede mejorar se reduce. Se puede observar que, para las dos primeras simulaciones de este conjunto, habiendo un total de 100 iteraciones solo en tres ocasiones se encuentra una solución mejor, además, en dichos casos la soluciones alcanzadas no tiene un coste bajo en comparación con el resto.

Se concluye que el algoritmo diseñado obtiene mejores resultados utilizando una solución de partida de baja calidad ya que esto permite explorar un espacio de búsqueda mayor. Para este tipo de soluciones de partida, la respuesta mejora usando un *tabu_size* igual a 4, es decir, a la mitad del tamaño del vecindario.

En este punto, queda comprobado el correcto funcionamiento del algoritmo ya que éste explora un espacio de soluciones amplio obteniendo soluciones satisfactorias en tiempo reducido. En el siguiente apartado se analiza la validez hidráulica de las soluciones obtenidas.

6.4. Evaluación hidráulica de las soluciones

Para comprobar las características hidráulicas de la red ante las distintas soluciones se han programado una serie de funciones en Python que permiten la representación en gráficas de los parámetros fundamentales. Las gráficas se generan utilizando el paquete *Matplotlib* de Python, permitiendo controlar de forma visual y rápida, el cumplimiento general de los requisitos técnicos de la red. La presión en los nodos no puede ser menor a 30 m.c.a., excepto para el nodo fuente, en el que no se impone ningún límite.

Aunque estas gráficas se extraen de todas las simulaciones realizadas, solo se muestran a continuación, las relativas a la solución encontrada en la primera simulación.

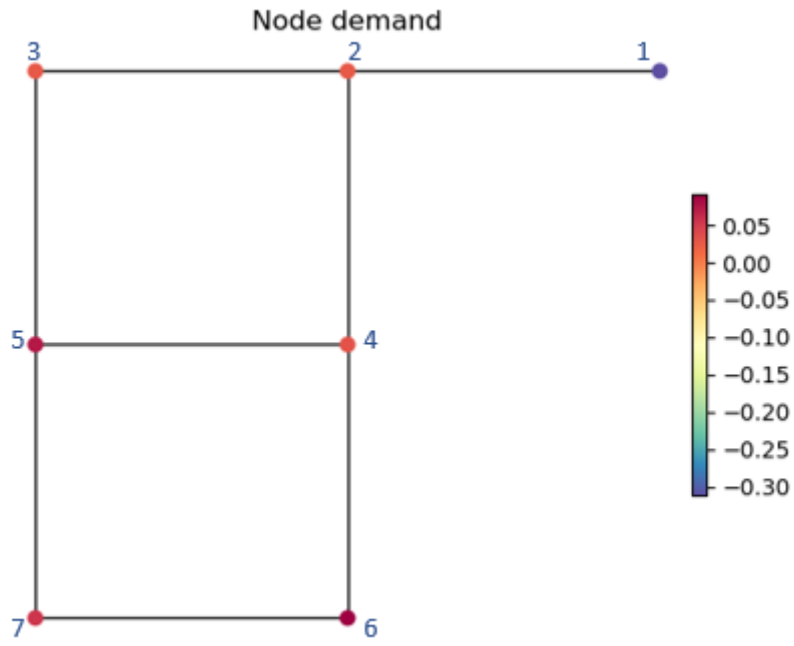


Figura 42 - Demanda de los nodos - Simulación 1
Fuente: Elaboración propia

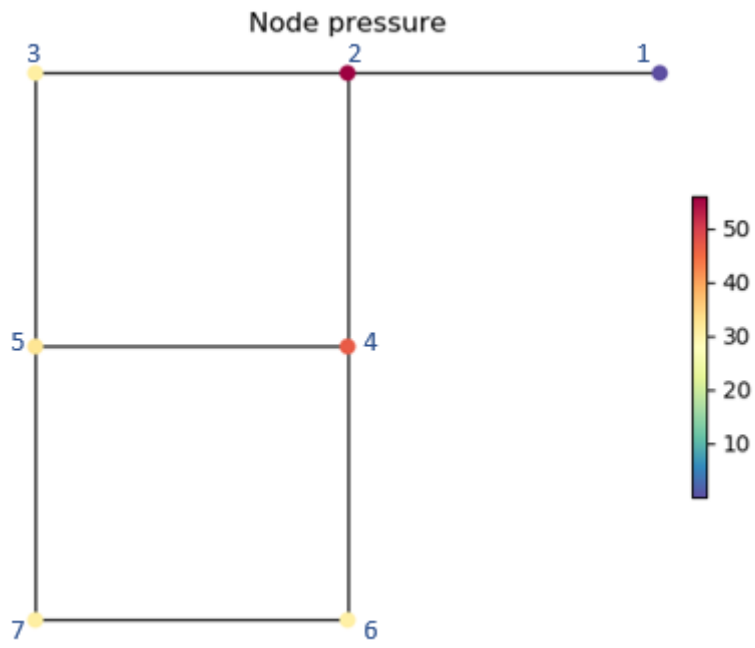


Figura 43 - Presión en los nodos - Simulación 1
Fuente: Elaboración propia

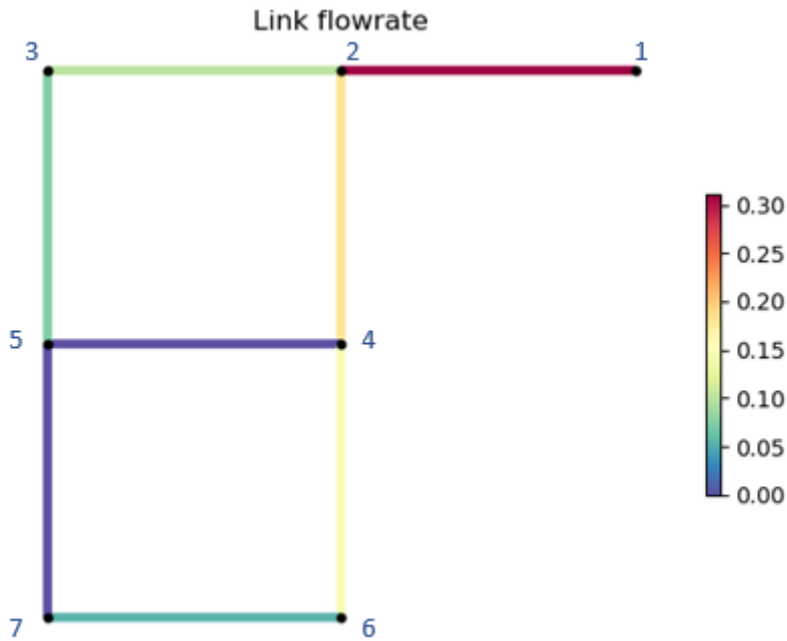


Figura 44 - Caudal en los arcos de la red - Simulación 1
Fuente: Elaboración propia

Las figuras 42, 43 y 44 muestran los parámetros característicos de la red para la mejor solución encontrada en la simulación 1. En la figura 42 se observa la demanda de cada nodo, esto era un dato de partida, no obstante, a través de esta gráfica se comprueba que dicha demanda es atendida. Recordad que las unidades de medida de la demanda son m^3/s , por lo que varía de $-0.311m^3/s$ para el nodo 1, o nodo fuente, hasta $0.092m^3/s$ para el nodo 6, nodo con mayor demanda. Un nodo con demanda negativa es un nodo que suministra agua a red, mientras que una demanda positiva significa que el nodo es consumidor de agua.

En la figura 43 se observan las presiones correspondientes a cada nodo de la red. A simple vista se comprueba que todas superan los 30 m.c.a., excepto la presión del nodo 1 que no tenía límite inferior. Se puede comprobar que los nodos 2 y 4 tienen una presión mayor al resto de nodos. Es natural que cuanto más cerca esté un nodo del nodo fuente (nodo 1), mayores presiones alcanza.

Por último, la figura 44 permite observar la cantidad de caudal que atraviesa cada arco de la red. Esta gráfica muestra el recorrido del caudal, siendo útil para comprobar que tuberías tendrán un uso mayor. En la red representada se observa que los arcos 4 y 8, transportan significativamente menos caudal que el resto de arcos, ya que son de color azul intenso.

Se procede a representar en diferentes gráficas el diámetro, el caudal y la velocidad de cada arco de la red para las distintas soluciones encontradas. Además, se representan las presiones en los nodos.

En las doce simulaciones realizadas se encuentran cinco soluciones diferentes, ya que en muchas de las simulaciones se llega a la misma solución final.

Tabla 40 - Simulaciones agrupadas según mejor solución

Simulación	Coste mejor solución [u.m.]
1, 5, 11, 12	420,000
2	448,000
3, 4, 6, 9, 10	424,000
7	482,000
8	442,000

La tabla 40 recoge las diferentes soluciones alcanzadas y las simulaciones en las que se llega a dicha solución. Como puede observarse las dos soluciones con menor coste, 420,000 u.m. y 424,000 u.m., se logran en nueve de las simulaciones, las tres simulaciones restantes alcanzan soluciones con un coste mayor. En las cuatro gráficas posteriores se presentan los valores de los resultados obtenidos para las cinco soluciones.

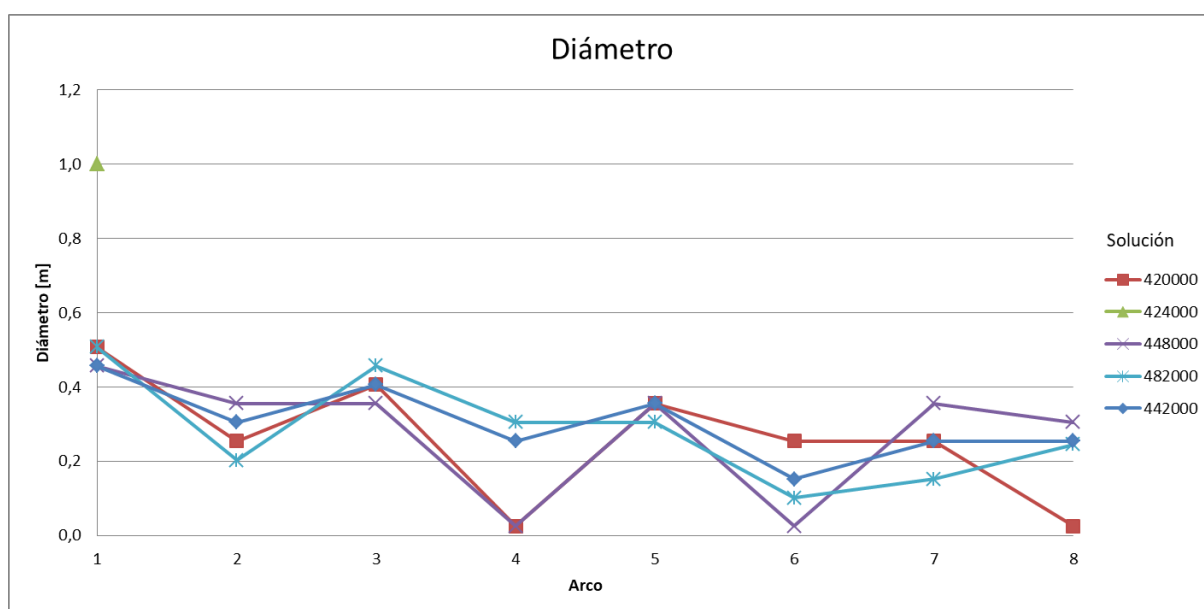


Figura 45 - Diámetro de los arcos de la red
Fuente: Elaboración propia

La figura 45 muestra los distintos diámetros asignados a cada arco de la red para las cinco soluciones alcanzadas. Se observa que las dos soluciones cuyo coste es menor, líneas roja y verde del gráfico, dos de los arcos tienen diámetros mucho menores que el resto. Esto se debe a que al optimizar una red, el resultado siempre será una red ramificada a no ser que se penalice la fiabilidad. En el caso de estudio no se permite eliminar arcos, dado que el menor diámetro de la gama comercial mide 1 pulgada, lo que es igual a 0.0254 metros. Por ello, buscando la optimización del problema se reduce al máximo los diámetros de dos de los arcos, si dichos arcos desaparecieran, quedaría una red ramificada.

Otra de las deducciones de la gráfica anterior es que al primer arco no se asocian diámetros menores a 0.4572. Al tener que transportar dicho arco todo el caudal requerido por los nodos consumidores no puede tener diámetros demasiado pequeños.

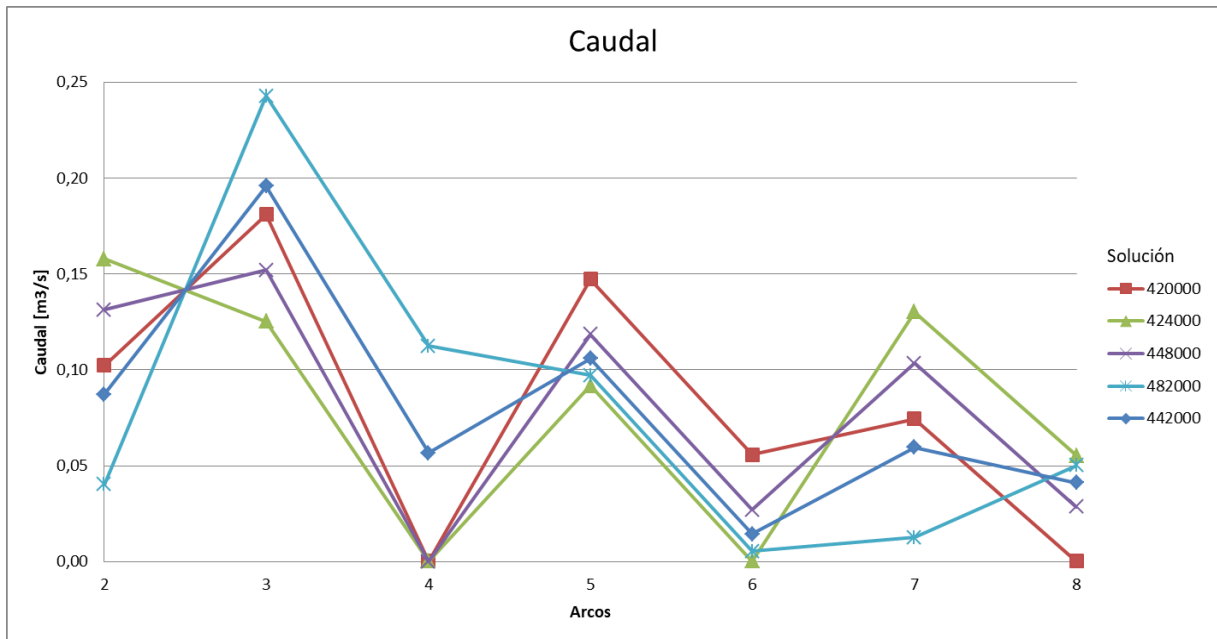


Figura 46 - Caudal de los arcos de la red
Fuente: Elaboración propia

Para una mejor visualización del rango de valores que toman los caudales, en la gráfica de la figura 46 se muestra el valor absoluto de los mismos. Los valores negativos para caudales significan que el sentido de la corriente es el contrario al establecido inicialmente. Se decide no representar el arco 1, arco que une el nodo fuente con la red, ya que el caudal que atraviesa el mismo es igual en todas las simulaciones, tomando éste un valor de $0.3111 \text{ m}^3/\text{s}$.

El agua que llega al nodo 2 desde el nodo fuente se divide obligatoriamente en dos caminos, el del arco 2 y el del arco 3. Se observa que para las mejores soluciones encontradas, la cantidad de caudal que toma cada tramo se compensa. Por el contrario, para las soluciones de peor calidad, líneas azules del gráfico, se advierte que hay mucha diferencia en la cantidad de agua que circula por los arcos 2 y 3.

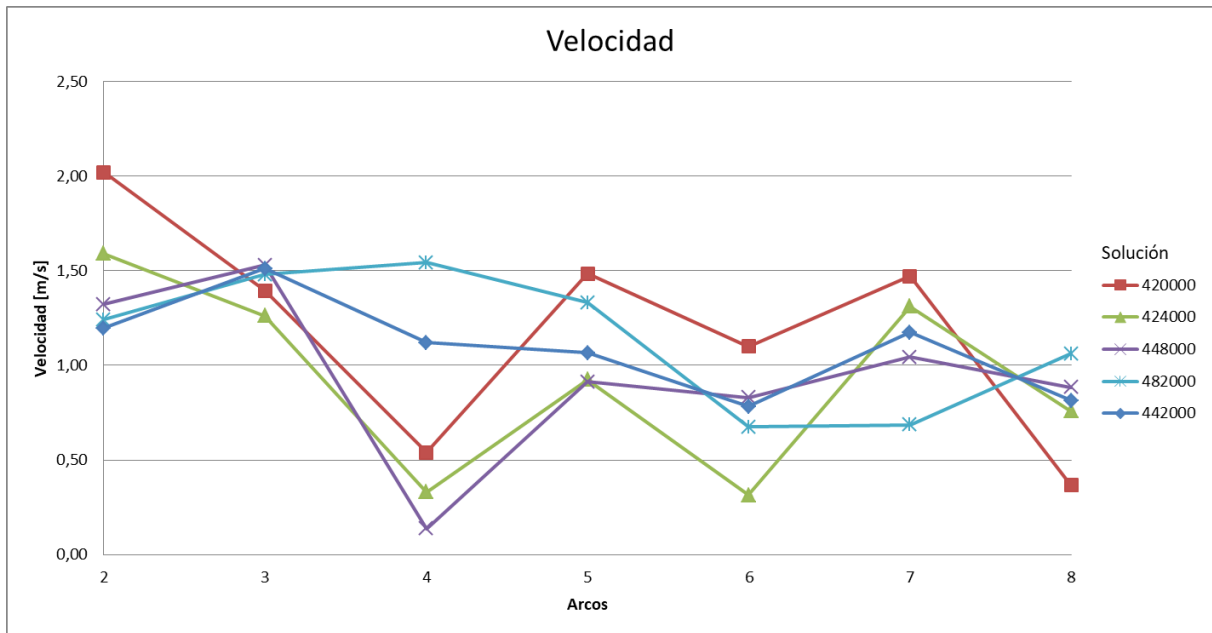


Figura 47 - Velocidad en los arcos de la red
Fuente: Elaboración propia

En la figura 47 se observa el rango en el que varía la velocidad del agua en los arcos o tuberías de la red para las diferentes soluciones obtenidas. Aunque no era una restricción del modelo, se comentó que, de forma orientativa, la velocidad del agua por las tuberías de la red debía ser mayor a 0.3 m/s y menor a 2.0 m/s. Esto evitaría estancamiento del agua y el deterioro o rotura de tuberías. Como puede observarse en la gráfica anterior, sólo en una de las soluciones, la que tiene un coste de 448,000 u.m., existe una tubería con una velocidad menor a la aconsejada. Esta solución no es de gran calidad. En ningún caso se supera el límite superior recomendado. Se puede concluir que, añadiendo las restricciones de presión en los nodos, el sistema alcanza valores de velocidad dentro del rango recomendado.

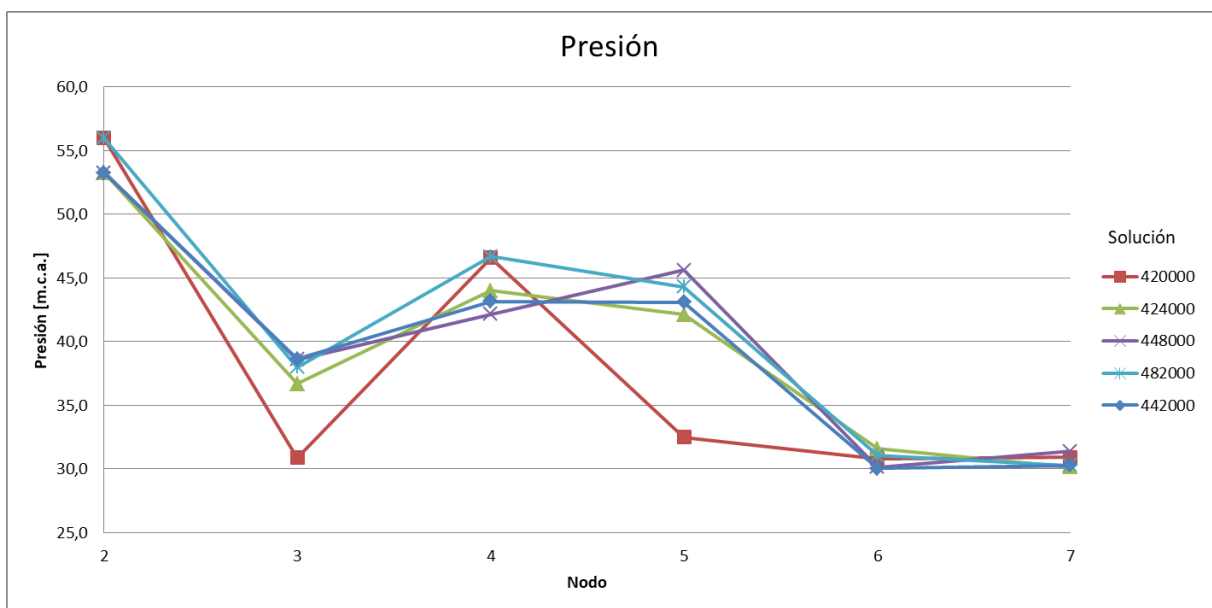


Figura 48 - Presión en los nodos de la red
Fuente: Elaboración propia

En la figura 48 se representa la presión en cada nodo de la red para las cinco soluciones, comprobándose que siempre se superan los 30 m.c.a., mínima presión de servicio establecida. La presión en el nodo 2 es mayor que la del resto de nodos, al ser el primer nodo en recibir el caudal del nodo fuente, esto era de esperar. En los nodos 6 y 7, la presión es muy cercana al límite, al ser los nodos más lejanos al nodo fuente, alcanzar una presión en ellos conllevaría un incremento en los costes. La mejor solución encontrada con coste igual a 420,000 u.m., se caracteriza porque cuatro de sus nodos tienen una presión muy cercana al límite inferior establecido. Esta solución se representa con color rojo en la cuatro gráficas anteriores. Al optimizar la red se intenta reducir los diámetros respetando las restricciones, por ello, conseguir ajustar la presión de los nodos a la mínima permitida ayuda a minimizar los costes de la red.

6.5. Comparación de los resultados con otros enfoques previos

Una vez validado el algoritmo, dado que se ha verificado que se alcanzan soluciones factibles y que la solución mejora al ajustar los parámetros del modelo, se procede a valorar la eficacia del mismo. Se revisan las soluciones halladas por otros autores, comparando la mejor solución alcanzada con el algoritmo de búsqueda tabú diseñado.

Alperovits and Shamir (1977) lograron una reducción del coste de la red del 3% pasando éste de valer 493,776 a 479,525 u.m. Más tarde, muchos autores utilizaron dicha red para verificar el funcionamiento de diferentes algoritmos de optimización y comparar los resultados con los alcanzados inicialmente.

Tabla 41 - Resultados obtenidos por otros autores para la red de Alperovits

Método	Autor	Coste Mínimo	Tipo de solución
PL	Alperovits and Shamir (1977)	479525	Tubería dividida
PL	Kessler and Shamir (1989)	418000	Tubería dividida
AG	Savic et al. (1997)	419000	Discreta
SA	Cunha et al. (1999)	419000	Discreta
AG	Matías (2003)	419000	Discreta
TS	Cuhna et al. (2004)	420000	Discreta
PSA	Suribabu et al. (2006)	419000	Discreta
MA	Baños et al. (2010)	419000	Discreta

En la tabla 41 se muestran los resultados más relevantes obtenidos por algunos de los autores que enfrentaron este problema para el caso particular de la red de Alperovits, ya que es la resuelta en este trabajo.

La primera columna de la tabla muestra las siglas del método utilizado en cada trabajo, siendo:

- PL: Programación lineal
- AG: Algoritmo genético
- SA: Recocido simulado
- TS: Búsqueda tabú

- PSA: Optimización por enjambre de partículas
- MA: Algoritmo memético

En la siguiente columna se muestran los autores y el año de la publicación, y a su derecha el coste de la mejor solución presentada por los mismos.

Por último, se define el tipo de solución. En los primeros dos trabajos mostrados en la tabla, las soluciones alcanzadas son del tipo tubería dividida, del inglés split-pipe. Cada arco de la red puede tomar dos diámetros diferentes, es decir, se dividirían los arcos en tramos de longitud variables y a cada tramo se le asigna un diámetro de la gama comercial. Aunque se observa que el menor coste es alcanzado por Kessler and Shamir (1989), este tipo de solución no es realista, ya que existen arcos donde el 99% de su longitud toma un diámetro y el 1% restante otro diferente.

Cuando surgieron las metaheurísticas las soluciones discretas se impusieron sobre las continuas. Savic et al. (1999) fueron los primeros en aplicar algoritmos genéticos al problema de diseño de redes de distribución de agua, logrando una solución cuyo coste es 419,000 u.m., considerablemente menor al coste inicial encontrado por Alperovits. No obstante, los tiempos de resolución eran muy elevados. Con el avance de la tecnología, el tiempo de resolución disminuye. Esto y el ajuste de ciertos parámetros de este metaheurística hicieron que otros autores consiguieran soluciones de la misma calidad en tiempos mucho menores [18].

La solución hallada por Cunha et al. (2004) utilizando el algoritmo de búsqueda tabú no es la mejor encontrada hasta el momento. Sin embargo, en su trabajo estos autores defienden que para redes más complejas, y por tanto más realistas, el algoritmo si logra mejorar las soluciones encontradas por otros autores que utilizan distintas técnicas.

En este trabajo se genera un algoritmo de búsqueda tabú añadiendo características nuevas al formulado por Cunha et al. (2004), y modificando otras, con el fin de mejorar los resultados obtenidos aplicando el método al caso de estudio. Se añade un criterio de aspiración y una generación aleatoria de soluciones para conseguir explorar un espacio de búsqueda mayor. A pesar de conseguir buenos resultados, no se observa una mejora con respecto a los resultados anteriores. Como se mencionó, el número de posibles configuraciones de los parámetros de la red hacen que el espacio de soluciones sea inmenso, en concreto hay $14^8 = 1,475,789,056$ posibles soluciones.

La mejor solución encontrada se consigue para un tabu_size igual a 4 y 100 iteraciones. El tiempo de resolución es 35 segundos.

Tabla 42 - Mejor solución encontrada y coste

Solución de partida	Coste [u.m.]
[0.6096,0.6096,0.6096,0.6096,0.6096,0.6096,0.6096,0.6096]	4,400,000
Solución final	Coste [u.m.]
[0.5080, 0.2540, 0.4064, 0.0254, 0.3556, 0.254, 0.254, 0.0254]	420,000

En la tabla 42 se muestra la solución de partida y la solución final de la primera simulación, donde se alcanza la mejor solución encontrada en este trabajo. Se observa que la reducción del coste es inmensa, no obstante, la solución de partida era de muy mala calidad, por lo que parece lógica esta mejora. El coste de la solución final es 420,000 u.m., es el mismo coste que el encontrado por Cunha et al. (2004).

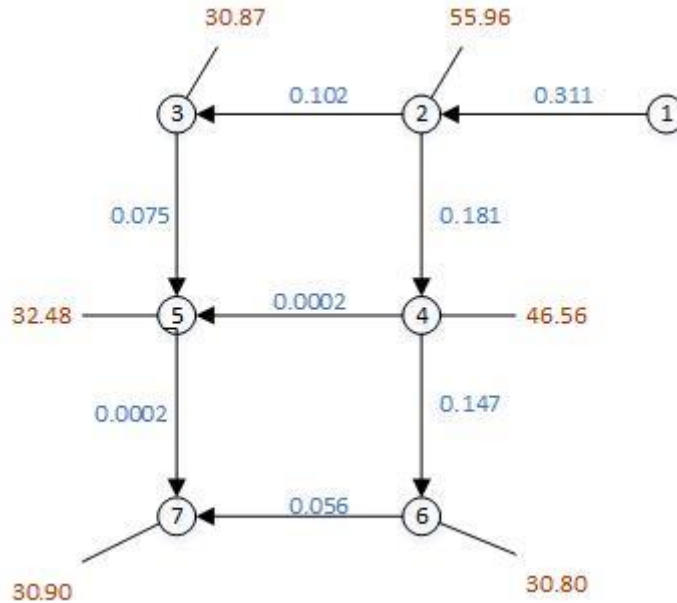


Figura 49 - Parámetros de red para la mejor solución encontrada
Fuente: Elaboración propia

Para una mejor interpretación de la solución se muestran en la figura 49 la presión en cada nodo en m.c.a., color rojo, y el caudal en m3/s de cada arco, en azul. Las presiones en los últimos nodos que reciben agua en la red, nodos 3, 5, 6 y 7 son muy cercanas al límite establecido. Un hecho esperado ya que menores diámetros significan menor coste y más pérdidas de energía. Debido a esto, para un caudal inicial fijo, al disminuir lo diámetros hay más pérdidas de presión en las tuberías y, por ello, la presión será menor a medida que el agua avanza.

7. Conclusiones

El objetivo de este trabajo fin de máster era el diseño, desarrollo y validación de un algoritmo de búsqueda tabú para la resolución del problema de diseño de redes de distribución de agua.

A lo largo del trabajo se ha explicado la problemática y su situación actual. El crecimiento demográfico de las ciudades incurre en constantes ampliaciones de las redes de distribución. Además, el desarrollo de tecnología avanzada y la aparición de nuevos materiales de construcción, junto con el deterioro que sufren las redes, hace que la reposición de tramos, o incluso redes completas, sea una tarea cada día más común.

Ante esta situación, cobra gran importancia la mejora del diseño y la planificación de las redes hidráulicas de distribución, problemática que carece de atención en la literatura científica en comparación a otros problemas de redes, como son los de red de transporte o red eléctrica.

Para introducir el tema, en este trabajo se definen los tipos de redes, sus componentes y las funciones principales que éstas desarrollan. Una vez entendida la problemática, se explica el modelo matemático tradicional, que incluye las ecuaciones de conservación de masa y energía, propias de las redes hidráulicas. Las características del modelo hacen de él un problema no lineal mixto y entero. Esto se debe a la naturaleza discreta de los diámetros, ya que éstos deben ser escogidos de entre los disponibles en una gama comercial. Y a la no linealidad de las ecuaciones de conservación de masa y energía.

El algoritmo de búsqueda tabú diseñado explora el espacio de soluciones del problema, definido por el número de variables del mismo. Para la aplicación del algoritmo a un caso concreto se escoge la red de Alperovits, red famosa en la literatura ya que muchos autores han publicado los resultados obtenidos al aplicar distintas técnicas de resolución a dicha red.

Se utiliza el lenguaje Python para programar todos los procedimientos y cálculos de resolución diseñados. Este lenguaje cuenta con multitud de librerías, las cuales sirven de soporte para realizar funciones específicas. Se realizó una búsqueda encontrando una librería llamada WNTR que permite la definición de redes hidráulicas y la resolución de las mismas. El estudio y el manejo de esta librería ha sido uno de los puntos claves del presente trabajo. Con ella se consigue generar la red de Alperovits y resolverla tantas veces como es necesario.

Con el objetivo de comprobar si el algoritmo funciona correctamente y permite obtener buenas soluciones en relación a las alcanzadas anteriormente por otros autores, se simulan diferentes escenarios. La mejor solución encontrada tiene un coste de 420,000 u.m. A pesar de lograr buenos resultados en tiempos de resolución bajos, no se consigue alcanzar la solución de menor coste encontrada en la literatura científica, 419,000 u.m.,

utilizando otros métodos de resolución. Además, no hay ninguna prueba de que esta solución sea un mínimo global del problema, es decir, es posible que dicha solución aun pueda optimizarse.

Sería de especial interés la aplicación de la técnica diseñada a redes más complejas como las presentadas en este trabajo, red de Hanoi y de New York. Para comprobar si los tiempos de resolución aumentan y si las soluciones encontradas también son de calidad en relación a las encontradas por otros autores.

Por último, se plantea como línea de trabajo a seguir en el futuro:

- La modificación el algoritmo de búsqueda tabú añadiendo memoria a largo plazo.
- El contraste de la solución alcanzada añadiendo términos de penalización a la función de evaluación de costes.
- La generación con el paquete WNTR de las redes de Hanoi y de New York con el fin de verificar el funcionamiento del algoritmo de búsqueda Tabú diseñado y comparar los resultados con los obtenidos mediante otros métodos de resolución.

8. Referencias

- [1] Datos del agua, Fundación Aqua. Enlace: <https://www.fundacionaquae.org>. Fecha de última consulta: 26.06.2018.
- [2] Programa de las Naciones Unidas para el Medio Ambiente, titulado GEO4: Perspectivas del Medio Ambiente Mundial, 2006. Datos: Shikanov y Radoo, 2003.
- [3] Instituto Español de Estadística, Agricultura y Medio Ambiente, Agua, Estadísticas sobre el suministro y saneamiento del agua, 2014, Código IOE 30077.
- [4] Enlace: <https://www.romanorumvita.com/>. Fecha de última consulta: 06.07.2018.
- [5] Instrucciones técnicas para redes de abastecimiento, PD 005 02, Revisión Nº4, EMASESA, 21 de Octubre de 2013.
- [6] Jaime Navarro & Ana Prieto, Apuntes Redes de abastecimiento de agua en poblaciones, Asignatura: Servicios Urbanos, Plan 2011, E.T.S. Arquitectura, Universidad de Sevilla.
- [7] E. Alperovits & U. Shamir, Design of Optimal Water Distribution Systems, Water Resources Research, 1977, Vol 13. No 6.
- [8] O. Fujiwara & D.B. Khang, A Two-Phase Decomposition Method for Optimal Design of Looped Water Distribution Networks, Water Resources Research, 1990, Vol 26, Issue 4, pp. 539-549.
- [9] Quindry, G . E., E. D. Brill, and J. C. Liebman, Optimization of looped water distribution systems, J. Environ. Eng. Div. Am. Soc. Civ. Eng., 1981, 107(EE4), pp. 665-679.
- [10] C. R. Suribabu & T. R. Neelakantan, Design of water distribution networks using particle swarm optimization, Urban Water Journal, June 2006, Vol. 3, No. 2, pp. 111 – 120.
- [11] Cristóbal Madero y de Miguel, Diseño y planificación de redes hidráulicas de distribución de Agua Potable, Trabajo fin de Máster, 2011.
- [12] L.F. Moody, Friction Factors for Pipe Flow, Transactions of the ASME, 1944, vol 66, pp. 671-684.

- [13] Kessler, A., Shamir, U., Analysis of the linear programming gradient method for optimal design of water supply networks. *Water Resources Research*, 1989, 25 (7), pp. 1469–1480.
- [14] Varma, K.V.K., Narasimhan, S., Bhallamudi, S.M., Optimal design of water distribution systems using an NLP method. *Journal of Environmental Engineering*, 1997, ASCE 123 (4), pp. 381–388.
- [15] Cunha, M.C., Sousa, J., Water distribution network design optimization: Simulated annealing approach. *Journal of Water Resources Planning and Management*, 1999, ASCE 125 (4), pp. 215–221.
- [16] Cunha, M.C., Sousa, J., Hydraulic infrastructures design using simulated annealing. *Journal of Infrastructure Systems*, 2001, ASCE 7 (1), pp. 32–39.
- [17] Meirelles G., Melo B., Luvizotto Jr. E., Optimal design of water supply networks using an energy recovery approach, *Renewable Energy* 117, 2018, pp. 404-413.
- [18] Matías, A., Diseño de redes de distribución de agua contemplando la fiabilidad mediante algoritmos genéticos, Ph.D. thesis, Politech, 2003, Univ. of Valencia, Valencia, Spain.
- [19] Afshar, M.H., Application of a mix–min ant system to joint layout and size optimization of pipe networks. *Engineering optimization*, 2006, 38 (3), pp. 299 – 317.
- [20] M. Cunha & L. Ribeiro, Tabu search algorithms for water network optimization, *European Journal of Operational Research* 157, 2004, pp. 746–758.
- [21] Python Software Foundation. Downloaded from: <https://www.python.org/>
- [22] Fred Glover, Future paths for integer programming and links to artificial intelligence, *Comput. & Ops. Res.* Vol 13, No. 5, 1986, pp. 533-549.
- [23] Alejandro Escudero & Pablo Aparicio, Apuntes de Modelado y optimización de problemas de gestión, E.T.S.I., Universidad de Sevilla, 2018.
- [24] PyCharm, Python IDE for professional developers. Downloaded from: <https://www.jetbrains.com/pycharm/>
- [25] Klise, K.A., Bynum, M., Moriarty, D., Murray, R. , A software framework for assessing the resilience of drinking water systems to disasters with an example earthquake case study, *Environmental Modelling y Software*, 2017, Vol 95, pp. 420-431.

- [26] Klise, K.A., Hart, D.B., Moriarty, D., Bynum, M., Murray, R., Burkhardt, J., Haxton, T., Water Network Tool for Resilience (WNTR) User Manual, U.S. Environmental Protection Agency Technical Report, 2017, EPA/600/R-17/264, 47p.
- [27] Rossman, L.A., EPANET, User's Manual, 2000 (US Environmental Protection Agency: Cincinnati).
- [28] R. Baños, C. Gil, J. Reca, F.G. Montoya, A memetic algorithm applied to the design of water distribution networks, Applied Soft Computing, 2010, Vol 10, pp. 261–266.
- [29] Savic, D.A., Walters, G.A., Genetic algorithms for least cost design of water distribution networks. Journal of Water Resources Planning and Management, 1997, ASCE 123 (2), pp. 67–77.

```

# Autor: Alicia Robles Velasco
# Date: 03.07.2018

import wntr
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from time import time
import random
import math
import operator
import copy
import pickle

def Alperovits_Network(D):
    # --- Units --- #
    # demand: m3/s
    # coordinates: m
    # elevation: m
    # length: m
    # diameters: m
    Demand=[]
    Demand_in=[100,100,120,270,330,200] # In m3/h
    for i in range(len(Demand_in)):
        Demand.append(Demand_in[i]/3600) # Converted into m3/s

    # --- Create Network --- #
    wn = wntr.network.WaterNetworkModel()
    # --- Nodes --- #
    wn.add_reservoir('res', base_head=210, head_pattern=None,
coordinates=(2000,2000))
    wn.add_junction('node2', base_demand=Demand[0],elevation=150.0,
coordinates=(1000,2000))
    wn.add_junction('node3', base_demand=Demand[1],elevation=160.0,
coordinates=(0,2000))
    wn.add_junction('node4', base_demand=Demand[2],elevation=155.0,
coordinates=(1000,1000))
    wn.add_junction('node5', base_demand=Demand[3],elevation=150.0,
coordinates=(0,1000))
    wn.add_junction('node6', base_demand=Demand[4],elevation=165.0,
coordinates=(1000,0))
    wn.add_junction('node7', base_demand=Demand[5],elevation=160.0,
coordinates=(0,0))
    # --- Links --- #
    wn.add_pipe('pipe1', 'res', 'node2', length=1000, diameter=D[0],
roughness=130,minor_loss=0.0, status='OPEN')
    wn.add_pipe('pipe2', 'node2', 'node3', length=1000, diameter=D[1],
roughness=130,minor_loss=0.0, status='OPEN')
    wn.add_pipe('pipe3', 'node2', 'node4', length=1000, diameter=D[2],

```

```

    roughness=130,minor_loss=0.0, status='OPEN')
wn.add_pipe('pipe4', 'node4', 'node5', length=1000, diameter=D[3],
roughness=130,minor_loss=0.0, status='OPEN')
wn.add_pipe('pipe5', 'node4', 'node6', length=1000, diameter=D[4],
roughness=130,minor_loss=0.0, status='OPEN')
wn.add_pipe('pipe6', 'node6', 'node7', length=1000, diameter=D[5],
roughness=130,minor_loss=0.0, status='OPEN')
wn.add_pipe('pipe7', 'node3', 'node5', length=1000, diameter=D[6],
roughness=130,minor_loss=0.0, status='OPEN')
wn.add_pipe('pipe8', 'node5', 'node7', length=1000, diameter=D[7],
roughness=130,minor_loss=0.0, status='OPEN')

return wn

def Decrease_diameter(D,i):
# --- Commercial diameters [m] --- #
Commercial_D=[0.0254,0.0508,0.0762,0.1016,0.1524,0.2032,0.254,0.30
48,0.3556,0.4064,0.4572,0.508,0.5588,0.6096]
new_D=[]
for j in range(len(D)):
    new_D.append(D[j])
    if j == i:
        del new_D[j]
        if D[j]==Commercial_D[0]:
            new_D.append(D[j])
        else:
            aux=[Commercial_D[z-1] for z in
range(1,len(Commercial_D)) if D[j] == Commercial_D[z]]
            new_D.append(aux[0])
return new_D

def Increase_diameter(D,i):
# --- Commercial diameters [m] --- #
Commercial_D=[0.0254,0.0508,0.0762,0.1016,0.1524,0.2032,0.254,0.30
48,0.3556,0.4064,0.4572,0.508,0.5588,0.6096]
new_D=[]
for j in range(len(D)):
    new_D.append(D[j])
    if j == i:
        del new_D[j]
        if D[j]==Commercial_D[13]:
            new_D.append(D[j])
        else:
            aux=[Commercial_D[z+1] for z in
range(len(Commercial_D)-1) if D[j] == Commercial_D[z]]
            new_D.append(aux[0])
return new_D

def Change_Network_Diameters(wn,D):
for i in range(1,len(wn.links)+1):
    pipe=wn.get_link('pipe'+str(i))
    pipe.diameter=D[i-1]
return wn

def Eval_Network_Technical(wn,D):
# ----- Evaluation through WNTR package: EpanetSimulator ----- #
min_H=30
wn=Change_Network_Diameters(wn,D)
epanet_sim = wntr.sim.EpanetSimulator(wn)
epanet_sim_results = epanet_sim.run_sim()
Cont=0

```

```

# ----- Constrains: Pressure in the nodes ----- #
for i in range
(len(epanet_sim_results.node['pressure'].values[0])):
    if epanet_sim_results.node['pressure'].values[0][i]>min_H:
        Cont+=1
if Cont>=(len(epanet_sim_results.node['pressure'].values[0])
-1):
    Technical_Criterium=True
else:
    Technical_Criterium=False

return Technical_Criterium

def Invalid(Invalid_solution,D):
Invalid=False
for i in range(len(Invalid_solution)):
    if D==Invalid_solution[i]:
        Invalid=True
return Invalid

def Eval_Network_Cost(D):
# ----- Commercial diameters [m] ----- #
Commercial_D=[0.0254,0.0508,0.0762,0.1016,0.1524,0.2032,
0.254,0.3048,0.3556,0.4064,0.4572,0.508,0.5588,0.6096]
# ----- Cost of diameters [u.m./m] ----- #
Cost_D=[2,5,8,11,16,23,32,50,60,90,130,170,300,550]
Cost=0
# ----- Lenth of the arcs ----- #
L=[1000,1000,1000,1000,1000,1000,1000,1000]
for i in range(len(D)):
    for j in range(len(Commercial_D)):
        if D[i]==Commercial_D[j]:
            Cost+=(Cost_D[j]*L[i])
return Cost

def Tabu_Search(D):
# ----- Parameters ----- #
tabu_size = len(D)//4
n_iteration=1000
Graph_BS=[]#To stock Best Solution
Graph_CS=[]#To stock Current Solution
Invalid_solutions=[] #To stock Invalid Solutions
wn=Alperovits_Network(D)

# ----- Initialitiation ----- #
current_solution=D[:]
current_solution_value=Eval_Network_Cost(current_solution)
best_solution=current_solution[:]
best_solution_value=current_solution_value
Graph_BS.append(best_solution_value)
Graph_CS.append(current_solution_value)
tabu_list=np.zeros([len(current_solution)],dtype=int)
Invalid_solutions=[[0 for i in range(len(current_solution))]]

for _ in range(n_iteration):
# ----- Neighbourhood ----- #
neighbourhood=[]
neighbourhood_value=[]
sec=[]
for i in range(len(current_solution)):
    neighbour=Decrease_diameter(current_solution,i)

```

```

if Invalid(Invalid_solutions,neighbour)==False:
    if Eval_Network_Technical(wn,neighbour)==True:
        # ----- Aspiration Criterion ----- #
        if tabu_list[i]==0 or Eval_Network_Cost
            (neighbour)<best_solution_value:

            neighbourhood.append(neighbour)
            neighbourhood_value.append(Eval_Network
                _Cost(neighbour))
            sec.append(i)

        else:
            Invalid_solutions.append(neighbour)

if len(neighbourhood)>0:
    current_solution_value,sec,current_solution=min(zip
        (neighbourhood_value,sec,neighbourhood))
    Graph_CS.append(current_solution_value)

    # ----- Update best solution ----- #
    if current_solution_value < best_solution_value:
        best_solution=current_solution[:]
        best_solution_value=current_solution_value
        Graph_BS.append(best_solution_value)
    # ----- Update tabu list ----- #
    tabu_list[sec]=tabu_size+1
    for i in range(len(tabu_list)):
        if tabu_list[i]>0:
            tabu_list[i]-=1
else:
    # --- Any valid solution : Increase a diameter randomly --- #
    aux=[i for i in range(len(current_solution)) if
        tabu_list[i]==0]
    random.shuffle(aux) #Random choice

    current_solution=Increase_diameter(
        current_solution,aux[0])

    current_solution_value=Eval_Network_Cost(current_solution)
    Graph_CS.append(current_solution_value)
    sec=aux[0]
    tabu_list[sec]=tabu_size+1
    for i in range(len(tabu_list)):
        if tabu_list[i]>0:
            tabu_list[i]-=1
    # ----- Graph of solutions ----- #
    Graph_solution(Graph_BS,Graph_CS)
    wn=Alperovits_Network(best_solution)
    Graph_network(wn)
    # ----- Velocity ----- #
    epanet_sim = wntr.sim.EpanetSimulator(wn)
    epanet_sim_results = epanet_sim.run_sim()
    print('Pressure results - Best
        Solution:',epanet_sim_results.node['pressure'])
    print('Flowrate results - Best
        Solution:',epanet_sim_results.link['flowrate'])

velocity=[]
velocity=[4*epanet_sim_results.link['flowrate'].values[0][i]/
    ((best_solution[i]**2)*3.14) for i in range(len(best_solution))]

```



```

print('Velocity results - Best Solution:', velocity)
# ---- Invalid Solutions ---- #
print("Número de soluciones no validas encontradas:"
      , len(Invalid solutions))

return best_solution_value, best_solution

def Graph_solution(Graph_BS, Graph_CS):
    colors = ['b', 'c', 'y', 'm', 'r', 'g', 'k', 'w']

    plt.plot(Graph_CS, color=colors[4], linewidth=1)
    plt.title("Current Solution Value")
    plt.savefig("C:/Users/alici/Documents/Universidad/Máster/TFM/
                Current_Solution.png")
    plt.show()

    plt.plot(Graph_BS, color=colors[4], linewidth=1)
    plt.title("Best Solution Value")
    plt.savefig("C:/Users/alici/Documents/Universidad/Máster/TFM/
                Best_Solution.png")
    plt.show()

def Graph_network(wn):
    epanet_sim = wntr.sim.EpanetSimulator(wn)
    epanet_sim_results = epanet_sim.run_sim()

    wntr.graphics.plot_network(wn, node_attribute = 'elevation',
                               node_size=50, title='Node elevation')
    plt.savefig("C:/Users/alicia/Documents/Universidad/Máster/TFM/
                Node_elevation.png")
    wntr.graphics.plot_network(wn, node_attribute=
                               epanet_sim_results.node['demand'].loc[0, :], node_size=50,
                               title='Node demand')
    plt.savefig("C:/Users/alicia/Documents/Universidad/Máster/TFM/
                Node_demand.png")
    wntr.graphics.plot_network(wn, node_attribute=
                               epanet_sim_results.node['pressure'].loc[0, :], node_size=50,
                               title='Node pressure')
    plt.savefig("C:/Users/alicia/Documents/Universidad/Máster/TFM/
                Node_pressure.png")
    wntr.graphics.plot_network(wn, node_attribute =
                               epanet_sim_results.node['head'].loc[0, :], node_size=50,
                               title='Node head')
    plt.savefig("C:/Users/alicia/Documents/Universidad/Máster/TFM/
                Node_head.png")
    wntr.graphics.plot_network(wn, link_attribute =
                               epanet_sim_results.link['flowrate'].loc[0, :], link_width=4,
                               title='Link flowrate')
    plt.savefig("C:/Users/alicia/Documents/Universidad/Máster/TFM/
                Link_flowrate.png")
    plt.show()

def main():
    # ---- Initial solution A ---- #
    #D=[0.6096,0.6096,0.6096,0.6096,0.6096,0.6096,0.6096,0.6096]

    # ---- Initial solution B ---- #
    D=[0.508,0.4064,0.4064,0.254,0.0508,0.3048,0.3048,0.1016]

```

```
t1=time()  
print(Tabu Search(D))  
t2=time()  
print('time execution:',t2-t1)  
  
if __name__ == "__main__":  
    main()
```