Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería de Transporte y Logística

Abril, 2020.

TAREA 1

La pandemia

ICT-2233 Flujo en Redes

Prof. Mathias Klapp
Ayudante: Pablo Seisdedos (pcseisdedos@uc.cl)

En esta tarea su grupo estudiará la propagación de un virus mediante herramientas de Flujo en Redes combinadas con simulación computacional.

Esta tarea se estructura en dos partes descritas a continuación.

Parte I: Modelo de contagio

Considere un grupo de n personas a ser estudiados durante 90 días. Se sabe que al comienzo del horizonte ($t = 0$) existe un paciente cero $p_0 \leq n$ que está recién infectado.

Además, suponga que usted posee registro de todas las reuniones desarrolladas entre grupos de personas codificadas como elementos $(P, t) \in R$ de un conjunto R, donde cada uno registra una reunión en el día t de un grupo $P \subset \{1, \dots, n\}$ de dos o más personas. Por ejemplo, el siguiente conjunto

$$R = \{(\{1,2,4\}, 1), (\{1,3,6\}, 3), (\{5,6,7,8\}, 2)\},$$

indica que las personas $1,2$ y 4 se reunieron en el día $t = 1$, que $1,3$ y 6 se reunieron en el día $t = 3$, y que $5,6,7$ y 8 se juntaron en $t = 2$. Considere que cada reunión dura un día y que una persona no participa en más de una reunión al día.

El modelo de contagio del virus es el siguiente:

1. Si una persona sana $p \leq n$ se reúne el día t con al menos una persona enferma, entonces tiene una probabilidad independiente $q_p \in (0,1)$ de contagiarse en t y ser un vector de contagio desde el día $t + 1$.
2. Desde que la persona p está infectada, demora un tiempo aleatorio de r_p días en sanarse y quedar inmune. Es decir, si es contagiada en el día t estará infectada hasta el día $t + r_p$ (inclusive) quedando inmune desde el día $t + r_p + 1$.
3. Considere, además, que durante la enfermedad pasa un tiempo sin que la persona sepa que está infectada y, por lo tanto, sigue participando de reuniones hasta saberlo. Suponga que si se contagió en el día t, entonces descubre que está infectada en $t + c_p \leq t + r_p$. Por lo que suspende todas sus reuniones desde $t + c_p + 1$.

El grupo debe realizar el siguiente análisis:

i) Dado un paciente cero en $t = 0$, el grupo debe determinar a qué subconjunto de personas en $\{1, \dots, n\} \setminus \{p_0\}$ podría eventualmente contagiar en Δt días. Para esta pregunta, considere que $r_p = c_p = \Delta t$ para toda persona $p \leq n$.

Para ello el grupo debe diseñar un grafo que modele la evolución temporal del contagio. En el reporte el grupo deberá explicar su grafo y comentar qué significan los conjuntos de nodos y arcos. Luego, deben implementar este grafo en Python y justificar la estructura de datos que ocuparon para almacenar la red computacionalmente. Por ejemplo, determinar uso de memoria y complejidad asociada a funciones de búsqueda en la red. Para esto deben crear la función **crear_grafo(personas, reuniones)** que reciba los nombres de la base de datos a utilizar para crear el grafo (por ejemplo "personas_5000.txt" "y reuniones_5000.txt"), y retorne el grafo creado. Luego, deben construir una función llamada **determinar_contagiados(grafo, p_0, Δt)** que reciba como argumento el grafo creado, el índice del paciente cero y el número de días transcurridos, y retorne un conjunto de personas posiblemente contagiadas.

ii) En este punto, el grupo debe simular un escenario de contagio desde el paciente cero.

Para ello debe simular contagios ocurridos en cada reunión utilizando los datos de probabilidad de contagio q_p disponibles en **personas_15000.txt** y **personas_5000.txt** y los conjuntos de reuniones disponibles en **reuniones_15000.txt** y **reuniones_5000.txt**. También, suponga que la cantidad de días r_p que la persona p se mantiene infectada después de contagiarse es independiente e idénticamente distribuida (*iid*) con distribución $\mathrm{r_i} \sim \mathrm{Poisson}(15)$. Además, cada persona se demora $c_p = \min\{r_i, \mathrm{x_i}\}$ días en darse cuenta del contagio, donde $\mathrm{x_i} \sim \mathrm{Binomial\ Negativa}(6, 0.5)$.

NOTA: Un método para generar un escenario de una variable aleatoria X on densidad acumulada $F(x)$ consiste en generar un valor aleatorio $U \sim \mathrm{Uniforme}(0,1)$ y luego generar el valor deseado aplicando la función de densidad inversa $X \sim F^{-1}(U)$. Por ejemplo, para generar una variable aleatoria $B \sim \mathrm{Bernoulli}(a)$ se pude simular un número aleatorio U y luego asignar $B = 1$ si $U < a$ y asignar $B = 0$ en otro caso.

Para esto disponen de la función **simulation_parameters(total,s)** que recibe la cantidad de personas y la semilla de simulación y retorna (r_p, c_p, U) que corresponden a diccionarios indexados por el id de las personas (de 0 a total) que contienen que la cantidad de días r_p que la persona p se mantiene infectada después de contagiarse, la cantidad de días c_p en darse cuenta del contagio, y un número generado con una distribución uniforme respectivamente.

ii.a) El grupo debe construir una función llamada **simular_contagio(grafo,p_0, Δt, s)** que reciba como argumento el grafo a utilizar, el índice del paciente cero, el número de días transcurridos y el escenario simulado s (utilice a s como la semilla de simulación). Debe

retornar dos conjuntos: el primero es un conjunto de personas infectadas y el segundo un conjunto de personas recuperadas en el día Δt, además, debe retornar la información necesaria para construir el gráfico pedido a continuación.

ii.b) También, se solicita que el grupo grafique curvas cumulativas con la cantidad de personas infectadas y recuperadas en cada día, además, en el mismo gráfico debe agregar la cantidad de infectados actuales. Para esto deben crear la función **graficar_infectados(grafico)**, en donde "grafico" corresponde al tercer elemento que retorna la función del apartado ii.a. Esta función debe imprimir en pantalla el gráfico solicitado y además debe guardarlo como png con el nombre "grafico.png" en la misma carpeta en donde se encuentra el archivo de código.

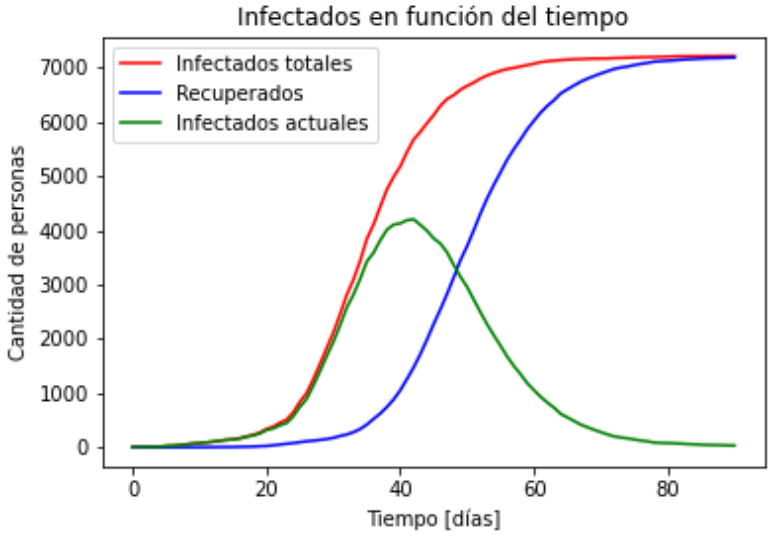


Gráfico 1: Ejemplo gráfico solicitado.
Fuente: Elaboración propia

Se espera que el grupo tome un escenario simulado y analice cualitativamente lo que ve en el grafo.

iii) Ahora, cada grupo debe estimar la probabilidad h_i de haberse contagiado cada individuo $i \in P$ en la red después de Δt días dado que el paciente cero es p_0. Para estimar h_i el grupo debe simular 1,000 escenarios independientes utilizando la función del punto anterior con un grafo de las bases de datos **personas_1000.txt** y **reuniones_1000.txt**, y calcular la fracción entre el número de escenarios en que se contagia la persona i sobre 1,000. Para esto, deberán construir una función computacional llamada **probabilidad_contagio(grafo,p_0,Δt)** que reciba como argumento el grafo ocupado, el índice del paciente cero y el número de días transcurridos y que retorne un diccionario que tenga como llaves a todos los individuos y como valor su probabilidad estimada de estar contagiado.

Parte II: Distribución de alimentos y productos básicos.

Suponga que en una ciudad se ha decretado cuarentena. Las autoridades locales han habilitado un conjunto E ubicaciones para abastecer a la ciudad de productos básicos donde la gente cercana a esos puntos puede retirar productos de forma segura. También existen A puntos de abastecimiento desde donde se distribuyen los productos de primera necesidad. Cada punto de abastecimiento $a \in A$ ofrece al sistema $b_{a,t}$ unidades de producto en el día t, mientras que cada ubicación de retiro $e \in E$ requiere $-b_{e,t}$ unidades en el día t. Asuma que para todo día t se cumple que:

$$\sum_{k=1}^{t} \sum_{e \in E} b_{e,k} \geq 0$$

También, asuma que costos de desplazamiento $c_{a,e}$ y las capacidades $u_{a,e}$ entre todos los puntos de abastecimiento y de encuentro existentes en la red. Además, considere que se puede almacenar hasta I_a unidades de inventario en cada punto de abastecimiento.

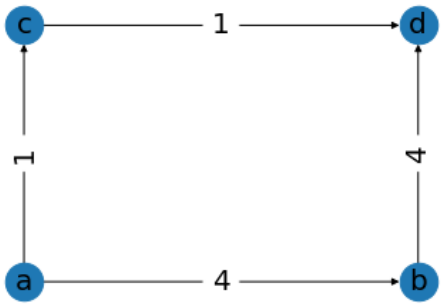
A su grupo se le solicita:

I) Modelar el problema de forma matemática como un PFCM indicando qué representa la función objetivo, parámetros, nodos, arcos, costos de la red, variables, restricciones, etc.

II) Representar gráficamente la red. Para esto debe ocupar la librería *Networkx* mostrando claramente los nodos y arcos de la red, además, se debe diferenciar con distintos colores los puntos de abastecimiento con los de encuentro. Para esto debe crear una función llamada **grafico_pfmc()** que guarde la representación gráfica de la red en formato pdf, notar que esta función no recibe parámetros puesto que lo que se busca es mostrar un esquema de la red, no graficar una red real.

III) Resolver el problema para los datos entregados en la carpeta Instancias parte II. Para esto debe crear una función llamada **resolver_pfmc(abastecimiento, encuentro, transporte, inventario)** que reciba los nombres de las bases de datos a utilizar. Luego, cada grupo deberá diseñar una forma de mostrar la solución obtenida, preferentemente guardando los resultados en un archivo Excel y explicando con palabras el formato en que este se encuentra, además, debe retornar los flujos óptimos en un diccionario cuyas llaves son los nodos de origen, los valores son un diccionario cuyas llaves son los nodos de destino y el valor es el flujo.

Por ejemplo, la siguiente solución:



Se representa de la forma:

{'a': {'b': 4, 'c': 1}, 'd': {}, 'b': {'d': 4}, 'c': {'d': 1}}

Entregables

Este trabajo contempla la elaboración de un reporte en pdf (hecho en Word o LATEX) y la entrega del código en Python (.py o .ipynb). También debe incluir un README en formato md o txt en donde se expliquen las distintas funcionalidades del código y se explicite su funcionamiento.

Especificaciones para el informe

El reporte debe ser ejecutivo y conciso no explicando más de lo que se pregunta. La extensión máxima de cada informe es de 12 páginas. No es necesario incluir una página de portada, ni resumen, ni índice. El anexo puede tener una extensión máxima de 5 páginas y toda figura, tabla y gráfico debe tener nombre, estar rotulado y ser autoexplicativa. Todos los gráficos deben tener título, nombre de ejes y leyenda. La letra debe ser Calibri o similar tamaño 11, a interlineado simple y justificado. El informe debe abordar, como mínimo, los siguientes puntos:

- Breve introducción: descripción general del contenido del informe y breve explicación de lo logrado en cada parte de la tarea.
- Desarrollo de cada parte:
    - Descripción técnica de la solución y metodología. ¿Qué algoritmos utilizó y por qué?
    - Estructura general de código implementado. Basta con un pseudocódigo (para ejemplos de pseudocódigos, revisar AMO), un diagrama de flujo, u otra forma de clarificar lo programado.
    - Análisis cualitativo de la solución.

Especificaciones para el código

Toda herramienta deberá ser programada en Python. Los códigos deben estar adecuadamente comentados e indentados. Cada programa será verificado con herramientas de prevención de plagio. Se permitirá utilizar código encontrado en Internet, pero debe citarse adecuadamente la fuente en el reporte. Los comentarios del código deben ser propios y debe explicar de forma resumida lo que hace el código citado y por qué es útil en la tarea.

No se permite el uso de librerías externas ni de *solvers* de optimización (Cplex, Gurobi, etc.) sin previa consulta al ayudante.

El uso de *Networkx* se limita a solo graficar y resolver un PFCM. Cualquier sección del código que utilice librerías prohibidas no será considerada en la corrección.

Si una función no cumple con el formato explicitado en el informe el resultado será considerado incorrecto. La corrección de los resultados será realizada de forma automática con instancias diferentes a las entregadas pero que cumplen con el mismo formato.

Otros aspectos generales

El trabajo se debe realizar en grupos de máximo tres alumnos organizado por los mismos estudiantes en CANVAS.

Se contemplará realizar una evaluación de pares después de la entrega en donde podría subir o bajar la nota de cada integrante del grupo.

Las entregas se pueden enviar hasta el martes 19 de mayo a las 23:59 vía CANVAS (los retrasos serán fuertemente penalizados). Se debe entregar un archivo `XX.rar` o `XX.zip` (donde `XX` es el número de grupo) que contenga los productos a entregar mencionados anteriormente. No envíe ejecutables.

Cualquier duda sobre el trabajo, hacerla preferentemente a través del foro del curso en CANVAS o a pcseisdedos@uc.cl .