

Díálogo

19 de agosto de 2020

Buenas tardes profesores de la comisión e invitados. Mi nombre es Gabriel Sanhueza. Mis profesores guías son Jimmy Gutierrez y Daniel Mora. Y ahora les voy a presentar la defensa de mi proyecto de titulación.

1. Introducción

1.1. Contexto

Como bien se sabe la escasez de agua potable es sin duda una problemática a nivel mundial. Es por esto que optimizar los sistemas que permiten la distribución de agua es importante. Sin embargo, esto no es simple puesto que los sistemas de distribución de agua deben estar activos las 24 horas del día asegurando los niveles de servicio mínimos. Adicionalmente, la optimización de estos sistemas involucra la participación de múltiples criterios lo que dificulta la toma de decisiones respecto a éstos.

En el ámbito de los redes de agua se pueden clasificar los problemas en dos categorías. Los problemas de diseño y de operación.

1.2. Problema

Los problemas identificados con este proyecto son los siguientes:

Primero, no se cuenta con las suficientes herramientas y tiempo para la correcta gestión de las redes de distribución de agua potable (RDA). Por lo tanto, no es posible utilizar los recursos asociados de forma eficiente en estos sistemas.

Junto a lo anteriormente mencionado, escoger las especificaciones de una RDA es una tarea difícil debido a que hay que evaluar el rendimiento general del sistema en función de un conjunto de variables que se mueven en un rango muy elevado de posibilidades.

1.3. Propuesta de solución

La solución que se propone para abordar el problema consiste en el desarrollo de una aplicación de escritorio escalable que permita optimizar el proceso de diseño y operación en redes de agua potable.

Los problemas que se propone implementar son dos:

El problema de diseño basado en el costo de tuberías abordado utilizando el Algoritmo Genético.

El problema problema de operación basado en el régimen de bombeo abordado a través del algoritmo Non-Dominated Sorting Genetic Algorithm II.

La elección de estos problemas se debe a los intereses de los integrantes del equipo del proyecto que ha solicitado este trabajo.

1.4. Usuarios

Se contemplan dos tipos de usuarios:

- Usuarios que trabajan directamente con la aplicación gráfica.

- Usuarios que utilizan la estructura para acoplar nuevos problemas desde el punto de vista computacional.

1.5. Objetivo general

El objetivo general de este trabajo es el:

Diseñar y desarrollar una aplicación extensible de escritorio para optimizar el diseño y operación de una red de distribución de agua potable.

1.6. Objetivos específicos

En cuanto a los objetivos específicos se tienen los siguientes:

- Diseñar software orientado a la optimización de RDA basado en la arquitectura lógica del framework multiobjetivo Jmetal.
- Implementar un algoritmo metaheurístico de optimización monoobjetivo para aplicar al problema de diseño de RDA.
- Implementar un algoritmo metaheurístico de optimización multiobjetivo para aplicar al problema de Régimen de bombeo en RDA.
- Diseñar e implementar la interfaz gráfica del sistema de optimización de redes de agua potable desarrollado durante este proyecto.

2. Metodología de desarrollo

Se escogió la metodología “Iterativa e Incremental” para llevar a cabo el desarrollo de este proyecto.

Las adaptaciones que son realizadas sobre la metodología en general consisten en disminuir la cantidad de documentación generada, permitir la realización de más de una fase a la vez y finalmente una única persona desempeña todos los roles establecidos para la metodología.

3. Desarrollo

Ahora se procede a hablar acerca del desarrollo del proyecto.

3.1. Tecnologías a utilizar

Las tecnologías utilizadas fueron Java, como lenguaje de programación, JavaFX, para implementar la interfaz y la librería EpanetToolkit para realizar las simulaciones hidráulicas y evaluar las soluciones generadas por los algoritmos metaheurísticos.

La elección de Java como lenguaje de programación se debe a que éste fue el solicitado por el cliente, puesto que permite la incorporación directa de algunas herramientas ya desarrolladas.

3.2. Requisitos

Durante la fase de requisitos, se capturaron y priorizaron los requisitos generados en cada iteración agregando estos finalmente a un documento de especificación formal de requisitos.

En total se obtuvieron 32 requisitos de usuarios y se generó como salida de esta fase un documento en que se especifican estos requisitos siguiendo el formato mostrado en la ilustración.

3.3. Diseño

Durante la fase de diseño se realizó la conceptualización del software generando como resultado un documento de diseño en que se especifica la arquitectura física y arquitectura lógica, así como una serie de diagramas de clase y secuencia junto con los diseños de las interfaces.

Como arquitectura lógica se utilizó Modelo-Vista-Controlador la cual fue escogida porque nos permite separar los componentes de la aplicación en módulos reduciendo la dependencia entre el código de la interfaz de usuario y la lógica de la aplicación. (Y la librería JavaFX trabaja en base a esta arquitectura)

3.4. Implementación

3.4.1. Java Reflection y Java Annotation

Una de las capacidades más importantes de la aplicación que es la extensibilidad, se logró haciendo uso de dos funcionalidades de Java, las cuales son la reflexión y las anotaciones.

Las anotaciones fueron utilizadas para agregar metadatos a los operadores y a la clase que hereden de la interfaz *Registrable*. Estas anotaciones son utilizados para identificar los parámetros requeridos para cumplir con su función.

Los Experimentos son contenedores que poseen varias instancias de un mismo algoritmo para resolver un problema, esto le da a la aplicación la capacidad de realizar simulaciones independientes.

Para agregar un nuevo algoritmo, operador o problema se debe implementar la interfaz correspondiente, presentada en el diagrama y posteriormente crear una clase que herede de *Registrable*. Ésta clase es utilizada como una plantilla para la creación de los Experimentos. Posteriormente, la clase debe ser registrada en una estructura de datos presente dentro de la aplicación.

Estas dos funcionalidades permiten que el usuario que quiera agregar nuevos operadores o problemas no necesite modificar el código de la interfaz de usuario para que los nuevos elementos sean visualizados en la aplicación.

3.4.2. Problema de diseño de RDA basado en el costo de las tuberías

El primero de estos problemas consiste en el problema de diseño abordado desde el enfoque monoobjetivo. Éste, busca seleccionar los diámetros de las tuberías adecuados para formar la red. Para ello se optimiza la función del costo de inversión de las tuberías el cual se calcula como la sumatoria del costo por diámetro por el largo de la tubería.

3.4.3. Problema de operación de RDA basado en el régimen de bombeo

En cuanto al problema operación, se utiliza el enfoque multiobjetivo para determinar en que momento las bombas se encuentran encendidas y apagadas durante el día. Para ello se optimizan los costos energético, así como el costo de mantenimiento de las bombas.

El costo energético para cada bomba en un tiempo determinado se calcula como precio de la energía por el consumo energético por el estado de la bomba, tomando este último el valor 0 o 1 dependiendo si la bomba está apagada o encendida respectivamente.

En cuanto al costo de mantenimiento, éste se calcula como “el número de cambios de estado desde apagado a encendido por cada bomba en cada uno de los periodos de tiempo simulados”

3.4.4. GA

El algoritmo implementado para el problema de diseño mencionado anteriormente es el Algoritmo Genético. Los pasos para realizar este algoritmo son:

1. Crear la población inicial
2. Evaluar esta población

3. Iterar hasta alcanzar la condición de término establecida aplicando los operadores de selección y reproducción para generar una nueva población.
4. Una vez alcanzada la condición de termino se obtiene el mejor individuo de la población.

3.4.5. NSGAI

En cuanto al algoritmo implementado para el problema de operación se encuentra el algoritmo NSGAI. Éste algoritmo sigue el mismo procedimiento mencionado anteriormente para el Algoritmo Genético, pero en vez de obtener como resultado una única solución se obtiene un conjunto de ellas conocido como la Frontera de Pareto.

Una cosa importante a mencionar para este algoritmo es la función de remplazo utilizada.

Para realizar el remplazo se deben operar tanto la población original así como la descendiente generada a través del proceso de selección y reproducción. Estas dos poblaciones se agrupan y posteriormente se ordenan sus individuos en diversos Frentes formado por las soluciones que no se dominan entre sí. Posteriormente, se agregan a la nueva población las soluciones de los frentes hasta alcanzar el tope establecido. Si el último frente a agregar no cabe en su totalidad, éste debe ser ordenado en base a su densidad y las mejores soluciones son las conservadas.

3.4.6. Evidencia del desarrollo (VIDEO)

3.5. Pruebas

Durante la fase de pruebas se realizaron pruebas manuales y automatizadas, las cuales fueron especificadas en su documento respectivo.

En total se realizaron 39 pruebas automatizadas y 16 pruebas manuales.

En el documento de especificación de los casos de prueba se utiliza el formato presentado en la ilustración de ejemplo.

4. Metodología de evaluación

La metodología de evaluación escogida para aplicar sobre el sistema construido consiste en los estudios de caso.

La realización del estudio de caso se llevo a cabo dividiendo éste en cuatro etapas. Las cuales son diseño del estudio, recolección de datos, el analisis de datos y finalmente las conclusiones del estudio.

5. Evaluación

A continuación se presentan los detalles de la evaluación realizada. (Diseño, recolección, ejecución y presentación de los resultados).

5.1. Diseño del caso

Como estudio de caso se establece la aplicación desarrollada durante este proyecto y se busca responder a la pregunta “¿Cómo el sistema desarrollado trabaja en la practica?”.

Las características de la aplicación evaluadas fueron las siguientes:

- **Funcionalidad**
- **Usabilidad**
- **Utilidad**
- **Utilidad del manual de usuario**

5.2. Análisis de datos

Como resultado de la evaluación de la funcionalidad se obtuvo que el 92.3 % de las respuestas indican que los requisitos capturados fueron cumplidos.

Como resultado de la evaluación de usabilidad se obtuvo un puntaje promedio de 79.2 % con una desviación estándar de 9.5 % dando como resultado, utilizando la escala de la ilustración una evaluación de “Buena”.

En cuanto a la evaluación de la utilizada aproximadamente el 89 % de las respuestas apuntan a que la aplicación es útil para realizar tareas de optimización en redes de agua potable.

Como parte de la evaluación utilizada para evaluar el manual de usuario, el 44 % de las respuestas indican que este fue útil, mientras que el 22 % indica que no fue útil para comprender la aplicación.

6. Conclusión

Finalmente, como resultado de este proyecto se posee una aplicación escalable, la cual permite extender los algoritmos, problemas u operadores sin necesidad de conocer en detalle el funcionamiento total de la aplicación. Adicionalmente, la aplicación incorpora por defecto 2 problemas y dos algoritmos. Los cuales son el problema de diseño basado en el costo de las tuberías y el problema de operación basado en el régimen de bombeo junto con el Algoritmo Genético y el algoritmo NSGAI.

Adicionalmente, se establece que los objetivos establecidos al comienzo del desarrollo son logrados exitosamente y que la metodología aplicada era la adecuada para generar la suficiente documentación con el fin de permitir a otros desarrolladores continuar con el desarrollo del proyecto.

Finalmente, para la evaluación realizada que la aplicación cumple con los criterios de funcionalidad, usabilidad y utilizad. Mientras que para el manual de usuario de la aplicación se establece que éste debe ser mejorado, detallando más algunos temas, junto con agregar ejemplos acerca de lo mencionado.

7. Trabajo Futuro