Algoritmia Avanzada: Teoría de Grafos

Dr. Arno Formella

26 de enero de 2006

Universidad de Vigo
Departamento de Informática
Área de Lenguajes y Sistemas Informáticos
E-32004 Ourense

`http://www.ei.uvigo.es/˜formella`
`formella@ei.uvigo.es`

1. Curso

La página incial del curso es:

`http://www.ei.uvigo.es/~formella/doc/tc05`

Estos apuntes se acompañan con ilustraciones en pizarra dónde se explica las notaciones y el funcionamiento de los algoritmos.

El texto es meramente una brevísima introducción (5 horas) a la teoría de grafos donde se pincela ciertos aspectos más bien para motivar y despertar interés por este campo fascinante de la Teoría de Grafos.

caminos más cortos: busca o desarrolla un algoritmo que enumera todos los caminos más cortos que una variable de entrada k que existen entre dos vértices en un grafo plano.

camino de peso mínimo: busca o desarrolla un algoritmo que calcule el camino de peso mínimo entre dos vértices en un grafo ponderado, si se permite ciclos de pesos negativos, pero se pone como restricción adicional que ninguna arista se recorre más de una vez (pero si se puede visitar un vértice más de una vez).

3. Motivación

Grafos de forma intuitiva se encuentran por ejemplo en las siguientes situaciones:

- planos o mapas de carreteras o calles
- redes de flujos (datos, líquidos)
- redes de transportes urbanos
- conexiones químicas entre átomos de una molécula
- relaciones de vecinidad en un mapamundi
- relaciones de interferencia entre antenas en un sistema de comunicación inalámbrica
- los enlaces entre documentos en el Internet

Es decir, un grafo es el *concepto abstracto* detrás de la representación de relaciones (aristas) entre entidades (vértices).

Problemas que se quiere resolver:

- Dado un traje completo para vestirse con el fin de ir a una boda en Galicia en invierno, es decir, llueve, ¿En qué orden hay que ponerse las cosas?

- Dado una fiesta con igual número de chicas y chicos, ¿Cuáles son las condiciones para que sea posible organizar un baile de todos (en parejas chica–chico)?
- Dado una descripción de las interconexiones de un circuito electrónico, ¿Cómo posicionar los chips con sus interconexiones en una placa?
- Dado un conjunto de casas por proteger de fuego, ¿Dónde posicionar los bomberos disponibles para minimizar su radio de acción?

4. Nociones básicas

Las notaciones que se suelen usar en la teoría de grafos son muy intuitivas y se basan ya casi siempre en la nomen clatura inglesa. Se aprovecha de interpretaciones sencillas y expresivas de los símbolos disponibles desde otras ramas de la matemática.

Notaciones:

V	conjunto de vértices o nodos (o a veces, puntos)
$[V]^r$	conjunto de todos los subconjuntos de V de tamaño r
$E \subseteq [V]^2$	conjunto de aristas
$v \in V$	vértice
$e = \{x, y\} \in E$	arista
$\{x, y\} \Longleftrightarrow: xy$	xy es arista
$G = (V, E)$	grafo
$V(G)$	vértices del grafo G
$E(G)$	aristas del grafo G
$v \in G :\Longleftrightarrow v \in V(G)$	v es vértice del grafo G
$e \in G :\Longleftrightarrow e \in E(G)$	e es arista del grafo G
$\lvert V\rvert =: n$	número de vértices
$\lvert V\rvert = \lvert V(G)\rvert = \lvert G\rvert$	notaciones equivalentes
$\lvert E\rvert =: m$	número de aristas
$\lvert E\rvert = \lvert E(G)\rvert = \lVert G\rVert$	notaciones equivalentes

Notaciones:

$E(X,Y)$	conjunto de $X-Y$-aristas
$E(v) := E(v, V \setminus \{v\})$	conjunto de aristas incidentes a v
$N(v)$	conjunto de vértices adyacentes a v, vecinos
$d(v) := \lvert E(v)\rvert = \lvert N(v)\rvert$	grado del vértice v
$d_G(v)$	grado del vértice $v \in G$
$\delta(G)$	grado mínima de los vértices en G
$\Delta(G)$	grado máxima de los vértices en G
$\epsilon(G) := \lvert E\rvert/\lvert V\rvert$	grado medio de los vértices en G

Notaciones:

$G \setminus e$	grafo $(V, E \setminus \{e\})$
$G \setminus v$	grafo $(V \setminus \{v\}, E \setminus E(v))$
$G \setminus E'$	grafo $(V, E \setminus E')$
$G \setminus V'$	grafo $(V \setminus V', E \setminus E(V'))$

Vocabulario:

vecino	v es vecino de w, si $vw \in E(v)$, es decir, si v y w son adyacentes
vecina	e es vecina de f, si $e \cap f \neq \emptyset$ es decir, e y f son incidentes al mismo vértice
independiente	vértices (o aristas) no adyacentes son independientes; un conjunto de vértices (o aristas) que mutuamente son independientes es un conjunto independiente
completo	un grafo es completo, si todos sus vértices son vecinos
partición	el conjunto de conjuntos $\{V_0, ..., V_{r-1}\}$ es una partición de V, si $V = \bigcup_i V_i$, $V_i \neq \emptyset$, y $\forall i \neq j : V_i \bigcap V_j = \emptyset$

Notaciones:

$\alpha(G)$	cardinalidad del conjunto de vértices independientes más grande del grafo G

Es decir, en la mayoría de los casos se entiende como grafo solamente el caso en el cual existe como mucho una arista (o arista dirigida) entre dos vertices diferentes.

5. Representación

Se puede visualizar un grafo pintando sus vértices como puntos y sus aristas como líneas entre los puntos correspondientes.

¿Cuáles son las operaciones que se quieren realizar con un grafo (y sus componentes)?

Se puede almacenar un grafo con tres métodos básicos:

- una matriz de adyacencia
  - matriz cuatrada, y en el caso simple, binaria (y simétrica si no es digrafo) que codifica si existe una arista entre dos vértices
  - complejidad de memoria $\Omega(n^2)$
- listas de adyacencia
  - lista o array de vértices que contiene en cada entrada una lista de los vértices adyacentes
  - complejidad de memoria $\Theta(n + m)$
- tablas de dispersión
  - lista o array de vértices que contiene en cada entrada una tabla de dispersión de los vértices adyacentes
  - complejidad de memoria $\Theta(n + m)$

¿Cuáles son las principales ventajas y desventajas de cada uno de los métodos?

(es decir, si x e y son vecinos en G, lo son también $\varphi(x)$ y $\varphi(y)$ en G'), entonces G es isomorfo a G', $G \simeq G'$ o también $G = G'$, es decir, se dice *el grafo* G.

Una función f sobre grafos con $f(G) = f(G')$ si $G \simeq G'$ se llama invariante. Por ejemplo, invariantes son:

- n
- m
- ¿hay otras?

No se conoce ninguna invariante sobre grafos que se puede calcular en tiempo polinomial (y determinista) que decida que dos grafos son isomorfos, pero tampoco se ha comprobado hasta hoy que el problema de isomorfismo entre grafos sea NP–completo.

Recordamos que significa NP–completo:

Un problema pertence a la clase de problemas NP–completo, si existe una maquina de Turing no–determinista que resuelve el problema en tiempo polinomial respecto a la longitud de entrada y si todos los demás problemas dentro de la clase como mucho son más fáciles.

Entonces sabemos sobre problemas NP–completos:

- Existe un algoritmo para su solución.
- Si alguien nos da una solución podemos comprobar en tiempo polinomial que de verdad es una solución.
- Si conocemos un algoritmo polinomial para un problema NP–completo sabemos implícitamente algoritmos para todos los problemas de la clase cuyos tiempos de cálculo siguen siendo polinomial.
- Siempre se puede usar búsqueda exhaustiva para resolver el problema de forma determinista (con tiempo de ejecución exponencial).
- La pregunta de existencia de algoritmos polinomiales y deterministas para problemas NP–

7. Grafos parciales y subgrafos

Sean $G = (V, E)$ y $G' = (V', E')$ dos grafos.

Notaciones:

$G \cup G' := (V \cup V', E \cup E')$	unión de grafos
$G \cap G' := (V \cap V', E \cap E')$	intersección de grafos

Vocabulario:

grafo parcial	G' es grafo parcial de G, $V' \subseteq V$ y $E' \subseteq E$
subgrafo	G' es subgrafo de G, si $G' \cap G = G'$, es decir, G' es un grafo parcial de G que contiene todas las aristas de G cuyos vértices incidentes están en V'.
inducido	si $V' \subseteq V$, el grafo $(V', E(V', V'))$ es el subgrafo G' de G inducido por V'

Notaciones:

$G' \subseteq G$	G' es grafo parcial de G
$G[V']$	subgrafo de G sobre V' asumiendo $V' \subseteq V$
$G[G']$	subgrafo de G sobre $V(G')$ asumiendo $G' \subseteq G$

Vocabulario:

r-partido	un grafo es r-partido, si existe una partición de V en r conjuntos independientes
bipartido	2-partido

8. Grafos especiales

longitud	número de aristas de un camino o ciclo
cíclico	un grafo que contiene un ciclo es cíclico
acíclico	un grafo que no contiene ningún ciclo es acíclico
cintura	un ciclo mínimo que un grafo contiene
circumferencia	ciclo máximo que un grafo contiene

Notaciones:

$g(G)$	longitud de la cintura de G, ($g(G) = \infty$ si G acíclico)
$D(G)$	longitud de la circumferencia de G ($D(G) = 0$ si G acíclico)

grafos bipartidos B

grafos bipartidos completos K^{n_1,n_2}

hipercubos Q^k

Propiedades (con excepciones para Q^0 y Q^1):

- $n = |V| = 2^k$
- k-regular
- bipartidos (¿Cómo particionar?)
- $g(Q^k) = 4$
- $D(Q^k) = 2^k$ (¿Cuál es un ciclo máximo?)

9. Conexión

Vocabulario:

componentes conexas	conjunto de subgrafos conexos y maximales

La distancia define una métrica, es decir,

1. $d(v, w) \geq 0$
2. $d(v, w) = 0 \iff v = w$
3. $d(v, w) = d(w, v)$
4. $d(u, w) \leq d(u, v) + d(v, w)$

Vocabulario:

conexo	v y w son conexos, si $d(v, w) < \infty$;
	G es conexo, si todas las parejas $v, w \in V$ son conexas
disconexo	G es disconexo si no es conexo
puente	arista $e \in G$ es una puente, G conexo, tal que $G \setminus e$ es disconexo
vértice de corte	vértice $v \in G$, G conexo, tal que $G \setminus v$ es disconexo

Notaciones:

$c(G)$	número de componentes conexas de G
$\kappa(G)$	cardinalidad mínima de un subconjunto de vértices de G
	tal que $G \setminus V$ sea disconexo
$\lambda(G)$	cardinalidad mínima de un subconjunto de aristas de G
	tal que $G \setminus E$ sea disconexo

Vocabulario:

k-conexo	G es k-conexo, si $\kappa(G) \geq k$
biconexo	2-conexo
k-aristoconexo	G es k-aristoconexo, si $\lambda(G) \geq k$
bloque	subgrafo máximo biconexo

Teorema:

las siguientes propiedades son equivalentes:

- G es un árbol
- entre cada pareja de vértices existe un camino único
- cada arista es un puente
- G es acíclico y $n = m - 1$
- G es conexo y $n = m - 1$
- G es acíclico y maximal en $|E|$
- G es conexo y minimal en $|E|$

Vocabulario:

árbol con raíz	un árbol con un vértice marcado como raíz
árbol libre	un árbol sin marca en ningún vértice
árbol generador	T es árbol generador de un grafo G, si $T \subseteq G$ y $V(T) = V(G)$

Teorema:

cada grafo G contiene un bosque generador, y si G es conexo, contiene un árbol generador (cuya raíz puede ser cualquier vértice)

11. Recorridos

Vocabulario:

12. Teoremas

Teorema (Euler):

$\sum_v d(v) = 2m$

Teorema:

cada grafo tiene un número par de vértices con grado impar

Teorema:

cada grafo G contiene un camino P con $\|P\| \geq \delta(G)$, y cada grafo G con $\delta(G) \geq 2$ contiene un ciclo C con $|C| > \delta(G)$

Teorema:

si G no trivial: $\kappa(G) \leq \lambda(G) \leq \delta(G)$

Teorema:

cada grafo G con $|E| > 1$ contiene un subgrafo H con $\delta(H) > \epsilon(H) \geq \epsilon(G)$

Teorema:

G es bipartido con $|V_0| \neq |V_1| \quad \Longrightarrow \quad G$ no es hamiltoniano

Teorema:

$\forall v, w \in V, vw \notin E : d(v) + d(w) \geq n \quad \Longrightarrow \quad G$ es hamiltoniano

Teorema:

G es hamiltoniano $\quad \Longrightarrow \quad \forall S \subset V : c(G \setminus S) \leq |S|$

Teorema:

decidir si un grafo G es hamiltoniano es NP-completo

¿Algoritmo que calcule un recorrido euleriano?

Teorema:

G k-conexo $\quad\Longrightarrow\quad |E| \geq \lceil kn/2 \rceil$

Teorema (Whitney):

G es 2-conexo ($|G| > 2$) $\quad\Longrightarrow\quad \forall v, w \in V \exists vPw, vQw : P \cap Q = \emptyset$

Es decir, existen dos caminos que no se intersecan entre todas las posibles parejas de vértices en G.

Teorema (Mader):

cada grafo G con $\epsilon(G) \geq 4k$ contiene un grafo k-conexo como grafo parcial

13. Algoritmos básicos

Algoritmo:

Exploración en profundidad (otros dicen recorrido en profundidad, o depth first search, DFS)

Con el algoritmo DFS se obtiene un árbol con raíz T y aristas de retroceso no pertenecientes a T. El algoritmo sirve entre otras cosas:

- determinar componentes conexas
- detectar puentes
- detectar vértices de corte
- detectar bloques
- ordenación topológica

v es vértice de corte $\iff$ no existe una arista de retroceso desde el subárbol debajo de v hacia un antecesor de v en T

DFS tiene complejidad $\Theta(n+m)$ (asumiendo listas de adyacencias, ¿y con los demás?)

Algoritmo:

Exploración en anchura (otros dicen recorrido en anchura, o breadth first search, BFS)

Con el algoritmo de BFS se obtiene un árbol con raíz T, aristas de retroceso no pertenecientes a T, y aristas de cruce. El algoritmo sirve entre otras cosas:

- determinar caminos lo más cortos entre vertices
- ordenación topológica

BFS tiene complejidad $\Theta(n+m)$ (asumiendo listas de adyacencias, ¿y con los demás?)

14. Grafos dirigidos

Sea G un grafo y $\bar{G}$ un grafo dirigido (digrafo).

Vocabulario:

fuertemente conexo	$\bar{G}$ es fuertemente conexo, si existe un recorrido entre cada pareja de vértices de G
orientable	G es orientable, si existe un grafo $\bar{G} \simeq G$ que es fuertemente conexo
componente fuertemente conexa	conjunto máximo de vértices de $\bar{G}$ cuyo subgrafo inducido es fuertemente conexo

Notaciones:

$d(v)$ grado entrante

(¿Qué se entiende bajo una orientación óptima? Depende: minimizar el promedio de las distancias, minimizar el máximo de las distancias, minimizar el máximo de las diferencias entre las distancias en G y en $\bar{G}$ correspondientes.)

Se puede explorar también un digrafo en anchura (BFS) o en profundidad (DFS). A parte de las aristas del árbol y las aristas de retroceso, DFS produce aristas de progreso y aristas de cruce.

DFS se usa para producir una ordenación topológica de un digrafo acíclico, es decir, en el orden aparece un vértice v antes de un vértice w, si existe un camino desde v a w.

DFS se usa para determinar los componentes fuertemente conexos.

¿Algoritmo que calcule los componentes fuertemente conexos?

se puede seguir también recorridos eulerianos:

Teorema:

$\bar{G}$ es euleriano $\iff$ $\bar{G}$ es conexo y $\forall v \in V : d_i(v) = d_o(v)$

¿Algoritmo que calcule un camino euleriano en un digrafo?

15. Grafos ponderados

Notaciones:

(G, W)	grafo ponderado con $W : E(G) \longrightarrow \mathbb{R}^+$
$w(G)$	peso de grafo G, $w(G) = \sum_{e \in G} w(e)$
$w(P)$	peso de camino P, $w(P) = \sum e \in Pw(e)$
$d(v, w)$	distancia entre dos vértices, $d(v, w) = \min_P \{w(vPw)\}$
$dt(v) = \sum_w d(v, w)$	distancia total de un vértice

si $w(e) = 1 \forall e \in E$ se reproduce la distancia de arriba

Notaciones:

	por los vértices con excentricidad mínima
mediana	la mediana del grafo G es el subgrafo de G inducido por los vértices con distancia total mínima

Teorema:

G conexo $\implies$ $\mathrm{rad}(G) \leq \mathrm{diam}(G) \leq 2 \cdot \mathrm{rad}(G)$

Teorema:

Todo grafo es centro de un grafo.

Teorema:

El centro de un árbol consiste en uno o dos vértices.

¿Algoritmo que calcule el centro de un árbol?

16. Árboles generadores y caminos mínimos

dado un grafo G (o un digrafo $\bar{G}$) ponderado

preguntas interesantes:

- ¿Cuál es un bosque generador con peso mínimo en G?
- dado un vertice $s \in \bar{G}$, ¿Cuáles son los caminos mínimos hacia los demás vértices?
- ¿Cuáles con los caminos mínimos entre todas las parejas de vértices?

Hacia cada vértice $u \in G$ existe un camino desde s con distancia mínima.

Las distancias a todos los vértices en uno de estos caminos a su vez son caminos mínimos.

Si G es conexo, se puede construir un árbol con raíz $s \in G$ que determina todos los caminos

Algoritmo de DIJKSTRA, complejidad $O(n^2)$

El algoritmo funciona igual con grafos como con digrafos.

¿Se puede permitir pesos negativos?

pueden existir ciclos negativos, es decir, ciclos con pesos negativos

¿Algoritmo que calcule un árbol mínimo desde un vértice s en G si G contiene pesos negativos?

Algoritmo de BELLMANN–FORD, complejidad $O(mn)$

El Algoritmo detecta la existencia de ciclos negativos en que caso termina sin solución.

¿Algoritmo que calcule los caminos mínimos entre todas las parejas de vértices incluyendo el caso de pesos negativos?

Algoritmo de FLOYD–WARSHALL, complejidad $O(n^3)$

El algoritmo detecta la existencia de ciclos negativos en que caso termina sin solución.

Mejoras:

si se tiene más información sobre los grafos y sus pesos se puede mejorar los algoritmos:

- si los pesos están confinados por una constante W:
- si el número de aristas m está confinado por $O(n \log n)$:

17. Minores y Extremalidad

contracción de arista

minor

minor topológico

18. Planaridad

Vocabulario:

planar	un grafo es planar cuando se puede pintar sus vértices y sus aristas en un plano de tal manera que ninguna pareja de aristas se interseca
región	una región es un área del plano donde se ha pintado un grafo planar que esté confinado por aristas
grafo triangular	un grafo es triangular, si en su representación planar en el plano toda región está confinada por tres aristas del grafo

Teorema (Euler):

G planar y conexo con $n \leq 1$, m y l $\quad \Longrightarrow \quad$ $n - m + l = 2$

Teorema:

G planar con $n \geq 3$ $\quad \Longrightarrow \quad$ $m \leq 3n - 6$

Teorema:

G maximal planar si es un grafo triangular (y contiene $3n - 6$ aristas)

Teorema (Kuratowski):

G es planar $\quad \Longleftrightarrow \quad$ no contiene un K^5 o un $K^{3,3}$ como minor (o minor topológico)

19. Coloración

- Coloración de vértices
- Coloración de aristas
- Coloración de grafos planos

$\chi(G)$	número mínimo de colores que se necesita para colorar los vértices de un grafo
$\chi'(G)$	número mínimo de colores que se necesita para colorar las aristas de un grafo

Teorema (Brooks):

G conexo $G \neq C^{\text{impar}}$ y $G \neq K^n$ $\implies$ $\chi(G) \leq \Delta(G)$

Teorema (Koenig):

G bipartido $\implies$ $\chi'(G) = \Delta(G)$

20. Flujos

1. Flujos y redes
2. Flujos y cortes
3. Flujos y coloración

Vocabulario:

viable	$f(e) \leq w(e)$, es decir, el flujo es como mucho igual a la capacidad
conservante	$\sum_v f_{in}(e) = \sum_v f_{out}(e)$, es decir, tanto como entra sale de un nodo (claro, excepto para fuente y afluente).

Teorema (Ford-Fulkerson):

Máximo flujo es igual a mínimo corte.

Algoritmo de FORD–FULKERSON, complejidad $O(F * m)$ siendo F el máximo flujo (asumiendo enteros como capacidades).

21. Emparejamientos

Vocabulario:

emparejamiento	$M \subset E$ y $\forall e, f \in M : e \cup f = \emptyset$, es decir, pares de aristas no tienen vértices en común
perfecto	M cubre todos los vértices de G
M-alternado	un camino en G alternando con arista de M
M-aumento	camino M-alternado que se puede aumentar

Teorema (Berge):

emparejamiento M es máximo $\Longrightarrow$ G no contiene caminos M-aumento

Teorema (Hall), teorema del matrimonio:

Sea $G = (U \cup V, E)$ un grafo bipartido. G tiene un emparejamiento completo (para U) $\Longrightarrow$ $\forall S \subseteq U : |N(S)| \geq |S|$, siendo $N(S)$ conjunto de vecinos de S.

22. Aplicaciones

1. Planificación
2. Diseño de circuitos
3. Juegos
4. Optimización
5. Teoría de Codificación
6. Visualización

Recorrido óptimo: por ejemplo algoritmo de EDMONDS–JOHNSON (1973)

árbol	tree
árbol generador	spanning tree
arista de cruce	cross edge
arista de progreso	forward edge
arista de retroceso	back edge
arista	edge
articulación	articulation
bosque	forest
camino	path
cintura	girth
conexo	connected
dirigido	directed
emparejamiento	matching
exploración en anchura	breadth first search
exploración en profundidad	depth first search
flujo	flow
fuertemente conexo	strongly connected
grado	degree
peso	weight
ponderado	weighted
puente	bridge
recorrido	walk
vecino	neighbor
vértice de corte	cutvertex

Inglés–Español (que no son triviales):

articulation	articulación
back edge	arista de retroceso
breadth first search	exploración en anchura
bridge	puente
connected	conexo
cross edge	arista de cruce
cutvertex	vértice de corte
degree	grado

neighbor	vecino
path	camino
spanning tree	árbol generador
strongly connected	fuertemente conexo
tree	árbol
walk	recorrido
weighted	ponderado
weight	peso

Bibliografía

Libros:

- Reinhard Diestel. *Graph Theory.* 2nd edition, Springer Verlag, 2000.
  Existe una versión electrónica no imprimible:
  `http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/download.html`
- Thomas H. Cormen, Charles E. Leiseron, Ronald L. Rivest. *Introduction to Algorithms.* McGraw Hill, 1996.

Enlaces: (como disponibles en noviembre 2003)

- `http://www.graphtheory.com`
- `http://www.ericweisstein.com/encyclopedias/books/GraphTheory.html`
- `http://mathworld.wolfram.com/Graph.html`

Apuntes: (como disponibles en noviembre 2003)

- Gregorio Hernández Peñalver, Universidad Politécnica de Madrid,