

# Herramienta para la optimización de redes de distribución de agua potable (Propuesta de proyecto final de carrera)

Gabriel Sanhueza Fuentes (Estudiante)  
Jimmy Gutierrez Bahamondes (Profesor guía)  
Carrera de Ingeniería Civil en Computación  
Universidad de Talca

6 de septiembre de 2019

## 1. Descripción de la propuesta

En el presente capítulo se presentan los conceptos básicos, el problema, su contexto, y la propuesta de solución.

### 1.1. Conceptos básicos del proyecto

En esta sección se definen algunos elementos básicos que serán necesarios para la realización del proyecto.

- **Optimización:** La optimización consiste en maximizar o minimizar un conjunto de funciones que matemáticamente pueden ser expresadas de la siguiente forma:

$$f_1(x), f_2(x), \dots, f_N(x), x = (x_1, \dots, x_d) | x \in X$$

sujeito a una serie de condiciones

$$h_j(x) = 0, j = 1, 2, \dots, J$$

$$g_k(x) \leq 0, k = 1, 2, \dots, K$$

siendo  $f_1, \dots, f_N$  funciones objetivos y  $h_j$  junto con  $g_k$  una serie de restricciones [1]. En donde de acuerdo a [2], donde  $X$  corresponde al conjunto de espacio de búsqueda, mientras que los valores obtenidos  $f_n(x)$  representa a los

valores alcanzados por cada uno de los objetivos. De acuerdo a la cantidad de funciones objetivos que se tenga, se establece que si  $N = 1$  la optimización es **mono-objetivo**, mientras que para  $N \geq 2$  es **multi-objetivo** [1]. En este punto se debe tener en cuenta que los objetivos planteados deben encontrarse en contradicción.

Debido a la definición de las restricciones es posible dividir el espacio de búsqueda en dos regiones [2]:

- Soluciones factibles: Compuesto por los elementos pertenecientes al espacio de búsqueda que satisfacen todas las restricciones.
  - Soluciones no factibles: Integrado por aquellos elementos que no cumplen todas las restricciones.
- **Heurísticas:** Métodos que permiten obtener soluciones de un problema haciendo uso de un bajo nivel de recursos y en un tiempo reducido [2].
  - **Metaheurística:** Heurísticas generalizadas para ser aplicadas en un amplio conjunto de problemas. [2].
  - **Algoritmos genéticos:** Estrategia de búsqueda de soluciones basada en la teoría de la evolución de Darwin. Para realizar esto, el algoritmo parte desde un conjunto de soluciones denominada población y iterativamente, lleva a cabo un proceso de reproducción, generando nuevas soluciones [3].
  - **NSGA-II:** Algoritmo que utiliza la cruce, mutación y reproducción para encontrar un conjunto de soluciones óptimas a problemas que cuentan con más de un objetivo [4].
  - **Epanet:** Software que permite simular el comportamiento hidráulico y la calidad del agua en redes de distribución de aguas compuesta por tuberías, nodos, bombas, válvulas y tanques de almacenamiento [5].
  - **Red de distribución de agua potable:** Conjunto de elementos enlazados de tal manera que permite suministrar cierta cantidad de agua a una presión establecida [6].

## 1.2. Contexto del proyecto

La escasez de agua potable es sin duda una problemática a nivel mundial. Dentro de este contexto, la optimización de los sistemas de distribución de agua potable es un problema sin resolver hasta la fecha.

A lo mencionado anteriormente, también se suma la escasez energética. Puesto que, es requerida para el tratamiento y distribución del agua. Es por esto que, es importante hacer un uso eficiente de la energía durante la operación del sistema.

La optimización de estos sistemas, a la vez, involucra la participación de múltiples criterios que deben ser tomados en cuenta a la hora de decidir. Sin embargo, la incorporación de estos criterios, involucra la generación de modelos cada vez más complejos.

Los algoritmos metaheurísticos han demostrado ser un mecanismo eficiente ante problemas de este tipo. Ya que, estos reducen el tiempo necesario que toma el evaluar todas las configuraciones posibles. Dado que, juzgan un conjunto menor de valores logrando como resultado una aproximación a la solución óptima.

### **1.3. Definición del problema**

Los encargados de implementar sistemas de distribución de agua potable, no cuentan con suficientes herramientas y tiempo para su correcta gestión. Por lo tanto, no es posible utilizar los recursos asociados de forma eficiente. Además, las herramientas existentes no satisfacen las necesidades de usabilidad y costo, debido a que son poco intuitivas y de pago.

El escoger las especificaciones de una red de agua potable ya es de por sí difícil debido a que hay que evaluar el rendimiento general del sistema alternando entre distintas configuraciones en busca de una solución que sea eficaz. Debido a esto, el uso de herramientas automatizadas que evalúen el rendimiento de las diversas combinaciones posibles viene a ser necesario.

A lo anterior se suma el hecho de que los interesados en esta área no manejan herramientas informáticas.

### **1.4. Trabajo relacionado**

En este apartado se mostraran algunos trabajos parecidos al que tiene como resultado este proyecto.

- **Magmoredes:** En [7] se describe la existencia de un software de diseño basado en micro-algoritmos genéticos multiobjetivos, que de acuerdo al autor, tiene un mejor rendimiento y es más eficiente debido a que requiere una menor cantidad de memoria y tiene un tiempo de computo mejor que el algoritmo NSGA-II. Este programa puede cargar cualquier red y realiza los cálculos utilizando librerías de java. Las funciones objetivos que este sistema intenta resolver son la optimización de los costo y la confiabilidad final de la red.

- **Hidra software:** Es un Software de paga que permite el diseño y la realización de distintos tipos de cálculos, como la capacidad de estanque, los recursos requeridos para la construcción de la red, demanda, diámetros, etc.
- **WaterGEMS:** De acuerdo a [8], este software permite la construcción de modelos geoespaciales; optimización de diseño, ciclos de bombeo y calibración automática del modelo; y la gestión de activos.

Los programas presentados anteriormente, resuelven problemas particulares que se dan en una red de distribución de agua potable.

### 1.5. Propuesta de solución

La solución que se propone para solventar el problema que motiva este trabajo es el desarrollo de una aplicación que sirva como una base sobre la cual poder agregar nuevas funcionalidades. Para esto la aplicación debe quedar bien documentada. Por lo tanto, lo que se tendrá al término del desarrollo del proyecto será un software escalable que tratara con dos problemas relacionados a la distribución de agua potable y que podrá ser ampliado en futuros trabajos.

Los problemas que se abordaran en el contexto de optimización de redes de agua potable serán:

- **Problema de diseño:** Se trata este problema desde un enfoque monoobjetivo implementando un algoritmo genético que buscara la optimización de los costo de inversión variando el diámetro de las tuberías.
- **Problema de operación:** Este problema se abordara desde el enfoque multiobjetivo. Para esto se implementara el algoritmo NSGA-II y se buscara la optimización de los objetivos de costos energetivos y el numero de encendidos y apagados de las bombas.

Tanto el algoritmo genético como el NSGA-II, permiten la utilización de distintos operadores de cruce y mutación que también serán implementados para ser utilizados.

La solución planteada supone además el diseño e implementación de una interfaz gráfica para ayudar al usuario en el uso de la herramienta.

## 2. Objetivos

A continuación se darán a conocer el objetivo general y los objetivos específicos que se desean lograr con el desarrollo del proyecto.

### **Objetivo general**

- Diseñar y desarrollar una aplicación de apoyo a la toma de decisiones, integrando algoritmos de optimización aplicados al problema de diseño y operación en redes de distribución de agua potable.

### **Objetivos específicos**

1. Generar soluciones frente al problema monoobjetivo de diseño de redes de distribución de agua potable a través de la implementación de algoritmos genéticos.
2. Generar soluciones frente al problema multiobjetivo de operación de redes de distribución de agua potable a través de la implementación de NSGA-II.
3. Diseñar e implementar interfaz grafica del sistema.
4. Evaluar el desempeño de los algoritmos, contrastando los resultados obtenidos en redes de benchmarking con óptimos conocidos.

## **3. Alcances**

- Este trabajo no contempla la creación de la red, por lo que estas serán ser ingresadas como entradas.
- Este trabajo solo contempla la utilización e implementación del algoritmo genético y NSGA-II.
- El producto obtenido a través de la realización de este trabajo solo sera compatible con el sistema operativo Windows. Esto se debe a que harán llamadas a librerías nativas para realizar los cálculos de las redes.

## **4. Metodología**

En esta sección se describirá y justificara la metodologías de desarrollo y evaluación que se usaran durante el desarrollo del proyecto..

### **4.1. Metodología de desarrollo**

Esta sección describe y justifica el uso de la metodología iterativo incremental.

#### **4.1.1. Iterativo incremental**

Esta metodología lleva a cabo el desarrollo de un proyecto de software dividiéndolo en iteraciones que generan un incremento, el cual contribuye en el desarrollo del producto final. Cada iteración se compone de las fases de análisis, diseño, implementación y testing. La fase de análisis lleva a cabo la obtención y definición de los requerimientos del software. La etapa de diseño se encarga de la conceptualización del software basado en los requerimientos definidos anteriormente. Durante la implementación se codifican las funcionalidades siguiendo las directivas establecidas durante el diseño, con el fin de satisfacer los requerimientos. Y finalmente, durante la fase de testing, se valida y verifica la correctitud de las funcionalidades implementadas, así como el cumplimiento de los requisitos. El hecho de llevar a cabo un desarrollo iterativo permite la obtención de retroalimentación del producto que se está desarrollando tempranamente y de esta manera poder refinar el trabajo en etapas posteriores del desarrollo. [9, 10, 11, 12].

Debido a que la metodología está pensada para ser llevada a cabo por un equipo de trabajo se adaptará la metodología para poder ser aplicada en el desarrollo llevado a cabo por una sola persona. Esta adaptación consiste en la disminución de la cantidad de la documentación generada, permitir llevar a cabo más de una fase de la iteración al mismo tiempo y los roles de analista, diseñador, implementador y tester serán realizados por una sola persona. Los documentos a generar por cada fase serán:

**Análisis:** El producto generado por esta fase será un documento de especificación de requisitos que constará de:

- Introducción
- Requisitos de usuario
- Requisito de sistema
- Matriz de trazado requisitos de usuario vs sistema.

**Diseño:** Esta fase generará como producto un documento de diseño que contará con los siguientes diagramas:

- Casos de uso
- Arquitectura lógica
- Diagrama de componentes
- Diagrama de clases

**Implementación:** Esta fase generara el código fuente de la aplicación y un manual de usuario de la aplicación.

**Pruebas:** Durante esta fase de realizaran la documentación y la realización de las pruebas sobre la aplicación.

- Se documentara las pruebas unitarias realizadas y sobre que componentes.
- Se documentara las pruebas de integración y que caso de uso cubren.

La razón por la que se utilizará esta metodología sobre otras es porque el producto resultante de este proyecto esta pensado para servir como base para futuros trabajos. Debido a esto es necesario documentar correctamente para que otros programadores puedan continuar con su desarrollo más adelante. Aunque existen otras metodologías como cascada u otras tradicionales, estas son difíciles de llevar a cabo por la cantidad de documentación que se requiere, mientras que metodologías de desarrollo ágil carecen en cuanto a la documentación que se necesita para el sistema a desarrollar. Además, esta metodología nos permite obtener una retroalimentación al final de la iteración, obtener nuevos requisitos que no hayan quedado definidos en etapas anteriores o refinar los requisitos y el diseño ya existente, permitiendo así mejorar la calidad del producto final.

La implementación de esta metodología para el desarrollo del proyecto se llevara a cabo repartiendo las tareas necesarias para el cumplimiento de los objetivos en iteraciones. De este modo al final de cada iteración se contará con un prototipo funcional de la aplicación sobre el que se agregaran las nuevas funcionalidades en las iteraciones siguientes.

## **4.2. Metodología de evaluación del proyecto**

Esta sección describe las metodologías que se aplicaran para la evaluación del proyecto. Estas serán pruebas unitarias, de integración y caso de estudio.

### **4.2.1. Pruebas unitarias**

Pruebas realizadas sobre un componente del programa, ya sean métodos u objetos. Estas pruebas se llevan a cabo variando los parámetros de entrada de los componentes para comprobar que la funcionalidad haya sido correctamente implementada. Además, deben ser realizadas aisladamente sin interacción con otros componentes o sistemas.[13].

El motivo para realizar estas pruebas sobre los componentes del sistema, es para poder detectar defectos y comprobar que el modulo se comporte de la manera esperada.

#### **4.2.2. Pruebas de integración**

Las pruebas de integración son utilizadas para comprobar las interfaces y las interacciones entre los módulos que conforman la aplicación [14]. Este tipo de pruebas se realizan después de la realización de las pruebas unitarias y permitirá verificar que los componentes interactúan adecuadamente.

#### **4.2.3. Caso de estudio**

La metodología de evaluación [15, 16] que se utilizara para la investigación es mediante casos de estudio. Un caso de estudio permite aplicar los resultados de implementación en un contexto real con el objetivo de responder la pregunta de investigación planteada. Esta metodología de evaluación considera aspectos formales para obtener evidencia. Los principales aspectos son:

- Describir el contexto de aplicación del caso
- Definición de objetivos experimentales
- Definición de características a evaluar
- Definición de sujetos de prueba
- Definición de un protocolo para conducir el caso de estudio
- Aplicación de caso de estudio en un conjunto de sesiones no controladas
- Aplicación de herramientas de obtención de evidencia empírica
- Análisis y evaluación de datos empíricos

### **5. Plan de trabajo**

En esta sección se detallara las tareas a realizar para cumplir con cada uno de los objetivos así como la fecha en que estas serán realizadas. Además, junto a este documento se adjuntada una carta Gantt que refleja este plan de trabajo.

#### **Etapas 1:** Desarrollar el objetivo 1 (12/08-07/10)

- Capturar y especificación de los requisitos (26/08-09/09)
- Construir planificación (12/08-09/09)
- Crear la arquitectura lógica (09/09-16/09)



- Especificar casos de uso (09/09-16/09)
- Crear diagrama de clases (09/09-16/09)
- Leer red de distribución de agua desde un archivo .inp (16/09-30/09)
- Parsear entrada del archivo y crear una representación de la red utilizando estructuras de datos. (16/09-30/09)
- Realizar pruebas unitarias (23/09-07/10)
- Realizar pruebas de integración (23/09-07/10)
- Entregar de incremento y primera versión de los documentos generados en cada fase, este cargara la red y la representara usando un conjunto de clases. (07/10)

**Etap 2:** Desarrollar el objetivos 2 (07/10-25/11)

- Capturar y especificación y refinación de los requisitos (07/10-14/10)
- Revisión la arquitectura lógica (07/10-14/10)
- Especificar casos de uso (07/10-14/10)
- Revisión y adaptación diagrama de clases (07/10-14/10)
- Corregir y refactorizar el código de la aplicación (07/10-28/10)
- Integrar algoritmo GA y sus operadores (07/10-11/11)
- Realizar pruebas unitarias (23/09-07/10)
- Realizar pruebas de integración (28/09-25/11)
- Entregar de incremento y segunda versión de los documentos generados en cada fase, este permitirá la ejecución del algoritmo GA a través de la terminal. (25/11)

**Etap 3:** Desarrollar el objetivos 3 (02/12-27/04)

- Capturar y especificación y refinación de los requisitos (02/12-09/12)
- Revisión la arquitectura lógica (02/12-09/12)
- Especificar casos de uso (02/12-09/12)

- Revisión y adaptación diagrama de clases (02/12-09/12)
- Corregir y refactorizar el código de la aplicación (09/12-23/12) retomando el (02/03-16/03)
- Integrar algoritmo NSGA-II y sus operadores (02/03-27/04)
- Realizar pruebas unitarias (06/04-27/04)
- Realizar pruebas de integración (06/04-27/04)
- Entregar de incremento y tercera versión de los documentos generados en cada fase, este permitirá la ejecución del algoritmo GA y NSGA-II desde la terminal (27/04)

**Etap4:** Desarrollar el objetivos 4 (27/04-06/07)

- Capturar y especificación y refinación de los requisitos (27/04-04/05)
- Revisión la arquitectura lógica (27/04-04/05)
- Especificar casos de uso (27/04-04/05)
- Revisión y adaptación diagrama de clases (27/04-04/05)
- Diseño de interfaces graficas (27/04-04/05)
- Corregir y refactorizar el código de la aplicación (27/04-18/05)
- Implementar Interfaz y los menús de interacción (04/05-18/05)
- Mostrar la representación de la red en la interfaz (11/05-01/06)
- Implementar interfaz para visualizar resultados de la simulación (11/05-01/06)
- Guardar resultados de la simulación (11/05-01/06)
- Realizar pruebas unitarias (25/05-08/06)
- Realizar pruebas de integración (25/05-08/06)
- Entregar de incremento y cuarta y última versión de los documentos generados en cada fase, este permitirá la ejecución del algoritmo GA y NSGA-II desde una interfaz gráfica (08/06)
- Realizar el caso de estudio (08/06-13/07)

## Referencias

- [1] Xin She Yang. Metaheuristic Optimization What is Metaheuristic Optimization. 6(2011):1–17, 2015.
- [2] Javier López. *Optimización multi-objetivo: aplicaciones a problemas del mundo real*. D - Editorial de la Universidad Nacional de La Plata, La Plata, ARGENTINA, 2013.
- [3] Dorothea Heiss-Czedik. An Introduction to Genetic Algorithms. *Artificial Life*, 3(1):63–65, 1997.
- [4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002.
- [5] Lewis Rossman. *EPANET 2.0 en Español. Analisis Hidraulico y de Calidad del Agua en Redes de Distribución de Agua. Manual del Usuario*. 2017.
- [6] Daniel Mora. DISEÑO DE REDES DE DISTRIBUCIÓN DE AGUA MEDIANTE ALGORITMOS EVOLUTIVOS. ANÁLISIS DE EFICIENCIA. 2012.
- [7] Pino V. Edwin, Valle C. Angely, Condori P. Franz, Mejia M. Jesus, Chavarri V. Eduardo, and Alfaro R. Luis. Diseño Óptimo de Redes de Distribución de Agua Usando Un Software Basado En Microalgoritmos Genéticos Multi-objetivos. *Ribagua*, 4(1):6–23, 2017.
- [8] Bentley. WaterGEMS® CONNECT Edition. 2017.
- [9] R Victor. Iterative and Incremental Development: A Brief History. 2003.
- [10] Susan M. Mitchell and Carolyn B. Seaman. A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review. *2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009*, (February 2008):511–515, 2009.
- [11] Robert C Martin. Iterative and Incremental Development ( IID ). *Design*, (Iid), 1999.
- [12] Adel Alshamrani and Abdullah Bahattab. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 12(1):106–111, 2015.

- [13] Ian Sommerville. *Software Engineering*. 9 edition, 2010.
- [14] Hareton K N Leung and Lee White. A study of integration testing and software regression at the integration level. *Conference on Software Maintenance*, pages 290–301, 1990.
- [15] Robert K. Yin. *Case study research : design and methods / Robert K. Yin*. 2009.
- [16] Forrest Shull, Janice Singer, and Dag I.K. Sjøberg. *Guide to advanced empirical software engineering*. 2008.