Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

TFG del Grado en Ingeniería Informática

**Sibling Rewiring 2.0**

Optimización multiobjetivo para reducir el riesgo de contagio en entornos escolares

Imagen que contiene Logotipo

Descripción generada automáticamente

Presentado por Rubén Arasti Blanco

en la Universidad de Burgos — 5 de julio de 2024

Tutores: Dr. José Manuel Galán Ordax

y Dra. Virginia Ahedo García

# Índice General

[Índice General 1](#_Toc169714649)

[Índice de Ilustraciones 2](#_Toc169714650)

[1. Introducción 3](#_Toc169714651)

[2. Objetivos del proyecto 4](#_Toc169714652)

[3. Conceptos teóricos 5](#_Toc169714653)

[3.1 Definición del problema 5](#_Toc169714654)

[3.2 Resolución del problema 5](#_Toc169714655)

[3.3 Aproximaciones para la Optimización Multiobjetivo 6](#_Toc169714656)

[3.4 Algoritmo genético 7](#_Toc169714657)

[3.4 Estrategia NSGA-II para optimización multiobjetivo 8](#_Toc169714658)

[4. Técnicas y herramientas 8](#_Toc169714659)

[Entorno de desarrollo 8](#_Toc169714660)

[Gestor de proyectos 8](#_Toc169714661)

[Documentación 8](#_Toc169714662)

[5. Aspectos relevantes en el desarrollo del proyecto 8](#_Toc169714663)

[6. Trabajos relacionados 9](#_Toc169714664)

[7. Conclusiones y Líneas de trabajo futuras 9](#_Toc169714665)

[Bibliografía 9](#_Toc169714666)

# Índice de Ilustraciones

[Ilustración 1: Ejemplo gráfico del frente de Pareto 6](#_Toc166662746)

# 1. Introducción

La pandemia del COVID-19 ha resaltado la importancia de implementar medidas eficaces para minimizar la transmisión de enfermedades en diversos entornos, incluyendo las instituciones educativas. En este contexto, las escuelas enfrentan el desafío de organizar a los estudiantes de manera que se reduzcan los riesgos de contagio.

Una de las estrategias que se implantaron durante la pandemia fue la implementación de los llamados grupos burbuja [1]. Estos grupos consisten en conjuntos de alumnos y profesores que interactúan exclusivamente entre ellos. Esta medida permite que los miembros de un mismo grupo compartan asistencia, espacio y actividad, mientras se limita estrictamente el contacto con personas de otros grupos. El objetivo principal de esta estrategia es contener y reducir la propagación del virus al minimizar las interacciones entre distintos grupos de estudiantes, haciendo más fácil el rastreo y la contención de los posibles brotes.

En esta implementación, se juntaba a los hermanos que estuvieran en la misma etapa y curso en el mismo grupo. Esta idea parte de la premisa de que los hermanos, al convivir en el mismo hogar, tenían una alta probabilidad de contagiarse mutuamente, y, por tanto, deberían estar en la misma clase para evitar la transmisión entre diferentes grupos.

Sin embargo, esta medida no contempló adecuadamente la situación de los hermanos que no estuvieran en la misma etapa o curso. Al pertenecer a clases distintas, estos hermanos seguían representando un potencial riesgo entre grupos diferentes, lo que minaba la efectividad de los grupos burbuja.

Sibling Rewiring (SiRe) es una aplicación diseñada para optimizar la organización de alumnos en grupos burbuja con el objetivo de reducir el riesgo de contagio. Fue desarrollada como Trabajo de Fin de Grado por María Ojeda Ruiz en la Universidad de Burgos en 2021 [2]. En esta continuación del trabajo original, se introducirán nuevas funcionalidades y mejoras significativas que culminarán en Sibling Rewiring 2.0.

El proyecto se basa en el marco conceptual de redes complejas [3]. Se han modelado dos redes principales. La primera, incluye a todos los alumnos como nodos con atributos de grupo, etapa y curso, y los enlaces representan sus interacciones: la pertenencia a la misma clase o al mismo hogar. La segunda, tiene como nodos las distintas clases del colegio y como enlaces las relaciones de hermanos entre las clases, siendo los pesos de los enlaces el número de relaciones y pudiendo existir auto enlaces.

Para optimizar la organización de los grupos, el sistema busca que los grupos burbuja estén bien compartimentados y que el riesgo de contagio entre ellos sea homogéneo, evitando penalizar a un grupo en detrimento de otro.

En la red de clases se persiguen tres objetivos:

1. Maximizar el número de componentes desconectados. Se busca maximizar el número de componentes o conjuntos de clases que no tienen enlaces entre sí, lo que minimiza la posibilidad de transmisión de enfermedades entre diferentes grupos burbuja.
2. Minimizar la variación en el tamaño de los componentes. Se busca que los componentes tengan tamaños similares.
3. Minimizar la variación en el número de enlaces dentro de cada componente. Se busca que la densidad de conexiones dentro de cada partición sea uniforme.

El riesgo de contagio en un componente es proporcional a su tamaño y a su número de relaciones entre hermanos, que se calcula como la suma de las ponderaciones de los enlaces.

Naturalmente, la metodología desarrollada en este proyecto no sólo es aplicable a la contención del COVID-19, sino a cualquier enfermedad contagiosa, como la gripe o la varicela. Esta versatilidad hace que el sistema sea una herramienta valiosa para la prevención y control de diversas infecciones en cualquier situación epidemiológica.

## Estructura de la memoria

## 1.2 Materiales adjuntos

# 2. Objetivos del proyecto

A continuación, se detallan los objetivos del proyecto en tres categorías principales: generales, técnicos y personales.

## 2.1 Objetivos generales

* Desarrollar una aplicación web para optimizar la organización de los alumnos en un colegio y reducir el riesgo de contagio.
* Implementar un algoritmo genético como método de optimización multiobjetivo para aportar diversidad de soluciones al problema.
* Mostrar gráficamente la evaluación de distintas soluciones exploradas y resaltar el frente de Pareto.
* Aportar una solución intuitiva para el usuario y con información relevante para la toma de decisiones.

## 2.2 Objetivos técnicos

* Desarrollar un algoritmo genético mediante el framework DEAP.
* Trabajar con la librería NetworkX para el manejo y análisis de redes.
* Desplegar la aplicación web en Netlify.
* Utilizar Git como herramienta de control de versiones.
* Utilizar Trello como herramienta de gestor de proyectos.
* Aplicar la metodología ágil SCRUM para la gestión del proyecto.

## 2.3 Objetivos personales

* Consolidar conocimientos sobre algoritmos genéticos y la optimización multiobjetivo.
* Adquirir experiencia en metodologías ágiles.
* Aprender el proceso de despliegue de una web.
* Poner en práctica los conocimientos de la ingeniería de software adquiridos durante la carrera.

# 3. Conceptos teóricos

## 3.1 Definición del problema

En este apartado se explicará cuál es el problema por resolver y qué datos se utilizarán para su resolución.

Se intenta reducir la probabilidad de contagio del virus SARS-CoV-2 entre los alumnos de un colegio. Cada estudiante tiene un nombre (identificador numérico), una etapa (infantil o primaria), un curso (1º, 2º, 3º, …) y un grupo (A, B, C, …). Llamaremos hermanos a los estudiantes que viven juntos.

De esta manera, cada estudiante se relaciona con los demás estudiantes de su clase y con sus hermanos si tiene. Para formalizar esto, definiremos un grafo con los estudiantes como nodos y sus relaciones con otros estudiantes como enlaces. Cada nodo tendrá los atributos nombre, etapa, curso y grupo.

Para saber quién es hermano de quién, definiremos una matriz con los atributos de cada estudiante y el nombre de su hermano en cada fila.

## 3.2 Resolución del problema

En este apartado se explicará cómo se tratarán los datos del problema para resolverlo. Para resolver el problema, cambiaremos a los hermanos de grupo de forma de que se reduzcan los enlaces entre estudiantes de distintas clases pero procurando que todas las clases tengan un riesgo similar de contagio.

Para hacer esto, definiremos un grafo cuyos nodos serán las clases y cuyos enlaces serán las relaciones de hermanos entre clases. Es decir, si un hermano de la clase 1º de primaria A tiene un hermano en la clase 2º de primaria B, los nodos “primaria1A” y “primaria2B” estarán conectados entre sí. Cada nodo tiene como atributos un nombre (cadena con la etapa, el curso y el grupo), una etapa, un curso, un grupo y unos estudiantes (lista con los nombres de los estudiantes de esa clase).

El primer objetivo es reducir el número de enlaces entre clases o aumentar el número de componentes del grafo. Por lo tanto, para optimizar el primer objetivo, se buscará maximizar el número de componentes del grafo de clases.

Sin embargo, este objetivo por sí solo causaría que todos los hermanos fueran enviados al mismo grupo X en todos los cursos, creando un desbalance en el riesgo de contagio: los alumnos de las demás grupos tendrían un riesgo muy bajo, mientras que los del grupo X tendrían un riesgo muy alto.

Para solucionar esto, el segundo objetivo buscará que el tamaño de los compontes sea parecido. Entonces, para optimizar el segundo objetivo, se buscará minimizar la varianza entre los tamaños de los componentes.

## 3.3 Aproximaciones para la Optimización Multiobjetivo

Como se ha dicho en el anterior apartado, el primer objetivo será maximizar el número de componentes del grafo de clases, el segundo objetivo será minimizar la varianza entre los tamaños de estos componentes y el tercer objetivo será minimizar la varianza entre el número de enlaces de cada componente.

Para comparar las distintas soluciones en este tipo de optimización, se suele optar por tres aproximaciones: la combinación de objetivos, la priorización de objetivos y el óptimo de Pareto.

La combinación de objetivos consiste en crear una única función objetivo resultado de la suma de las demás funciones objetivo. Se pueden ponderar los objetivos para que tengan la misma relevancia y multiplicar por -1 dependiendo si un objetivo se quiere minimizar o maximizar.

La priorización de objetivos consiste en establecer un orden para los objetivos de forma que dos soluciones tienen su primer objetivo igualado, se comparen mediante el segundo objetivo.

Estos dos últimos métodos sólo permiten ver una solución final, a pesar de que pueda haber varias soluciones con igual beneficio pero distintos valores para cada objetivo.

El óptimo de Pareto se define de la siguiente manera:

Una solución xi domina otra solución xj si las dos siguientes condiciones se cumplen:

1. La solución xi es igual o mejor que xj para todas las funciones objetivo
2. Gráfico, Gráfico de cajas y bigotes

   Descripción generada automáticamenteLa solución xi es estrictamente mejor que xj en al menos un objetivo

Ilustración 1: Ejemplo gráfico del frente de Pareto

En la ilustración 1 se muestra una gráfica donde los ejes representan los valores de dos funciones objetivo ζ1(x) y ζ2(x). En este ejemplo las dos funciones se quieren minimizar. Se puede ver como la solución 1 domina a la 3 porque tiene valores menores al menos en el primer objetivo y es menor o igual en el segundo objetivo. La solución 1 no domina a la 2 ni es dominada por esta, debido a que es mejor en el primer objetivo pero peor en el segundo. La solución 3 no domina a la 2 ni es dominada por esta por la misma razón que la comparación anterior. Se puede resumir en que las solución 1 y la solución 2 son no dominadas porque ninguna de ellas es superada en ambos objetivos simultáneamente por otra solución.

El conjunto de soluciones que no son dominadas por ninguna otra se denomina el frente de Pareto y representa las mejores soluciones encontradas. Esta aproximación aporta diversidad de soluciones y es la que utilizaré para la resolución del problema.

## 3.4 Algoritmo genético

CONCEPTOS DE CONVERGENCIA DIVERSIDAD Y EXPLORACIÓN.

### 3.4.1 Representación del genotipo

El genotipo será una lista de tamaño igual al número de hermanos. Cada elemento de la lista será un valor entre 0 y el número de grupos menos 1. Estos elementos representarán el grupo que se le asignará a cada hermano. Para saber a qué hermano se le asignará, se utilizará la posición de la lista.

### 3.4.2 Representación del fenotipo

### 3.4.3 Traducción del genotipo a fenotipo

### 3.4.4 Función de adaptación

### 3.4.5 Operadores genéticos

La probabilidad de cruce controla la proporción de soluciones que se someten a la recombinación genética. Si es muy alto puede causar una pérdida de diversidad genética y a una convergencia prematura y si es muy bajo puede ralentizar la capacidad de exploración del algoritmo. La mutación es sólo un operador secundario para restaurar el material genético. Sin embargo, la probabilidad de mutación es un parámetro crítico para el desempeño del algoritmo y altos valores de este parámetro transforman al algoritmo genético en una búsqueda aleatoria. La mutación previene al algoritmo de la convergencia prematura. Según la literatura científica, los valores de suelen estar entre 0.5 y 1.0 y los valores de suelen estar entre 0.005 y 0.05. [4]

Para los valores por defecto se escogerán 0.6 para , que no es un valor muy elevado y nos asegura una correcta exploración, y 0.05 para , que nos garantiza que hay suficiente diversidad genética.

## 3.4 Estrategia NSGA-II para optimización multiobjetivo

Para llevar a cabo la optimización del problema, se utilizará un algoritmo genético. Para la selección de los individuos se utilizará NSGA-II [5], que es un

# 4. Técnicas y herramientas

## Entorno de desarrollo

## Gestor de proyectos

He elegido Trello como gestor de proyectos. Es una aplicación web gratuita y fácil de utilizar.

## Documentación

La fuente tipográfica escogida es EB Garamond que tiene una licencia SIL Open Font License.[6]

# 5. Aspectos relevantes en el desarrollo del proyecto

# 6. Trabajos relacionados

# 7. Conclusiones y Líneas de trabajo futuras

# Bibliografía

[1] «¿Qué son los grupos burbuja? | Glosario Covid». Accedido: 20 de junio de 2024. [En línea]. Disponible en: https://www.unilabs.es/glosario/grupos-burbuja

[2] «Mariaojruiz/Sibling-Rewiring: Proyecto de la universidad de Burgos tutelado por José Manuel Galán y Virginia Ahedo. Aplicación que realizará las modificaciones necesarias en las aulas para disminuir el número de contagios entre los alumnos al máximo.» Accedido: 20 de junio de 2024. [En línea]. Disponible en: https://github.com/Mariaojruiz/Sibling-Rewiring

[3] M. E. J. Newman, «The Structure and Function of Complex Networks», *SIAM Rev.*, vol. 45, n.o 2, pp. 167-256, ene. 2003, doi: 10.1137/S003614450342480.

[4] M. Srinivas y L. M. Patnaik, «Adaptive probabilities of crossover and mutation in genetic algorithms», *IEEE Trans. Syst. Man Cybern.*, vol. 24, n.o 4, pp. 656-667, abr. 1994, doi: 10.1109/21.286385.

[5] K. Deb, A. Pratap, S. Agarwal, y T. Meyarivan, «A fast and elitist multiobjective genetic algorithm: NSGA-II», *IEEE Trans. Evol. Comput.*, vol. 6, n.o 2, pp. 182-197, abr. 2002, doi: 10.1109/4235.996017.

[6] E. Dunham, «License for font family 'EB Garamond’ » Font Squirrel». Accedido: 28 de abril de 2024. [En línea]. Disponible en: https://www.fontsquirrel.com/license/eb-garamond