

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

09 de septiembre de 2019

1. Descripción de la propuesta

La escasez de agua potable es un problema que esta presente a nivel mundial. Por esta razón, existe la necesidad de utilizar conscientemente el recurso hidráulico y buscar maneras de optimizar su distribución. Es en este punto, en donde se puede hacer uso de algoritmos metaheurísticos que permiten obtener soluciones buenas en un periodo de tiempo reducido a problemas presentes en la redes de distribución de agua potable. Es por ello que este proyecto busca llevar a cabo el desarrollo de una herramienta que permita la visualización y la resolución de problemas presentes en las redes de distribución de agua. Durante este proyecto se abarcara el problema de costos de diseño a través de la utilización de el algoritmo genético y del régimen de bombeo usando el algoritmo NSGA-II. Además, se busca que este sistema posea una arquitectura tal que le permita actuar como una base sobre la que sea posible ir agregando nuevos problemas de distribución de agua en futuros trabajos.

En el presente capítulo se presentan los conceptos básicos que son necesario tener claros durante el desarrollo de este proyecto. Posteriormente, se presenta el problema y contexto del proyecto. Finalmente, se presenta la propuesta de solución al problema identificado.

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:

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

sujeto 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; x_1, \dots, x_d variables de decisión, pertenecientes al espacio de búsqueda X ; y h_j junto con g_k , una serie de restricciones [1]. De acuerdo a la cantidad de funciones objetivos que se tenga, se establece que si $N = 1$ la optimización es **monoobjetivo**, mientras que para $N \geq 2$ se conoce como **multiobjetivo** [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:** Técnicas estocásticas que permitan acotar la búsqueda de soluciones en problemas cuyo espacio de búsqueda es de gran tamaño y que hacen inviable la aplicación de técnicas deterministas por el costo de tiempo que implican. Con la utilización de estas técnicas se espera encontrar soluciones buenas en un tiempo razonable, pero esto no está garantizado [1, 3].
 - **Metaheurísticas:** Algoritmos que permiten resolver un amplio rango de problemas de optimización empleando técnicas con algún grado de aleatoriedad para encontrar soluciones a un problema. Estos algoritmos no garantizan que la solución encontrada sea la óptima, pero permiten obtener generalmente aproximaciones a esta. La diferencia entre heurísticas y metaheurísticas, es que esta última puede ser aplicada a un amplio conjunto de problemas sin necesidad de realizar grandes cambios en el algoritmo, mientras que las heurísticas generalmente son aplicadas a un dominio específico [1, 4, 5].

- **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 itera iterativamente, lleva a cabo un proceso de reproducción, generando nuevas soluciones [6].
- **NSGA-II:** Algoritmo que utiliza el cruzamiento, mutación y reproducción para encontrar un conjunto de soluciones óptimas a problemas que cuentan con más de un objetivo [7].
- **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 [8].
- **Red de distribución de agua potable:** Conjunto de elementos enlazados de tal manera que permite suministrar cierta cantidad de agua a una presión establecida [9].

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 es un problema que ha sido abarcado de diversas maneras.

Otro problema que se ve afectado debido a la utilización de sistemas de distribución de agua potable es el consumo energético. Puesto que la energía, que es requerida para operar los elementos del sistema de distribución, es limitada y tiene un alto costo.

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 permiten recorrer en un tiempo razonable el espacio de búsqueda del problema en busca de buenas soluciones que idealmente están próximas a la solución óptima del problema.

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 distintas herramientas y el enfoque utilizado para resolver el problema con estas herramientas. En general, los enfoques consisten en usar software ya disponible o crear un software personalizado.

- **Magmoredes:** En [10] 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 que el algoritmo NSGA-II, debido a que requiere una menor cantidad de memoria y tiene un mejor tiempo de computo. 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.
- **WaterGEMS:** Software comercial que permite la construcción de modelos geoespaciales; optimización de diseño, ciclos de bombeo y calibración automática del modelo; y la gestión de activos. Este software a sido usado en [11] para llevar a cabo las simulaciones necesarias para su estudio. La metodología seguida para la utilización de este sistema consiste en ingresar los datos a WaterGEMS para correr las simulaciones. La limitación de este programa es que no permite la adición de nuevos algoritmos por parte del usuario, en el caso de que se quiera probar o mejorar algún algoritmo ya existente. Sin embargo, este sistema también tiene sus ventajas, por que ya incorpora algunos algoritmos predefinidos para resolver ciertos problemas. Además, de que cuenta con diversas funcionalidades como conexión con datos externos, operaciones de análisis espaciales, intercambio de datos con dispositivos o programas de administración, entre otros.
- **EPANET:** El enfoque seguido con la utilización de esta herramienta consiste en la automatización al momento de ejecutar los algoritmos y la posterior evaluación de los resultados utilizando la librería EPANET Toolkit. Este es

usado en [9] y consiste en la creación de un programa el cual implementa algoritmos metaheurísticos y los resultados obtenidos por estos son enviados a la EPANET Toolkit para evaluar la solución y determinar la factibilidad de esta. La ventaja de este enfoque es que permite una mayor flexibilidad en los algoritmos metaheurísticos utilizados y los problemas que se quieren resolver. Sin embargo, debido a que se necesita implementar los problemas y los algoritmos, este enfoque toma mucho tiempo.

Para el desarrollo de este proyecto se usa el enfoque basado en EPANET, puesto que esta ya es una librería de simulación hidráulica y permite enfocarnos en la resolución de los problemas usando algoritmos metaheurísticos. Tanto Magmoredes como WaterGEMS buscan resolver temas concretos en los sistemas de distribución de agua potable y puesto que el código de estos programas no está disponible públicamente o son un sistema que se comercializa sin permitir la modificación del sistema por parte de terceros, se busca con nuestro proyecto incorporar esta capacidad para que en futuros trabajos se pueda abarcar una mayor cantidad de problemas con nuestro sistema.

1.5. Propuesta de solución

La solución que se propone para abordar el problema consiste en el desarrollo de una aplicación que que permita buscar soluciones a dos de los problemas existentes en las redes de agua potable. Además, de hacer uso de una arquitectura que permita fácilmente extender el programa para abarcar nuevos problemas. Para lograr esto último la aplicación debe quedar bien documentada. Por lo tanto, lo que se tendrá al término del desarrollo del proyecto será un software escalable que tratará con dos problemas relacionados a la distribución de agua potable y que podrá ser ampliado en futuros trabajos.

Los problemas que se abordarán 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 buscará la optimización de los costos de inversión variando el diámetro de las tuberías.
- **Problema de operación:** Este problema se abordará desde el enfoque multiobjetivo. Para esto se implementará el algoritmo NSGA-II y se buscará la optimización de los objetivos de costos energéticos y el número de encendidos y apagados de las bombas.

Tanto el algoritmo genético como el NSGA-II, permiten la utilización de distintos operadores de cruzamiento 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.

2.0.0.1 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.

2.0.0.2 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 la interfaz gráfica del sistema de optimización de redes de agua potable desarrollado durante este proyecto.
4. Evaluar el desempeño de los algoritmos, contrastando los resultados obtenidos en redes de benchmarking con óptimos conocidos.

3. Alcances

Al final del periodo de desarrollo la herramienta contara con las siguientes prestaciones.

- El sistema permitirá la carga y la visualización de la red gráficamente.
- El sistema permitirá visualizar la configuración básica almacenada en el archivo inp de los elementos de la red.

- El sistema solo resolverá dos clases de problemas de optimización, uno mono-objetivo y el otro multiobjetivo. El problema monoobjetivo será el de los costos de inversión. En cuanto al problema multiobjetivo, este será el de los costos energéticos y el número de encendidos y apagado de las bombas.
- El sistema únicamente contara con dos algoritmos implementados los cuales serán el algoritmo genético y NSGA-II. El algoritmo genético será el usado para tratar el problema monoobjetivo, mientras que NSGA-II será aplicado al multiobjetivo.
- El sistema permitirá visualizar el o los resultados obtenidos al finalizar la ejecución del algoritmo.
- El sistema permitirá generar archivos .inp con la configuración de la solución obtenida a través de los algoritmos. Debido a que el archivo .inp establece una gran cantidad de configuraciones, las únicas que se permitirán modificar serán las configuraciones asociadas a los conexiones (junctions) y tuberías (pipes).
- El sistema permitirá generar archivos con las soluciones obtenidas para cada problema, es decir, el valor de los objetivos y las variables de decisión involucradas.
- El sistema permitirá la visualización de una gráfica la cual permitirá visualizar la variación de la solución durante la ejecución de los algoritmos.
- La grafica unicamente estará disponible en problemas con uno o dos objetivos.
- La comparación con las redes de benchmarking consistirá únicamente, en presentar una tabla comparativa entre los resultados presentes en la literatura y los obtenidos a través de nuestro sistema, para cada uno de los problemas de redes de distribución de agua potable de nuestro sistema.

Este proyecto no contempla la creación de la red por lo que estas deberán ser ingresadas como entradas al programa, es decir, estas deberán ser creadas usando EPANET o manualmente, pero siguiendo el formato establecido por EPANET. Además, esta herramienta únicamente podrá ser ocupada en equipos cuyo sistema operativo sea Windows debido a que se realizan llamadas a librerías nativas.

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 esta desarrollando tempranamente y de esta manera poder refinar el trabajo en etapas posteriores del desarrollo. [12, 13, 14, 15].

Debido a que la metodología esta pensada para ser llevada a cabo por un equipo de trabajo se adaptara 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 mas de una fase de la iteración al mismo tiempo y los roles de analista, diseñador, implementador y tester sera realizado por una sola persona. Los documentos a generar por cada fase serán:

4.1.1.1 Análisis:

El producto generado por esta fase sera un documento de especificación de requisitos que constara de:

- Introducción: En este apartado del documento se hará una introducción al problema que se ha identificado.
- Requisitos de usuario: Consiste en la recopilación de lo requisitos de los usuarios que deben ser cumplidos al final del periodo de desarrollo.

- Requisito de sistema: Son los requisitos, desde un punto de vista mas técnico, que son necesarios para satisfacer los requisitos de usuario.
- Matriz de trazado requisitos de usuario vs sistema: Matriz que permite ver la trazabilidad de los requisitos de usuario con los de sistema.

4.1.1.2 Diseño:

Esta fase generara como producto un documento de diseño que contara con los siguientes diagramas:

- Casos de uso: Serie de diagramas que permiten ver la interacción que el usuario tendrá con el sistema.
- Arquitectura lógica: Descripción a alto nivel del software y los componentes que lo componen.
- Diagrama de componentes: Permite ver la división del sistema y la interacción entre los distintos componentes [16].
- Diagrama de clases: Describe la relación entre las distintas clases presentes en la solución propuesta.

4.1.1.3 Implementación:

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

4.1.1.4 Pruebas:

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

- Se documentará las pruebas unitarias realizadas y sobre que componentes.
- Se documentará 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 llevará 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 aplicarán 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.[17].

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 [18]. 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 [19] 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

Para llevar a cabo la realización del caso de estudio sobre este sistema, se enviara el programa para que sea probado a personal de la Universidad Politécnico de Valencia junto con un manual de uso. Posteriormente, se llevara a cabo la recolección de datos usando una técnica de segundo grado, la cual sera una encuesta de preguntas cerradas, que deberá ser respondida por aquellos que hayan hecho uso del software.

Como caso de estudio se tendrá el proyecto desarrollado. Mientras que el objetivo de la realización del caso de estudio será el determinar como fue recibido el proyecto, es decir, si los sujetos encontraron este útil y fácil de usar. En cuanto a los sujetos de prueba, estos serán el personal que haya utilizado el sistema desarrollado. La pregunta de investigación a la cual se buscara dar respuesta es:

¿Como fue recibido el software por parte de profesionales del área que trabajan en redes de distribución de agua potable?

Debido a que el tiempo disponible para el desarrollo de este proyecto es bajo, se realizara un estudio de caso simplificado. Este estudio de caso se llevara a cabo utilizando unicamente métodos cuantitativos. Se contara con una sola fuente de datos y se simplificara el protocolo para conducir el caso de estudio.

5. Plan de trabajo

En esta sección de detallarán 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.

5.0.0.1 Etapa 1:

Desarrollar el objetivo 1 (12/08-07/10)

- Implementar algoritmo genético
- Implementar operadores de selección, cruzamiento, mutación.
- Evaluar las soluciones generadas con el algoritmo genético usando Epanet Toolkit a través del modulo de EPANET.

5.0.0.2 Etapa 2:

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

- Implementar algoritmo NSGA-II.
- Implementar o adaptar los operadores de cruzamiento y mutación para ser usados con el algoritmo NSGA-II.
- Evaluar, usando el modulo de Epanet, las soluciones generadas a traves del algoritmo NSGA-II.

5.0.0.3 Etapa 3:

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

- Diseñar la interfaz gráfica e implementarla con el fin de facilitar la interacción del usuario con la aplicación.
- Probar la aplicación con usuarios para obtener una retroalimentación acerca de la interfaz y su usabilidad.

5.0.0.4 Etapa 4:

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

- Comparar los resultados obtenidos en el problema de diseño de redes, usando el algoritmo genético, con resultados presentados por diversos autores, que han tratado el mismo problema a través de otros algoritmos.
- Comparar los resultados obtenidos en el problema de operación, usando el algoritmo NSGA-II, con resultados presentados en la literatura.

Referencias

- [1] Xin She Yang. Metaheuristic Optimization. *Scholarpedia*, 6(2011):11472, 2015.
- [2] Omid Bozorg-Haddad, Mohammad Solgi, and Hugo A Loáiciga. *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. John Wiley & Sons, Incorporated, Newark, UNITED STATES, 2017.
- [3] Marc H.J. Romanycia and Francis Jeffry Pelletier. What is a heuristic? *Computational Intelligence*, 1(1):47–58, 1985.
- [4] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013.
- [5] Sean Luke. *Essentials of Metaheuristics*. Lulu, second edition, 2013.
- [6] Dorothea Heiss-Czedik. An Introduction to Genetic Algorithms. *Artificial Life*, 3(1):63–65, 1997.
- [7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002.
- [8] Lewis Rossman. *EPANET 2.0 en Español. Analisis Hidraulico y de Calidad del Agua en Redes de Distribución de Agua. Manual del Usuario*. 2017.
- [9] Daniel Mora. DISEÑO DE REDES DE DISTRIBUCIÓN DE AGUA MEDIANTE ALGORITMOS EVOLUTIVOS. ANÁLISIS DE EFICIENCIA. 2012.
- [10] 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.
- [11] Darshan Mehta, Vipin Yadav, Keyur Prajapati, and Sahita Waikhom. Design of optimal water distribution systems using WaterGEMS: a case study of Surat city. *E-proceedings of the 37th IAHR World Congress*, (December):1–8, 2017.
- [12] R Victor. Iterative and Incremental Development: A Brief History. 2003.

- [13] 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.
- [14] Robert C Martin. Iterative and Incremental Development (IID). *Design*, (Iid), 1999.
- [15] 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.
- [16] Donald Bell. UML basics : The component diagram. *IBM developerWorks*, (December):1–10, 2004.
- [17] Ian Sommerville. *Software Engineering*. 9 edition, 2010.
- [18] 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.
- [19] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.