

# Algoritmi per la visualizzazione

Prof.ssa Tiziana Calamoneri

## GENERALIZZAZIONE DELLA VISUALIZZAZIONE DI GRAFI:

IL CASO PARTICOLARE DEL DISEGNO ORTOGONALE

## Generalizzazione della visualizzazione di grafi (ortog.)

I grafi considerati per il dis. ort. sono:

- di grado al più 4 per il disegno 2D
- di grado al più 6 per il disegno 3D
- 2-connessi per il disegno 2D
- planari, per gli algoritmi basati sulla rappresentazione di visibilità.

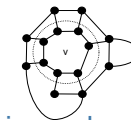
Vogliamo estendere gli algoritmi studiati a grafi che non possiedono queste proprietà...

## Estensione al caso di grado alto

## Dis. 2D ort. (rappr. di visibilità)

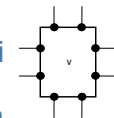
- [Fößmeier, Kaufmann '96]

Dato  $G$  tramite una sua rappr. piana, per ogni  $v$  di grado maggiore di 4:



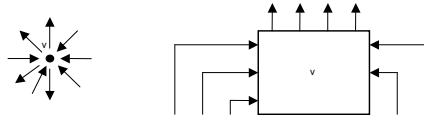
ciclo espansione di  $v$

- \* Costruire una rappr. piana in modo che i cicli espansione non abbiano svolte.
- \* Si può dim. che la rappr. di visibilità li rappresenta come rettangoli
- \* Alla fine si sostituisce ciascun ciclo con un nodo rettangolare. (TESINA)



## Dis. 2D ort. (st-numerazione) (1)

- Nodi=rettangoli
- Sul bordo, i punti a coord. intere sono detti *connettori*.
- Ogni connettore può avere un solo arco, tranne quelli ad angolo, che ne hanno 2.
- I rettangoli devono essere sufficientemente grandi da consentire l'attacco di tutti gli archi incidenti al nodo.



Algoritmi per la Visualizzazione  
Prof.ssa Tiziana Calamoneri

5

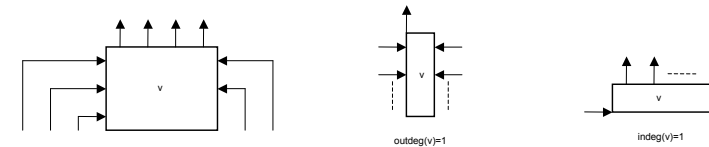
## Dis. 2D ort. (st-numerazione) (2)

**Algoritmo:**

**Input:**  $G$  2-connesso

**Output:** rappr. ort. di  $G$

- calcola una st-numerazione, inducendo un verso agli archi
- gli archi uscenti sono attaccati sulla parte sup. del rettangolo, gli archi entranti divisi ai lati
- grado entrante o uscente 1 o 0 (sorgente e pozzo) deve comunque garantire la forma rettangolare



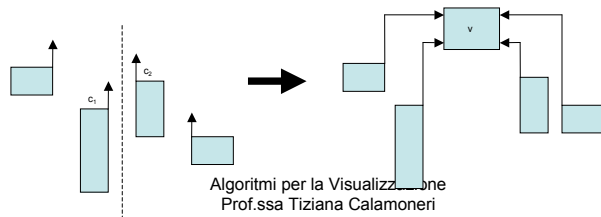
Algoritmi per la Visualizzazione  
Prof.ssa Tiziana Calamoneri

6

## Dis. 2D ort. (st-numerazione) (3)

**Dettagli dell'algoritmo**

- ogni arco ha esattamente una svolta
- per posizionare la scatola relativa a  $v$ :
  - localizza e conta tutte le colonne relative ai suoi archi entranti
  - se sono in num. pari, esistono  $c_1$  e  $c_2$  t.c.
    - $c_1$  è a sx di  $c_2$
    - sia  $c_1$  che  $c_2$  contengono un arco entrante in  $v$
    - ci sono  $\text{indeg}(v)/2$  archi a sx di  $c_1$
    - ci sono  $\text{indeg}(v)/2$  archi a dx di  $c_2$
  - $c_1$  e  $c_2$  stabiliscono come "tagliare" il disegno per inserire  $v$
  - se le colonne sono in num. dispari c'è una sola col.



Algoritmi per la Visualizzazione  
Prof.ssa Tiziana Calamoneri

7

## Dis. 2D ort. (st-numerazione) (4)

**Teorema:** Dato  $G$  2-connesso, l'algoritmo precedente produce un disegno ortogonale su griglia di  $G$  in  $O(m)$  tempo con le seguenti proprietà:

- il perimetro di ogni scatola è proporzionale al grado del nodo;
- la larghezza è al più  $m+n_{out1}+2$  e l'altezza è al più  $m/2+n_{odd}/2+n_{in1}+n_{in2}+2$  dove  $n_{out1}$  è il # di nodi con un arco uscente,  $n_{odd}$  il # di nodi con un # dispari di archi entranti,  $n_{in1}$  il # di nodi con un arco entrante ed  $n_{in2}$  il # di nodi con 2 archi entranti;
- ogni arco ha al più una svolta.

Algoritmi per la Visualizzazione  
Prof.ssa Tiziana Calamoneri

8

## Dis. 2D ort. (st-numerazione) (5)

### Dim:

- il perimetro di ogni scatola è proporzionale al grado del nodo;
  - ogni arco ha al più una svolta;
  - dimensione del disegno:
    - larghezza=somma delle larghezze delle scatole= somma di:
      - $\text{outdeg}(v)$  se  $\text{outdeg}(v) > 1$
      - $2 \text{ outdeg}(v)$  se  $\text{outdeg}(v) = 1$
      - $2$  se  $\text{outdeg}(v) = 0$  (nodo pozzo)
- Totale =  $m + n_{\text{out}1} + 2$

## Dis. 2D ort. (st-numerazione) (6)

### Dim (segue):

- dimensione del disegno:
    - altezza=somma delle altezze delle scatole= somma di:
      - $\text{indeg}(v)/2$  se  $\text{indeg}(v)$  pari +
      - $(\text{indeg}(v)+1)/2$  se  $\text{indeg}(v)$  dispari +
      - $2 \text{ indeg}(v)$  se  $\text{indeg}(v) = 1$  o  $= 2$  +
      - $2$  se  $\text{indeg}(v) = 0$  (nodo sorgente)
- Totale =  $m/2 + n_{\text{odd}}/2 + n_{\text{in}1} + n_{\text{in}2} + 2$

**CVD**

## Dis. 2D ort. (st-numerazione) (7)

### Osserva:

- non si attaccano archi alla parte inferiore delle scatole perché le colonne relative agli archi entranti non sono contigue quindi non sarebbe garantito il perimetro delle scatole limitato dal grado.
- questo algoritmo si può migliorare usando la variante di Papakostas e Tollis (coppie di riga e di colonna) per disegnare il grafo.

## Dis. 3D ortogonale (1)

### Osserva:

- il grafo di massimo grado, a parità di nodi, è  $K_n$ , e tutti i grafi di  $n$  nodi sono suoi sottografi.
- i grafi completi sono spesso critici per le considerazioni di limitazione inferiore (vedi, ad es. nel caso rettilineo).

Nel seguito dimostriamo che:

- $K_n$  non si può disegnare senza svolte, se  $n$  grande abbastanza
- dis. di  $K_n$  in vol.  $O(n^3)$  con 1 svolta per arco
- dis. di  $K_n$  in vol.  $O(n^3)$  con 2 svolte per arco
- dis. di  $K_n$  in vol.  $O(n^{2.5})$  con 3 svolte per arco

## Dis. 3D ortogonale (2)

**TH:**  $K_n$  non si può disegnare senza svolte in modo 3D ortogonale su griglia, se  $n$  grande abbastanza.

**DIM.** Indipendentemente da come siano rappresentati i nodi del grafo (come scatole o come segmenti), un suo disegno ortogonale 3D senza svolte partiziona gli archi in **tre classi**, una per ogni asse cartesiano, infatti gli archi senza svolte di un disegno ortogonale su griglia devono essere segmenti paralleli ad uno dei tre assi.

Ognuna di queste classi di archi può essere interpretata come la rappresentazione di visibilità di un sottografo del grafo di partenza.

Per completare la dimostrazione, abbiamo bisogno dei seguenti risultati...

## Dis. 3D ortogonale (3)

segue dim.

**LEMMA:**  $K_{56}$  non ammette una rappresentazione di visibilità.

**DEF. 3-Ramsey number  $R(r, b, g)$ :**

il più piccolo numero tale che ogni arbitraria partizione degli archi di  $K_{R(r, b, g)}$  che li colori rispettivamente di rosso, blu e verde induce o un sottografo  $K_r$  rosso, o un sottografo  $K_b$  blu, o un sottografo  $K_g$  verde.

Questo numero esiste ed è finito.

## Dis. 3D ortogonale (4)

segue dim.

Prendiamo  $n = R(56, 56, 56)$ .

In qualunque modo  $K_{R(r, b, g)}$  venga rappresentato come disegno rettilineo ed in qualunque modo i suoi archi vengano partizionati in tre classi che possiamo pensare colorate con i colori rosso, blu e verde, si otterrà almeno un  $K_{56}$  per la definizione di Ramsey number. Ma un  $K_{56}$  non ammette rappresentazione di visibilità per il teorema precedente.

Quindi  $K_{R(56, 56, 56)}$  non può essere rappresentato con un disegno ortogonale 3D senza svolte.

## Dis. 3D ortogonale (5)

**TH:** Esiste un dis. ortog. 3D su griglia di  $K_n$  in vol.  $O(n^3)$  ed ogni arco ha al più una svolta.

**DIM.** Dim. costruttiva (algoritmo). I **nodi** sono rappresentati da **segmenti**.

**Algoritmo:**

- \* numera i nodi in modo arbitrario  $v_1, v_2, \dots, v_n$ .
- \*  $v_i$  si rappresenti come segmento // asse  $z$  sulle coord.  $(i, i)$  del piano  $xy$ .
- \*  $e=(v_i, v_j), i < j$ , si rappresenti con una svolta per i pti  $(i, i, z), (i, j, z), (j, j, z)$ , con  $z$  da determinare.



## Dis. 3D ortogonale (6)

segue dim.

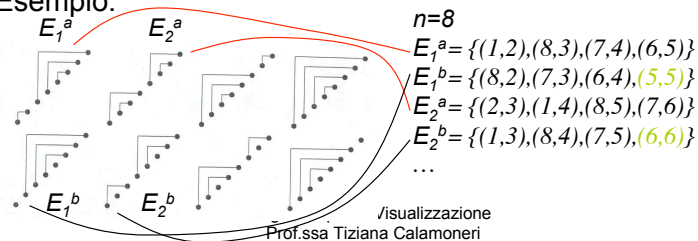
Per definire la coord. z in modo da non avere incroci, si partizionino gli archi nei seguenti insiemi:

$$E_i^a = \{(v_{i-l+1}, v_{i+l}), l = 1, 2, \dots, n/2\} \text{ ed}$$

$$E_i^b = \{(v_{i-l}, v_{i+l}), l = 1, 2, \dots, n/2 - 1\} \text{ con } i=1, 2, \dots, n/2$$

(addizioni mod n)

Esempio:



17

## Dis. 3D ortogonale (7)

segue dim.

Dobbiamo dim. che questi insiemi partizionano gli archi di  $K_n$ , e che non ci sono sovrapposizioni o incroci né tra gli archi di  $E_i^a$  né tra quelli di  $E_i^b$ .

$$E_i^a = \{(v_{i-l+1}, v_{i+l}), l = 1, 2, \dots, n/2\} \text{ N.B. } i-l+1 \text{ e } i+l: \text{ diversa parità}$$

$$E_i^b = \{(v_{i-l}, v_{i+l}), l = 1, 2, \dots, n/2 - 1\} \text{ N.B. } i-l \text{ e } i+l: \text{ stessa parità}$$

Per ogni arco  $(u,v)$ , vediamo se hanno parità diversa o uguale e risolviamo uno dei due sistemi:

$$u = i-l+1$$

$$u = i-l$$

$$v = i+l$$

$$v = i+l$$

Unica sol.  $\Rightarrow$  partizione.

Algoritmi per la Visualizzazione Prof.ssa Tiziana Calamoneri

18

## Dis. 3D ortogonale (8)

segue dim.

Per dim. che non si generano incroci, siano  $(u,v)$  e  $(x,y)$  due archi dello stesso insieme (ad es.  $E_i^a$ ).

Allora esistono  $l$  ed  $l'$  t.c.:

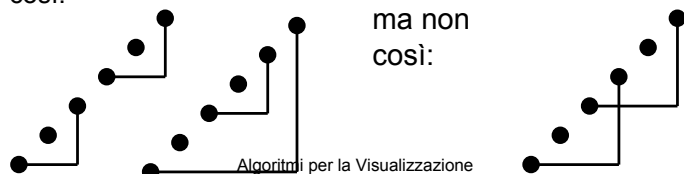
$$u = i-l+1$$

$$x = i-l'+1$$

$$v = i+l$$

$$y = i+l'$$

Facendo i conti si dimostra che gli archi possono essere messi così:



19

## Dis. 3D ortogonale (9)

segue dim.

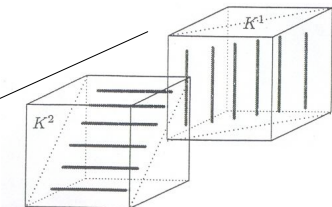
Posizioniamo ogni insieme su un diverso z-piano  $\Rightarrow n$  z-piani differenti  $\Rightarrow \text{Vol} = n \times n \times n$

**Oss.**  $E_i^a$  ed  $E_i^b$  possono essere disegnati sullo stesso z-piano riflettendo gli archi di  $E_i^b$  risp. alla diag.  $\Rightarrow \text{Vol} = n \times n \times n/2$ .

Questo risultato si può migliorare:

$\text{Vol} = n \times n/2 \times n/2$   
e meno svolte!

dis. di  $K_{n/2}$



Algoritmi per la Visualizzazione Prof.ssa Tiziana Calamoneri

20

## Dis. 3D ortogonale (10)

**TH:** Esiste un dis. ortog. 3D su griglia di  $K_n$  in vol.  $O(n^3)$  ed ogni arco ha al più due svolte.

**DIM.** Dim. costruttiva (algoritmo). I nodi sono ancora rappresentati da segmenti.

**Algoritmo:**

- \* numera i nodi in modo arbitrario  $v_1, v_2, \dots, v_n$ .
- \*  $v_i$  si rappresenti come segmento // asse z sulle coord.  $(i, 1)$  del piano xy.
- \*  $e=(v_i, v_j), i < j$ , si rappresenti con due svolte per i pti  $(i, 1, z), (i, y, z), (j, y, z), (j, 1, z)$  con y e z da determinare.

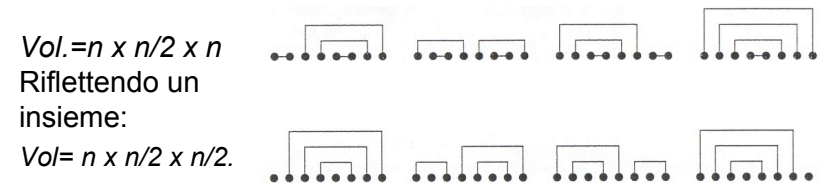


## Dis. 3D ortogonale (11)

segue dim.

La coord. z è definita analogamente a prima.

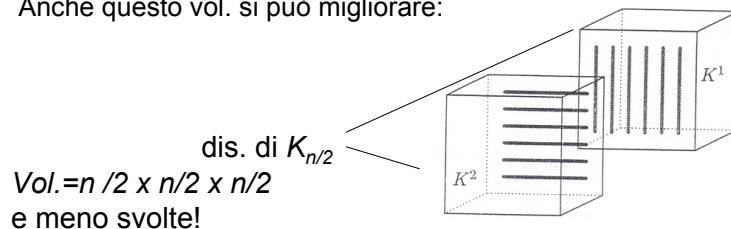
La coord. y è posta come la parte intera sup. di  $(j-i)/2$  (intuitivamente, più l'arco congiunge punti lontani e più la sua coord. y è grande).



## Dis. 3D ortogonale (12)

segue dim.

Anche questo vol. si può migliorare:



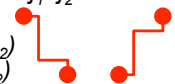
## Dis. 3D ortogonale (13)

**TH:** Esiste un dis. ortog. 3D su griglia di  $K_n$  in vol.  $O(n^{2.5})$  ed ogni arco ha al più tre svolte.

**DIM.** Dim. costruttiva (algoritmo). I nodi sono ancora rappresentati da segmenti.

**Algoritmo:**

- \* numera i nodi in modo arbitrario tramite coppie ordinate  $(i, j), 1 \leq i \leq N, 1 \leq j \leq N (N = n^{1/2})$ .
- \*  $(i, j)$  si rappresenti come segmento // asse z sulle coord.  $(2i, 2j)$  del piano xy.
- \*  $e = ((i_1, j_1), (i_2, j_2))$  senza perdere di gen.  $i_1 \leq i_2$ , e se  $i_1 = i_2$  allora  $j_1 < j_2$ .  
Se  $j_1 > j_2 \Rightarrow$  L-arco; altrimenti  $\Rightarrow$   $\Gamma$ -arco  
L-arco:  $(2i_1, 2j_1), (2i_1+1, 2j_1), (2i_1+1, 2j_2+1), (2i_2, 2j_2+1), (2i_2, 2j_2)$   
 $\Gamma$ -arco:  $(2i_1, 2j_1), (2i_1+1, 2j_1), (2i_1+1, 2j_2-1), (2i_2, 2j_2-1), (2i_2, 2j_2)$



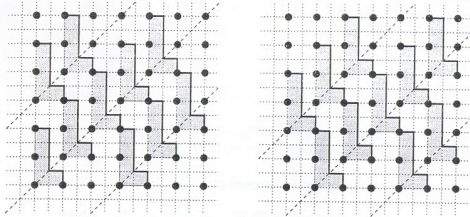
## Dis. 3D ortogonale (14)

segue dim.

Dividiamo gli  $L$ -archi ed i  $T$ -archi in  $N(N-1)$  gruppi ciascuno  $E_{dx,dy}$  con  $0 \leq dx \leq N-1$  e  $1 \leq dy \leq N-1$ .

**N.B.** Ogni gruppo  $E_{dx,dy}$  consiste degli archi  $((i_1, j_1), (i_2, j_2))$  per cui si ha:  $i_2 = i_1 + dx$  e  $j_2 = j_1 - dy$ .

Informalmente, ogni  $E_{dx,dy}$  ricopre tutti gli archi dello stesso tipo ( $L$  o  $T$ ) aventi la stessa dimensione nel disegno.



ma gli archi dello stesso tipo sono più di questi...

25

## Dis. 3D ortogonale (15)

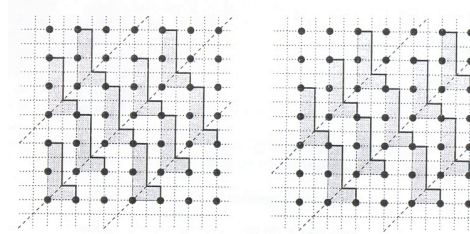
segue dim.

Dividiamo ogni gruppo  $E_{dx,dy}$  in  $dx+dy$  insiemi:

$E_{dx,dy}^p$  = insieme degli archi di  $E_{dx,dy}$  per cui  $j_2 - i_1 = p \pmod{dx+dy}$  con  $p=0, 1, \dots, dx+dy-1$ .

Informalmente, gli angoli in basso a sx (alto a sx) di ogni

$E_{dx,dy}^p$  si trovano tutti sulla stessa diagonale.



**N.B.** nessun arco in  $E_{dx,dy}^p$  incrocia con altri poiché le diag. sono a dist.  $2(dx+dy)$

me  
ri

26

## Dis. 3D ortogonale (16)

segue dim.

Assegniamo un diverso piano  $z$  ad ogni  $E_{dx,dy}^p$  e disegniamo tutti gli  $L$ -archi. Analogamente per i  $T$ -archi.

Volume:  $X=Y=N$

$Z$ : 2 volte il numero degli insiemi  $E_{dx,dy}^p$  :  
 $0 \leq dx \leq N-1$  e  $1 \leq dy \leq N-1$ ;  $p=0, 1, \dots, dx+dy-1$

$$\Rightarrow N(N-1)2N = 2N^3 = 2n^{3/2}$$

Volume =  $O(n^{5/2})$

## Estensione al caso semplicemente connesso

# Estensione a grafi semplicemente connessi

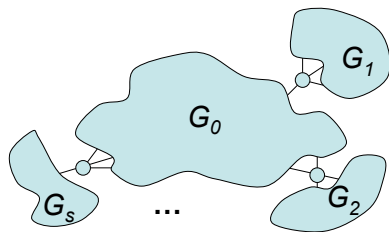
Sappiamo rappresentare grafi 2-connessi in modo ortogonale su griglia (tramite rappr. di visibilità e tramite st-numerazione).

Estendiamo a grafi semplicemente connessi in due modi diversi...

# Primo metodo (1)

- Dato  $G$  qualunque di max gr. 4, in tempo  $O(n)$  è possibile determinare le sue componenti 2-connesse (*blocchi*) e i suoi punti di articolazione.
- Dall'alg. di Biedl-Kant: ogni blocco con  $n$  nodi si può disegnare con area  $(n+1) \times n$  e al più  $2n-1$  svolte.
- L'algoritmo disegna tutti i blocchi separatamente. Problema: punti di articolazione?

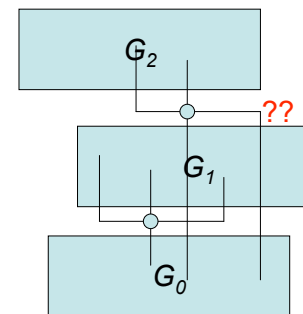
# Primo metodo (2)



$s$  blocchi connessi a  $G_0$  tramite pti di artic.  $v_1, v_2, \dots, v_s$ .

- $G_0$  biconnesso e disegnato con Biedl-Kant.
- $v_1$  attaccato inserendo una nuova riga.
- $G_1$  biconnesso e disegnato con Biedl-Kant, attaccando al resto del disegno.
- e così via...

# Primo metodo (3)

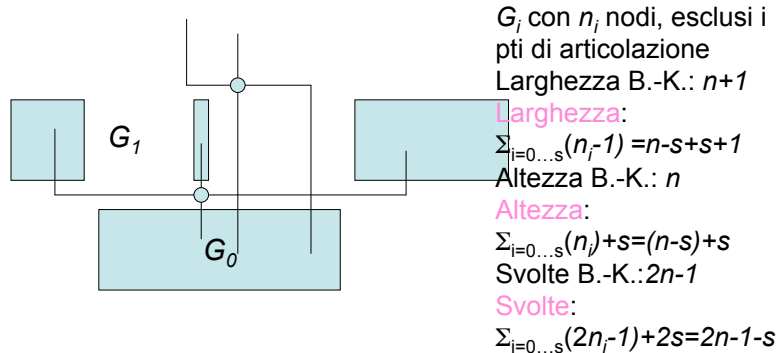


## Algoritmo:

- $G_0$  biconnesso e disegnato con Biedl-Kant.
- $v_1$  attaccato inserendo una nuova riga.
- $G_1$  biconnesso e disegnato con Biedl-Kant, dividendolo in al più 3 "fette" - una per ogni connessione - con quella centrale di largh. 1 e attaccando al resto del disegno.
- e così via...



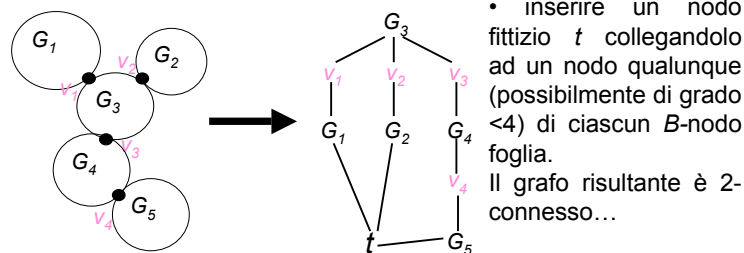
## Primo metodo (4)



## Secondo metodo (1)

- Metodo dello' *aumentazione*.
- Si consideri l'albero  $T$  dei blocchi e dei pti di articolazione definito come:
  - $T$  ha un  $B$ -nodo per ogni blocco
  - $T$  ha un  $C$ -nodo per ogni pto di articolazione
  - un arco collega un  $B$ -nodo ad un  $C$ -nodo se il pto di art. associato al  $C$ -nodo appartiene al blocco associato al  $B$ -nodo
- Il grafo così costruito non può avere cicli  $\Rightarrow$  è un albero.
- Si può costruire in tempo lineare nella dimensione del grafo di partenza

## Secondo metodo (2)



- calcolare una st-numerazione,
- eliminare  $t$  dal grafo ottenendo una sorgente e molti pozzi,
- applicare B.-K.

## Secondo metodo (3)

- questo metodo non garantisce la stessa area e lo stesso # di svolte del caso 2-connesso (crescono entrambe con il # di pozzi), a meno che tutti i pozzi siano di grado  $<4$
- è molto più facile da implementare del metodo precedente


## Estensione a grafi non planari

## Planarizzazione (1)

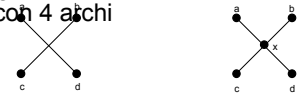
- Tecniche di planarizzazione motivate dalla disponibilità di molti algoritmi efficienti per grafi planari
- Idea: dato  $G$  non planare, si può trasformarlo in planare ed eseguire un algoritmo per grafi planari
- Problema: queste tecniche modificano  $G$  in input, quindi bisogna tentare di minimizzare le modifiche
- Soluzioni possibili:
  - minimo # di incroci
  - massimo sottografo planare

entrambe NP-ardue...

## Planarizzazione (2)

- Unica soluzione possibile: euristiche
- **Metodo 1:**
  - cancella nodi fino a quando  $G$  risultante sia planare
  - tecnica poco usata (1. NP-completo decidere se basta cancellare  $k$  nodi; 2. il grafo è un altro!!)
- **Metodo 2:**
  - vertex splitting 
  - stessi problemi del metodo precedente (1. NP-completo decidere se basta splittare  $k$  nodi; 2. il grafo è un altro!!)

## Planarizzazione (3)

- **Metodo 3:**
    - rimpiazza gli incroci con dei nodi e i due archi che incrociano con 4 archi 
- Una *planarizzazione di  $G$*  è un grafo ottenuto tramite una successione di queste operazioni, ed il grafo risultante è planare.
- Ogni grafo ammette una planarizzazione
  - Per passare dal disegno del grafo planare al disegno del grafo originario si devono ri-sostituire gli incroci ai nodi fittizi.
  - **N.B.** il risultato non è planare e non è minimizzato il # di incroci!!

# Planarizzazione (3)

- Metodo 4:
  - rimuovi archi finché  $G$  risultante è planare.
  - N.B. è NP-completo decidere se basta rimuovere  $k$  archi per rendere  $G$  planare
  - Soluzione 1: ridursi ad un albero ricoprente (troppo pochi archi rimasti!!)
  - Soluzione 2: passare per la sol. del problema MAX PLANAR SUBGRAPH. NP-arduo  $\Rightarrow$  MAXIMAL PLANAR SUBGRAPH
- MAXIMAL PLANAR SUBGRAPH:
  - parti da  $G'$  vuoto
  - finché ci sono archi non controllati  
inserisci un arco di  $G$  se non introduce incroci
  - complessità:  $O(nm)$  - test di planarità ad ogni passo
  - [Di Battista & Tamassia '90]: complessità migliorata ad  $O(m \log n)$  usando una struttura dati sofisticata (SPQR-trees)