

Proyecto de curso

Moderando el conflicto interno de opiniones en una red social

Análisis y Diseño de Algoritmos II
Escuela de Ingeniería de Sistemas y Computación



Profesor Jesús Alexander Aranda Bueno

Monitor Mauricio Muñoz

Marzo de 2025

1. Introducción

El presente proyecto tiene por objeto verificar que los estudiantes han adquirido el siguiente resultado de aprendizaje: Aplica técnicas de algoritmos voraces y dinámicos, en la construcción de algoritmos, para solucionar problemas de naturaleza combinatoria. Para ello, los estudiantes deben demostrar que logran:

- Aplicar cada estrategia de diseño (fuerza bruta, voraz y programación dinámica) a un ejemplo práctico.
- Usar un enfoque de fuerza bruta para resolver un problema y determinar si la estrategia conlleva a una solución óptima.
- Usar un enfoque voraz para resolver un problema y determinar si la estrategia conlleva a una solución óptima.
- Usar programación dinámica para resolver un problema comprendiendo que dicha estrategia conlleva a una solución óptima.
- Reconocer las ventajas y desventajas de utilizar diferentes estrategias de diseño (fuerza bruta, voraz y programación dinámica).
- Comparar distintas estrategias (fuerza bruta, voraz y programación dinámica) para un problema dado a partir del análisis del respectivo análisis de complejidad y optimalidad.

Para ello el estudiante:

- Desarrolla un programa utilizando un lenguaje de programación adecuado para resolver en grupo un proyecto de programación planteado por el profesor.
- Escribe un informe de proyecto, presentando los aspectos más relevantes del desarrollo realizado, para que un lector pueda evaluar el proyecto.

- Desarrolla una presentación digital, con los aspectos más relevantes del desarrollo realizado, para sustentar el trabajo ante los compañeros y el profesor.

2. El problema de moderar el conflicto interno de opiniones en una red social (ModCI)

2.1. El problema

Las redes sociales les ofrecen a sus usuarios fácil acceso de información y una amplia audiencia. Hay diversos temas sobre los cuales se puede opinar en una red social, en ocasiones, una misma persona puede tener o expresar opiniones completamente opuestas sobre ideas, que son relativamente similares. Este fenómeno se conoce como conflicto interno y es de mucho interés estudiarlo.

Si el conflicto interno es muy alto en una red social puede significar que hay una alta inconsistencia y maleabilidad en la opinión de los participantes, dado que no refleja una postura clara sobre ciertos temas. Por lo tanto, **es interesante desarrollar mecanismos para reducirlo**. Esto se puede lograr invirtiendo recursos que permitan aumentar la conciencia y educar a las personas sobre el desconocimiento profundo sobre un mismo tema, esto se hace con el objetivo de que las opiniones expresadas por una persona sobre temas similares sean más coherentes entre sí. Este es un proceso arduo, complejo y costoso que puede tomar un tiempo significativo en dar resultados.

Este proyecto del curso es una simplificación del anterior problema.

Consideraremos una red social y la opinión con respecto a dos afirmaciones referentes a un tema. Una red está formada por n grupo de agentes, cada uno los cuales tiene una opinión con respecto a dos afirmaciones, dichas opiniones se cuantifican en un valor entero entre -100 y 100 , donde -100 indica una opinión totalmente desfavorable mientras 100 indica una opinión completamente favorable. Una opinión refleja un mayor nivel de moderación, entre más cercana esté a 0 . Y entre más alejada esté la opinión de 0 , esta refleja un mayor nivel de extremismo. Adicionalmente, cada agente tiene un nivel de rigidez o terquedad para cambiar su opinión con respecto a sus afirmaciones, la rigidez o terquedad corresponde a un valor real entre 0 (mínima rigidez) y 1 (máxima rigidez), que modela cuán fácil o difícil es cambiar la opinión del agente, siendo 0 la rigidez de un agente que cambia su opinión con mucha facilidad y 1 la rigidez de un agente que cambia su opinión con mucha dificultad.

El conflicto interno de la red depende del nivel de conflicto interno de cada uno de sus agentes. Lo que se quiere en este proyecto es reducir, lo más que se pueda, el conflicto interno de la red a partir del cambio de opinión de sus agentes.

Ahora, para poder cambiar la opinión de un agente se necesita invertir un esfuerzo, cuyo valor depende de las opiniones del agente sobre las dos afirmaciones y de su nivel de rigidez o terquedad. Toda red cuenta con un valor máximo disponible, que se puede destinar para cambiar la opinión de los agentes de la red, este valor se representa con un número entero positivo.

Con el presente proyecto se pretende abordar, a manera de ejercicio, el problema de determinar sobre cuáles agentes de la red se deben hacer los esfuerzos para el ajuste de opinión, para poder reducir el conflicto interno de la red en la mayor medida posible, respetando el valor máximo disponible para esta tarea. A este problema lo abreviaremos en adelante como *ModCI*

2.2. Formalización

Una red social \mathcal{RS} es una pareja $\langle SAG, R_max \rangle$, donde SAG es una secuencia de grupo de agentes $SAG = \langle sa_0, \dots, sa_{n-1} \rangle$, y $R_max \geq 0 \in \mathbb{N}$ representa el valor entero máximo con que se cuenta para moderar las opiniones de la red \mathcal{RS} .

Un grupo de agentes sa_i es una tupla $\langle n_i^{\mathcal{RS}}, o_{i,1}^{\mathcal{RS}}, o_{i,2}^{\mathcal{RS}}, r_i^{\mathcal{RS}} \rangle$, donde $n_i^{\mathcal{RS}}$ representa el número de agentes que pertenecen al grupo de agentes i , $o_{i,1}^{\mathcal{RS}}$ representa la opinión de cada uno de los agentes del

grupo de agentes i de la red RS sobre la afirmación 1, $o_{i,2}^{\mathcal{RS}}$ representa la opinión de cada uno de los agentes del grupo de agentes i de la red RS sobre la afirmación 2, y $r_i^{\mathcal{RS}}$ representa el nivel de rigidez de cada agente del grupo de agentes i de la red RS para $0 \leq i < n$. Es decir, cada miembro del grupo de agentes i tiene la misma opinión sobre la afirmación 1, sobre la afirmación 2 y el mismo nivel de rigidez.

El valor del conflicto interno de una red RS se puede definir de la siguiente manera:

$$CI(\mathcal{RS}) = \frac{\sum_{i=0}^{n-1} (n_i^{\mathcal{RS}} * (o_{i,1}^{\mathcal{RS}} - o_{i,2}^{\mathcal{RS}})^2)}{\sum_{i=0}^{n-1} n_i^{\mathcal{RS}}}$$

Una estrategia de cambio de opinión de una red \mathcal{RS} de n grupos de agentes, es una secuencia $E = \langle e_0, e_1, \dots, e_{n-1} \rangle$ donde $e_i \in \{0, 1, 2, \dots, n_i^{\mathcal{RS}}\}$; e_i indica el número de agentes del grupo de agentes i a los cuales se le ha cambiado su opinión por medio de E .

Aplicar una estrategia de cambio de opinión E a una red \mathcal{RS} de n grupos de agentes, denotado $ModCI(\mathcal{RS}, E)$ da como resultado una nueva red \mathcal{RS}' con la misma estructura de la red \mathcal{RS} , pero donde la opinión de los agentes cuya opinión ha sido cambiada ahora tienen el mismo valor de opinión para ambas afirmaciones, y la de los otros agentes no ha cambiado. Es decir: si $ModCI(\mathcal{RS}, E) = \mathcal{RS}'$ entonces

$$n_i^{\mathcal{RS}'} = \begin{cases} n_i^{\mathcal{RS}} - e_i & \text{si } e_i > 0 \\ n_i^{\mathcal{RS}} & \text{si } e_i = 0 \end{cases}$$

Se asume en el proyecto, que los agentes a los cuales se le ajustó su opinión por medio de la estrategia E ya no hacen parte de la red resultante \mathcal{RS}' .

El valor del esfuerzo de ajustar las opiniones sobre la red RS con la estrategia E se define de la siguiente forma:

$$Esfuerzo(\mathcal{RS}, E) = \sum_{i=0}^{n-1} [|o_{i,1}^{\mathcal{RS}} - o_{i,2}^{\mathcal{RS}}| * r_i^{\mathcal{RS}} * e_i]$$

Una estrategia de cambio de opinión E sobre la red RS es aplicable si y solo si el esfuerzo de moderar la red según dicha estrategia es menor o igual a R_{max} , el valor máximo con que se cuenta para cambiar las opiniones de la red RS .

El problema de la moderación o cambio de opiniones para minimizar el conflicto interno sobre una red social, $ModCI$, se define entonces así:

Entrada: Una red social $\mathcal{RS} = \langle SAG, R_{max} \rangle$

Salida: Una estrategia de cambio de opinión E aplicable para la red social \mathcal{RS} (es decir tal que $Esfuerzo(\mathcal{RS}, E) \leq R_{max}$), tal que $CI(ModCI(\mathcal{RS}, E))$ sea mínimo.

2.3. ¿Entendimos el problema?

2.3.1. Ejemplo 1

■ Entrada: $\mathcal{RS}_1 = (\ll 3, -100, 50, 0.5 \gg, \langle 1, 100, 80, 0.1 \rangle, \langle 1, -10, 0, 0.5 \gg, 80)$

■ Estrategia de moderación E_1 de RS_1 : $E_1 = \langle 0, 1, 1 \rangle$

Nótese que $Esfuerzo(\mathcal{RS}_1, E_1) = 0 + \lceil 20 * 0.1 * 1 \rceil + \lceil 10 * 0.5 * 1 \rceil = 7 \leq 80$ por lo que la estrategia E_1 es aplicable.

El nivel de conflicto interno de $Mod(\mathcal{RS}_1, E_1) = \mathcal{RS}'_1$ es

$$CI(\mathcal{RS}'_1) = \frac{3 * (150)^2}{3} = 22500$$

- Estrategia de moderación E_2 de RS_1 : $E_2 = \langle 1, 0, 0 \rangle$

Nótese que $Esfuerzo(\mathcal{RS}_1, E_1) = \lceil 150 * 0.5 * 1 \rceil + 0 + 0 = 75 \leq 80$ por lo que la estrategia E_2 es aplicable.

El nivel de conflicto interno de $ModCI(\mathcal{RS}_1, E_2) = \mathcal{RS}_1''$ es

$$CI(\mathcal{RS}_1'') = \frac{2*(150)^2 + 20^2 + 10^2}{4} = 11375$$

- En este caso es mejor solución E_2 que E_1 . ¿Habrán mejores? ¿Habrán estrategias de moderación no aplicables?

2.3.2. Ejemplo 2

- Entrada: $\mathcal{RS}_2 = (\ll 3, -100, 100, 0.8 \gg, \langle 2, 100, 80, 0.5 \rangle, \langle 4, -10, 10, 0.5 \gg, 400)$

- Estrategia de moderación E_1 de RS_1 : $E_1 = \langle 0, 1, 2 \rangle$

Nótese que $Esfuerzo(\mathcal{RS}_2, E_1) = 0 + \lceil 20 * 0.1 * 1 \rceil + \lceil 20 * 0.5 * 2 \rceil = 2 + 20 \leq 400$ por lo que la estrategia E_1 es aplicable.

El nivel de conflicto interno de $Mod(\mathcal{RS}_2, E_1) = \mathcal{RS}_1'$ es

$$CI(\mathcal{RS}_1') = \frac{3*(200)^2 + 1*(20)^2 + 2*(20)^2}{6} = 20200$$

- Estrategia de moderación E_2 de RS_2 : $E_2 = \langle 2, 2, 2 \rangle$

Nótese que $Esfuerzo(\mathcal{RS}_2, E_2) = \lceil 200 * 0.8 * 2 \rceil + \lceil 20 * 0.5 * 2 \rceil + \lceil 20 * 0.5 * 2 \rceil = 360 \leq 400$ por lo que la estrategia E_2 es aplicable.

El nivel de conflicto interno de $ModCI(\mathcal{RS}_2, E_2) = \mathcal{RS}_1''$ es

$$CI(\mathcal{RS}_1'') = \frac{1*(200)^2 + 0 + 2*(20)^2}{3} = 13600$$

- En este caso es mejor solución E_2 que E_1 . ¿Habrán mejores? ¿Habrán estrategias de moderación no aplicables?

3. El proyecto

Su proyecto consiste en explorar tres alternativas para solucionar el problema, una de fuerza bruta, una voraz y una usando programación dinámica, programarlas, analizarlas y hacer un informe al respecto.

3.1. Usando fuerza bruta

Considere el algoritmo que genera todas las soluciones posibles y entre ellas escoge la mejor.

3.1.1. Complejidad

¿Cuál es el orden de complejidad del algoritmo? Argumente su respuesta.

3.1.2. Corrección

¿El algoritmo siempre da la respuesta correcta al problema? Argumente su respuesta.

3.2. Usando un algoritmo voraz

3.2.1. Describiendo el algoritmo

Describa un algoritmo voraz para abordar el problema (ustedes como grupo deciden su estrategia voraz; la idea es que sea la más sencilla posible que ojalá les permita encontrar el óptimo siempre o la mayoría de las veces). **Se dará el puntaje máximo por este criterio a los mejores algoritmos voraces en términos de relación calidad de la solución / tiempo usado para encontrarla.**

3.2.2. Entendimos el algoritmo

Calcule la salida de su algoritmo para las entradas presentadas en 2.3.1 y 2.3.2 y calcule el costo de la solución. ¿Es la solución óptima?

Describa al menos 4 entradas más, calcule la solución entregada por el algoritmo y verifique si es o no la óptima. Guarde los resultados en una tabla.

3.2.3. Complejidad

¿Cuál es el orden de complejidad del algoritmo? Argumente su respuesta.

3.2.4. Corrección

¿El algoritmo siempre da la respuesta correcta al problema? Argumente su respuesta. Si la respuesta es negativa, adicionalmente analice cuándo si y cuándo no da la respuesta correcta.

3.3. Usando programación dinámica

Una alternativa a las dos anteriores sería utilizar programación dinámica. Para hacerlo es necesario comprobar que el problema tiene la propiedad de subestructura óptima y dejarse guiar por ella.

3.3.1. Caracterizando la estructura de una solución óptima

Caracterice la estructura de una solución óptima y a partir de ella describa el conjunto de subproblemas para los cuáles será necesario calcular las soluciones óptimas. ¿Cuántos subproblemas hay?

3.3.2. Definiendo recursivamente el valor de una solución óptima

Defina recursivamente el costo de una solución óptima para cada subproblema definido en el punto anterior.

3.3.3. Describiendo el algoritmo para calcular el costo de una solución óptima

Describa el algoritmo que calcula el costo de una solución óptima al problema original, basado en el punto anterior.

3.3.4. Describiendo el algoritmo para calcular una solución óptima

Tome el algoritmo descrito en el punto anterior y adicione:

- Una estructura que permita almacenar datos para construir una solución óptima cuyo valor sea el calculado por el algoritmo.
- Un algoritmo que recorra esa estructura construyendo una solución de costo óptimo.

3.3.5. Complejidad

Complejidad en tiempo. Calcule la complejidad en tiempo del algoritmo en términos de n .

Complejidad en espacio. Calcule la complejidad en espacio del algoritmo en términos de n .

¿Es útil en la práctica? Suponga que tiene un equipo que procesa 3×10^8 operaciones por minuto y que utiliza 4 bytes de almacenamiento por celda.

Si $n = 2^k$:

- Argumente para qué valores de k el tiempo que se tomará el equipo en resolver el problema es aceptable o inaceptable en la práctica.
- Argumente para qué valores de k el espacio que se tomará el equipo en resolver el problema es aceptable o inaceptable en la práctica.

[Sugerencia:] Utilice las siguientes aproximaciones:

$$n + 1 \sim n, \quad A + 1 \sim A$$

$$1 \text{ minuto} = \frac{1}{3^{42} 5^{52}} \text{ años}$$

$$1 \text{ año} = 3^{42} 5^{52} \text{ minutos} = 518400 \text{ minutos}$$

$$2^{10} \text{ Bytes} = 1 \text{ KB}$$

$$2^{10} \text{ KB} = 1 \text{ MB}$$

$$2^{10} \text{ MB} = 1 \text{ GB}$$

$$10^3 \text{ MB} \sim 1 \text{ GB}$$

3.4. Implementación

Cada grupo debe entregar el código correspondiente a:

- En el ambiente de programación de su preferencia, un programa que contenga una función por cada una de las soluciones diseñadas y analizadas (fuerza bruta, voraz y programación dinámica). Cada función recibe una red social \mathcal{RS} y devuelve una tripleta $(E, Esfuerzo(\mathcal{RS}, E), CI(ModCI(\mathcal{RS}, E)))$ que contiene E , la respuesta al problema, y el valor del esfuerzo de esa solución y del conflicto interno de la red moderada. Esas funciones en su código deben llamarse `modciFB`, `modciV` y `modciPD`.
- Usando la tecnología de su preferencia, una interfaz que permita leer entradas al problema desde un archivo de texto en un formato especificado, y escribir las salidas al problema en un archivo de texto en un formato especificado, así como visualizar las entradas, después de leerlas, y las salidas después de ser calculadas por cualquiera de las funciones solicitadas. Las salidas deben poder ser revisadas por medio de la interfaz en cuanto a su estructura y calidad (optimalidad). En particular debe ser fácil comprobar la corrección del costo de la solución dada.

Todos los programas deben correr bajo un mismo ambiente; se debe entregar el programa **ejecutable** y el código fuente.

3.4.1. Formato de las entradas

Las entradas vendrán en un archivo de texto con $n + 1$ líneas así:

```
n
n_0, op0_0, op0_1, rig0
n_1, op1_0, op1_1, rig1
```

```

.
.
.
n_(n-1), op(n-1)_0, op_(n-1)_1, rig(n-1)
R_max

```

Es decir, la primera línea trae el número de grupos de agentes de la red social (un entero n); las siguientes n líneas traen el número de agentes, la opinión sobre la afirmación 1, la opinión sobre la afirmación 2 y el nivel de rigidez de cada uno de los n grupos de agentes, y la última línea trae el valor máximo disponible para cambiar opiniones en dicha red.

3.4.2. Formato de las salidas

Las salidas se deben producir en un archivo de texto con $n + 1$ líneas así:

```

CI
Esf
mod0
mod1
mod2
.
.
.
mod(n-1)

```

Es decir, la primera línea trae el conflicto interno de la red una vez moderada usando la estrategia escogida, la segunda línea tiene el esfuerzo para llevar a cabo dicha estrategia; luego vienen n líneas, indicando el número de agentes efectivamente que fueron moderados de cada grupo de agentes, es decir indicando la estrategia que lleva a la solución óptima.

3.5. Entrega

La entrega se debe realizar vía el campus virtual en las fechas previstas para ello, por uno sólo de los integrantes del grupo.

La fecha de entrega límite es el 12 de abril de 2025 las 23:59.

Debe entregar un informe en formato pdf, los archivos fuente, un archivo Readme.txt que describa todos los archivos entregados y las instrucciones para ejecutar la aplicación. Todo lo anterior en un solo archivo empaquetado cuyo nombre contiene los apellidos de los autores y cuya extensión corresponde al modo de compresión. Por ejemplo ArandaDiazMuñoz.zip o ArandaDiazMuñoz.rar, o ArandaDiazMuñoz.tgz o ...

4. Sustentación y calificación

El trabajo debe ser sustentado por los autores en el día y hora especificado para su grupo. **La calificación del proyecto para el grupo** se hará teniendo en cuenta los siguientes criterios:

1. Informe (1/2) Debe responder a cada una de las solicitudes formuladas en el enunciado y debe contener al menos una sección dedicada a cada tipo de algoritmo implementado (fuerza bruta, voraz, programación dinámica), además de una sección dedicada a comparar los resultados de las tres soluciones implementadas (para esto use tablas y diseñe casos de prueba y documente cómo los diseñó) y una sección dedicada a concluir sobre las ventajas y desventajas de usar en la práctica los diferentes enfoques.

2. Implementación (1/2). Esto quiere decir que el código entregado funciona por lo menos para las diferentes pruebas que tendremos para evaluarlo y corresponde con todo lo solicitado (incluyendo la interfaz de lectura y visualización solicitada, y la generación de casos de prueba)

En todos los casos la sustentación será pilar fundamental de la nota individual asignada a cada integrante del grupo. En la sustentación se demuestra la capacidad del grupo de navegar en el código y realizar cambios rápidamente en él, así como la capacidad de responder con solvencia a las preguntas que se le realicen.

Cada persona de cada grupo, después de la sustentación, tendrá asignado un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de su sustentación. **Su nota definitiva será la calificación del proyecto para el grupo, multiplicada por ese valor.** Si su asignación es 1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0.

La idea es que lo que no sea debidamente sustentado no vale así funcione muy bien!!! Y que, del trabajo en grupo, es importante que todos aprendan, no sólo algunos.

Éxitos!!!