Trabajo Práctico: Algoritmos en Grafos

Investigación Operativa - Segundo Cuatrimestre 2015

1 Introducción

En el presente Trabajo Práctico se propone implementar algoritmos eficientes para la resolución de dos variantes sobre el problema del flujo máximo y el algoritmo que se deriva del teorema de Vizing para el coloreo de aristas en grafos simples.

2 Contexto

- Un problema al que se enfrenta el sistema educativo cada curso es el de asingar docentes nuevos a plazas que necesitan cubrirse en los diferentes colegios del país. Aún centrándonos tan solo en la provincia de Buenos Aires, los recursos humanos que se deben gestionar son demasiados para que la tarea se realice 'a mano'.

  Cada centro $i \in \{A, B, C, ...\}$ debe cubrir unas necesidades lectivas (necesita un maestro por grupo, para un total de C_i grupos). Cada maestro $j \in \{1, 2, 3, ...\}$ quiere laburar una cantidad de horas semanales determinada, lo que se traduce en que quiere ser asignado a una cantidad M_j de grupos. Esta asignación debe cumplir con la restricción de que un maestro sólo puede ser asignado a un grupo para impartir una materia para la que está capacidtado (oficialmente). Además, cada profesor tiene una preferencia w_{ij} por determinados centros. Se desearía cubrir completamente la demanda de los centros, maximizando la satisfacción global de los profesores con respecto a sus preferencias de centro. La siguiente figura ilustra la situación:

  Cómo se ve, el problema se puede modelar como un problema emperajemiento en grafos bipartitos, que puede resolverse adaptando los pesos, buscando un Flujo Máximo de Costo Mínimo.

- Un problema de cariz completamente diferente en apariencia es el de la transferencia de archivos en una red de computadoras. Entre cada par de computadoras hay una serie de archivos que quieren intercambiar, pero Cada computadara tiene un número limitado de puertos de comunicación, por lo que todas las tranferencias no pueden realizarse simultaneamente. El objetivo es decidir como organizar las tranfericias para que sea necesario el m'inimo número de tandas posibles para completarlas todas. Así expuesto, este problema puede modelarse como un problema de coloreo m'inimo de aristas en un multigrafo, permitiendo que en cada vértice se repitan p colores, que puede resolverse con $(\Delta G + m)/p$ colores (donde m es la m'axima múltiplicidad del multigrafo) Si suponemos que entre cada par de ordenadores solo hay un archivo a transferir y que cada ordenador tiene un único puerto de comunicaciones, el problema se reduce a uno de coloreo m'inimo de aristas en grafos simples.

  Cabe tener en cuenta que este problema puede plantearse así partiendo de que ya existe una asignaci'on de archivos por transferir a pares de ordenadores. Si esta asignación no existe pero se conoce que archivos están disponibles en, y que archivos requiere, cada computadora, podemos obtenerla resolviendo un problema de matching en un (multi)grafo dirigido, que puede resolverse mediante un problema de flujo.

3 Algoritmos

3.1 Flujo Máximo de Costo Mínimo (FMCM)

Se tiene un grafo dirigido $G(V,E)$, donde cada arista $(u,v) \in E$ tiene asociada una capacidad $C(u,v) > 0$ y un coste $W(u,v) > 0$. El grafo es además "unidireccional", es decir $C(u,v) > 0$ implica $C(v,u) = 0$.

Una modificación del algoritmo de Ford-Fulkerson visto en clase que resuelve el problema de encontrar el FMCM está dado por el siguiente pseudocódigo:

(1) Inicializamos las capacidades $C_f = C$ y los costos $W_f = W$.

(2) Construir un sub-grafo $G_f(V,E)$ donde la arista $(u,v) \in V$ sii: $C_f(u,v) > 0$.

(3) Buscar (mientras exista) un camino $p \in G_f$ de s a t, que minimize el costo W_f:

(4) Calcular c_{min}, la mínima capacidad a lo largo del camino p.

(5) Actualizar para cada $(u,v) \in p$

- $C_f(u,v) = C_f(u,v) - c_{min}$ *(mandamos flujo)*
- $C_f(v,u) = C_f(v,u) + c_{min}$ *(permitimos deshacer el camino)*
- $W_f(v,u) = -W(u,v)$

(6) El algorimo se detiene si el camino del paso (3) no existe.

3.2 Coloreo: Vizing

Dado un grafo simple y simétrico $G(V,E)$ sus aristas se pueden colorear con $\xi' \leq \Delta G + 1$ colores usando el algoritmo de vizing. El siguiente pseudocódigo muestra una adaptación del algoritmo original:

(1) SET AristasPintadas $\leftarrow$ 0

(2) ELEGIR e1 EN ListaAristasNoPintadas

(3) MIENTRAS AristasPintadas $< |E|$

  (i) SET reasigna $\leftarrow$ FALSE

  (ii) SET $v0 \leftarrow$ Extremo1(e1)

  (iii) SET $u \leftarrow$ Extremo2(e1)

  (iv) SET $c \leftarrow$ ColorComunSinUsar($v0$,u).

  (v) SI EsValido(c)
   - SET $e1.color \leftarrow c$
   - ELIMINA $e1$ DE ListaAristasNoPintadas

  (vi) SINO
   - SET $c1 \leftarrow$ ColorSinUsar($v0$, $ListaColores$)
   - SET $c2 \leftarrow$ ColorSinUsar(u, $ListaColores$)
   - SET $P \leftarrow$ CaminoAlternadoMaximo($c1$,$c2$,$v0$)
   - SI $u \neq$ Fin(P)
     - IntercambiaColores(P,c1,c2)
     - SET $e1.color \leftarrow c2$
     - ELIMINA $e1$ DE ListaAristasNoPintadas

- SINO
  - SET $e2 \leftarrow$ UltimaArista(P)
  - BorraColor(e2,c1)
  - SET $e1.color \leftarrow c1$
  - ELIMINA $e1$ DE ListaAristasNoPintadas
  - SET reasigna $\leftarrow$ TRUE

(vii) SI reasigna = TRUE

- SET $e1 \leftarrow e2$
- AÑADE $c1$ A $e1$.ColoresProhibidos

(viii) SINO

- SET AristasPintadas $\leftarrow$ AristasPintadas +1
- ELEGIR e1 EN ListaAristasNoPintadas

4 Tarea a realizar

Para que el TP se considere aprobado debe completarse AL MENOS UNO de los siguientes puntos:

- Implementar el algoritmo FMCM descrito en el punto 3.1 y escribir una rutina que genere un grafo aleatorio de entrada para el problema de asiganción de docentes, de tamaño arbitrario.

  Utilizando la librería "time" de Python, calcule cuánto tarda su algorimo en función del tamañao del problema y grafique. ¿Es polinomial? ¿Cuál es el tamaño máximo de un problema que puede resolver con su implementación en un tiempo de cómputo de 1 segundo, 10 segundos y 1 minuto?

- Implementar el algoritmo de coloración de aristas descrito en el punto 3.2 y escribir una rutino que genere un grafo aleatorio SIMPLE de tamaño arbirario para ser coloreado.

- Modificar el algoritmo FMCM para hallar un flujo máximo de mínimo costo en MULTIGRAFOS y usarlo para resolver el problema de matching asociado a la transferencia de archivos en una red de computadoras descrito en el apartaod 2. Usar el resultado para generar un grafo SIMPLE de modo que el coloreo de sus aristas resuelva el problema del 'scheduling' de las transferencias.

- Modificar el algoritmo de coloreo para que resuelve el problema de coloreo mínimo de aristas en un multigrafo y repetir el item anterior sin imponer que el resultado del matching sea un grafo simple.

5 Que debe entregarse

Deberán entregar un archivo `.zip` que contenga:

- Los archivos .py con el código que corresponda al punto "Tarea a realizar".

- Un archivo `.pdf` que contenga un pequeño informe del TP. En particular debería contener las secciones **Manual de Usuario** e **Implementación** que describan, respectivamente, como usar el código que nos evian y una explicación de porque se han decidido por la estructura y algoritmos particulares que han usado.