

4.2. Disegno 3D

Riferimenti: [KW01] pag. 172

Per approfondire: [DW04] pagg. 1-16.

C'è un numero enorme di metodi ed algoritmi per la creazione di visualizzazioni di grafi esteticamente gradevoli in due dimensioni. Anche se ciò non è ancora vero per disegni tridimensionali, tuttavia quest'area di ricerca si sta espandendo sempre più, motivata in parte dall'evoluzione dell'*hardware* per la grafica 3D, dalle applicazioni in *information visualization*, nella progettazione di circuiti VLSI e nell'area del *software engineering* [DW04]. Infine, rappresentare grafi in tre dimensioni offre vari benefici.

La dimensione aggiuntiva dà grande flessibilità per posizionare i nodi e gli archi del grafo; inoltre, è SEMPRE possibile evitare gli incroci. D'altra parte, sorgono nuovi problemi che hanno bisogno di essere risolti: gli attuali supporti di output hanno natura bidimensionale e forniscono una risoluzione comunque limitata. Perciò, la visualizzazione completa del grafo, specie se molto grande, risulta difficile da riassumere. Questo svantaggio viene indebolito dall'uso delle operazioni navigazionali come la rotazione, lo spostamento, e lo zoom. Queste operazioni, infatti, permettono un uso più completo dello spazio sullo schermo e permettono all'utente di risolvere le ambiguità (come ad esempio gli incroci apparenti) mentre viene mantenuta la sua mappa mentale (per la definizione formale, si veda oltre).

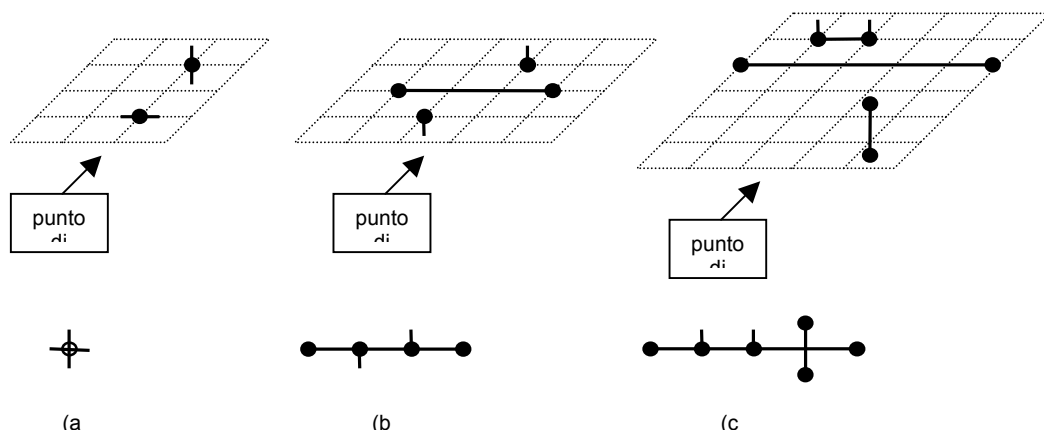
Per mantenere la perdita di informazione il più limitata possibile è necessario proiettare l'immagine 3D sul piano 2D. Il parametro principale per valutare la bontà di una proiezione è il punto di vista (*viewpoint*), cioè la posizione dell'osservatore insieme con la direzione dello sguardo.

Un primo punto di partenza per determinare il miglior punto di vista è l'osservazione che si deve preservare il messaggio del disegno. Ad esempio, si può considerare un buon punto di vista quello per cui gli archi non appaiono mai collineari, oppure quello che minimizza il numero di incroci apparenti, o infine quello che evita che più punti 3D vengano mappati nello stesso punto 2D.

Nel 1997, Webber [W97] definisce un buon punto di vista come quello che evita ciò che chiama occlusioni, che possono essere catalogate in 5 tipi diversi (v. figura 4.20):

- nodo-nodo: 2 nodi vengono mappati sullo stesso punto (v. figura 4.20 parte sinistra)
- nodo-arco: un nodo viene mappato come un punto interno di un arco (v. figura 4.20 parte centrale)
- arco-nodo: un arco attraversa un nodo che non è un suo estremo (v. figura 4.20 parte centrale)
- incrocio apparente (v. figura 4.20 parte destra)
- arco-arco: sovrapposizione di due archi, per cui uno 'scompare' (v. figura 4.20 parte destra).

Figura 4.20.



Webber presenta degli algoritmi per trovare il miglior punto di vista che eviti tutte le occlusioni. Introduce una misura di separazione rotazionale, in cui la qualità del punto di vista è definita come l'angolo misurato tra la direzione del punto di vista e il più vicino punto di occlusione. Il miglior punto di vista è quello che massimizza tale angolo.

Possiamo introdurre anche una misura di separazione osservata, in cui la qualità del punto di vista è definita come la più breve distanza euclidea tra le proiezioni dei due elementi che causano un'occlusione. Il miglior punto di vista ora è quello che massimizza la distanza tra la più vicina coppia che potrebbe generare occlusioni.

La misura di separazione rotazionale è più indicata per il disegno 3D interattivo (vedi sezione 4.3), mentre la misura di separazione osservata per il disegno statico. Il problema è che la complessità degli algoritmi per determinare il miglior punto di vista sotto una di queste misure è troppo alta ($O(m^4 \log m)$). Per questo sono stati sviluppati degli algoritmi di approssimazione veloci [W97].

A causa di queste ed altre problematiche, gli algoritmi di visualizzazione 3D per grafi, specie se grandi, non producono rappresentazioni appropriate per una buona visualizzazione. Il valore dei risultati sul disegno 3D ortogonale (a parte speciali casi di applicazione al VLSI) risiede per lo più nell'analisi teorica degli algoritmi.

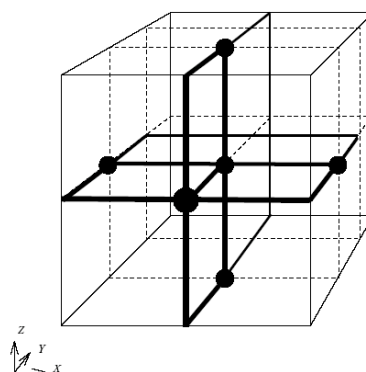
Nelle tre dimensioni, sono stati studiati diversi tipi di rappresentazione, che differiscono tra loro per il modo di rappresentare i nodi e gli archi. Nel seguito ci concentreremo sui tipi più diffusi: il disegno ortogonale ed il rettilineo.

4.2.1. Disegno ortogonale 3D

Riferimenti: [KW01] pp. 172, 176-180, 190-192; [ESW96] tutto.

Un disegno ortogonale 3D posiziona i nodi su punti della griglia a coordinate intere e gli archi vengono rappresentati come sequenze di segmenti contigui che corrono sulle linee della griglia. E' permesso agli archi di contenere svolte ma NON incroci. Questa non è, in verità, una grossa restrizione: è facile rendersi conto che al più raddoppiando le tre dimensioni del volume del disegno, gli incroci si possono sempre evitare. In Figura 4.21 è raffigurato il disegno ortogonale 3D di un grafo.

Figura 4.21.



Poiché ogni punto di griglia è l'intersezione di tre linee della griglia ogni grafo che ammette una rappresentazione 3D ortogonale ha al più grado 6.

Le misure più comuni per determinare la qualità della rappresentazione sono il volume del parallelepipedo circoscritto (*bounding box*) ed il massimo numero di svolte per arco. Usando delle

estensioni dei risultati di NP-completezza noti per il 2D, si dimostra che è NP-arduo minimizzare ciascuna di queste due misure.

Per quanto riguarda il numero di svolte per arco, possiamo osservare intanto che, così come nel caso bidimensionale, non possiamo garantire che non vi siano svolte, poiché ogni disegno ortogonale del triangolo richiede almeno una svolta; inoltre per rappresentare un arco tra due nodi che non siano sulla stessa linea della griglia si richiede di nuovo almeno una svolta; mentre per rappresentare un arco tra due nodi che non sono ne' sulla stessa riga ne' sulla stessa colonna della griglia si richiedono almeno due svolte; ad esempio K_5 ha un arco che richiede almeno 2 svolte [W03]. Questo ragionamento fornisce una limitazione inferiore sul numero di svolte per arco di un disegno ortogonale 3D.

Tra l'altro, ottimizzare il volume e tenere basso il numero di svolte per arco sembrano obiettivi conflittuali: cercando di abbassare il numero di svolte per arco si innalza il valore del volume, come si vede dai seguenti risultati:

- Nel 1996 Eades, Symvonis e Whitesides [ESW96] progettano un algoritmo per disegnare grafi in 3D in modo ortogonale in un volume $O(n^{3/2})$ e con al più 7 svolte per arco; dal punto di vista del volume, questo è il miglior risultato possibile, poiché Kolmogorov e Bardzin, nel 1967 [KB67] avevano dimostrato che questo volume è il minimo che ci si possa aspettare.
- Nel 1998 Wood [W98] presenta un algoritmo per disegnare grafi con un massimo di 4 svolte per arco, ma il volume richiesto è $2.37n^3$.
- Eades, Symvonis e Whitesides [ESW96] mostrano che il numero di svolte per arco può essere ridotto a 3 aumentando il volume di un fattore costante, $V=27n^3$.
- Facendo la restrizione che il grafo in input abbia massimo grado 5, Wood [W98b] migliora ulteriormente il risultato: in un volume n^3 si può disegnare un grafo con un massimo di 2 svolte per arco; egli inoltre congetture che potrebbe esistere un grafo di massimo grado 6 che non può essere disegnato con 2 svolte per arco.

Il problema di capire se 2 svolte per arco siano sufficienti per i grafi di massimo grado 6 rappresenta tuttora il più importante problema aperto in quest'area, e rimane anche quando ci si riduce a delle sottoclassi di grafi, come i planari, i grafi serie-parallelo e gli outerplanar di massimo grado 6. E' stato congetturato [ESW96] che un candidato naturale per rispondere negativamente a questo problema aperto fosse rappresentato dal K_7 , tuttavia Wood [W03] ha trovato un disegno ortogonale di questo grafo con 2 svolte per arco, lasciando ancora una volta il problema aperto insoluto.

Verranno di seguito presentati due algoritmi che visualizzano grafi di massimo grado 6 in modo ortogonale, il primo con al più 7 svolte per arco in area $O(n^{1/2}) \times O(n^{1/2}) \times O(n^{1/2})$, il secondo con al più 3 svolte per arco in area $O(n) \times O(n) \times O(n)$. Il contrasto tra questi due risultati evidenzia il già citato trade-off tra il numero di svolte per arco e la dimensione del disegno.

Entrambi gli algoritmi sfruttano la stessa fase di preprocessing basata sulla colorazione degli archi di un grafo, che sarà quindi descritta prima.

4.2.1.1. Fase di Preprocessing

Richiamiamo innanzi tutto alcune definizioni e teoremi della teoria dei grafi [LW92].

Definizione: Una *copertura di cicli* (cycle cover) di un grafo orientato è un sottografo ricoprente che consiste di cicli orientati (N.B. sono possibili sovrapposizioni, perché stiamo parlando di una copertura e non di una partizione).

Definizione: Un *circuito euleriano* è un cammino che tocca tutti gli archi una e una sola volta tornando al punto di partenza. Un grafo G si dice *grafo euleriano* se ammette un circuito euleriano.

Teorema (di Eulero). *Un grafo $G=(V,E)$ connesso è euleriano se e solo se tutti i suoi nodi hanno grado pari.*

Dimostrazione. Condizione necessaria: Sia G euleriano. Allora, per definizione esiste un circuito euleriano C che attraversa una e una sola volta tutti gli archi del grafo. Il circuito C tocca tutti i vertici di G . Inoltre C entra ed esce in ogni vertice (tranne che per il primo ed ultimo vertice da cui esce all'inizio e in cui entra alla fine) senza mai passare due volte sullo stesso arco. Quindi tutti i vertici devono avere necessariamente grado pari.

Condizione Sufficiente: Per ipotesi, ogni vertice di G ha grado pari. Costruiamo un circuito euleriano. Sia x un vertice di G . Usciamo da x attraverso un arco, entrando in un altro vertice y di G . Poiché y ha grado pari, possiamo uscire da y con un arco diverso da quello con cui siamo entrati. Possiamo ripetere il procedimento senza passare mai due volte su uno stesso arco e, poiché il grafo è finito, prima o poi torneremo sul vertice x . Abbiamo così costruito un ciclo, che non è necessariamente euleriano, perché potremmo non aver attraversato tutti gli archi del grafo. Allora consideriamo il grafo G' ottenuto da G cancellando gli archi del ciclo. Se tutti i vertici di G' hanno grado 0 allora il ciclo costruito è euleriano. Altrimenti consideriamo un vertice x' appartenente al ciclo rimosso che abbia ancora grado positivo. Tale vertice deve esistere perché G è connesso. Ripetiamo il procedimento sul grafo G' a partire da x' . Otteniamo un nuovo ciclo, che interseca il precedente nel vertice x' . L'unione dei due cicli è essa stessa un ciclo. Iterando il procedimento, prima o poi tutti gli archi del grafo verranno usati, ottenendo così un ciclo euleriano. **CVD**

Definizione. Un grafo si dice *bipartito* se l'insieme dei vertici si può ripartire in due sottoinsiemi disgiunti V_1 e V_2 in modo che gli archi uniscano un vertice di V_1 con uno di V_2 .

Definizione. In un grafo bipartito, un *accoppiamento* (*matching*) M da V_1 a V_2 è un sottoinsieme di archi disgiunti, cioè senza estremi in comune.

Definizione. Un accoppiamento M si dice *completo* se $|M|=|V_1|$, dove V_1 è l'insieme con cardinalità minore tra i due in cui è stato partizionato l'insieme dei nodi V .

Teorema (di Philip Hall). *Sia $G=(V_1,V_2,E)$ un grafo bipartito con $|V_1| \leq |V_2|$. Allora G ha un accoppiamento completo tra V_1 e V_2 se e solo se per ogni insieme S di k nodi di V_1 vi sono almeno k nodi di V_2 adiacenti ad uno dei nodi di S .*

Sia $adj(S)$ l'insieme dei nodi di V_2 adiacenti a qualche nodo di S , cioè $adj(S) = \{v \in V_2 : \exists u \in S, (u,v) \in E\}$, allora la condizione di Hall diventa: $|S| \leq |adj(S)|$ per ogni $S \subseteq V_1$.

Dimostrazione. Condizione necessaria: Se G ha un accoppiamento completo M ed S è un qualsiasi sottoinsieme di V_1 , allora ogni vertice in S è accoppiato da M con un differente nodo in $adj(S)$, segue che $|S| \leq |adj(S)|$.

Condizione sufficiente: Dimostriamo che se la condizione di Hall è verificata, allora esiste un accoppiamento completo. Per assurdo, sia verificata la condizione di Hall, ma non esista un accoppiamento completo, cioè sia M un accoppiamento massimo, con $|M| < |V_1|$; dimostriamo che esiste un accoppiamento M' con $|M'| = |M| + 1$, giungendo ad una contraddizione. L'idea della dimostrazione è costruire un cammino i cui archi siano alternativamente in M e fuori di M .

Diremo, per abuso di linguaggio, che un nodo appartiene ad M se è estremo di un arco di M . Per ipotesi, $|M| < |V_1|$, e dunque esiste un $u_0 \in V_1$, tale che $u_0 \notin M$. Sia $S = \{u_0\}$; per tale insieme vale che $1 = |S| \leq |adj(S)|$ per ipotesi, e quindi esiste un nodo $v_1 \in V_2$ adiacente ad u_0 . Se $v_1 \notin M$, allora $M' = M \cup \{e\}$, dove $e = (u_0, v_1)$, è l'accoppiamento richiesto. Quindi $v_1 \in M$, ed è accoppiato tramite M ad un qualche $u_1 \in V_1$, allora prendendo $S = \{u_0, u_1\}$ si ha $2 = |S| \leq |adj(S)|$, e deve esistere un altro nodo v_2 , distinto da v_1 , e adiacente ad u_0 o ad u_1 . Se $v_2 \notin M$, si hanno due possibilità:

- v_2 adiacente ad u_1 . Consideriamo il cammino v_2, u_1, v_1, u_0 , nel quale l'arco $e=(u_1, v_1)$ sta in M , mentre $f=(u_1, v_2)$ e $g=(u_0, v_1)$ no. Formiamo allora il nuovo accoppiamento M' aggiungendo a M gli archi f e g e togliendo e . M' ha un arco in più di M .
- v_2 adiacente ad u_0 . Formiamo il nuovo accoppiamento M' aggiungendo ad M l'arco (u_0, v_2) . Anche qui M' ha un arco in più di M .

Allora deve risultare $v_2 \in M$; v_2 è adiacente ad u_0 o ad u_1 con un arco che non sta nell'accoppiamento (in quanto $u_0 \notin M$ ed $u_1 \in M$ e u_1 è accoppiato con v_1), ma poiché sta in M deve stare su un arco di M . Dunque esiste $u_2 \in M$, $u_2 \neq u_0, u_1$, adiacente a v_2 . Prendiamo $S = \{u_0, u_1, u_2\}$. Avendosi $|S| = 3$ esiste un terzo vertice $v_3 \in V_2$ adiacente ad uno dei tre. Sia $v_3 \notin M$. Vi sono 3 casi:

- v_3 adiacente ad u_2 . Nel cammino v_3, u_2, v_2, u_0 l'arco (u_2, v_2) appartiene all'accoppiamento M , gli altri due no. Si ottiene un nuovo accoppiamento M' togliendo ad M l'arco (u_2, v_2) e aggiungendo gli altri due, e si ha $|M'| = |M| + 1$.
- v_3 adiacente ad u_1 . Nel cammino v_3, u_1, v_1, u_0 , cambiando stato al primo e al terzo arco si ottiene $|M'| = |M| + 1$.
- v_3 adiacente ad u_0 . Nel cammino v_3, u_0, v_1, u_1 si proceda come nei casi precedenti cambiando stato a (v_3, u_0) .

Continuando in questo modo, per la finitezza del grafo, si arriva necessariamente ad un nodo v_r che non appartiene ad M . Ognuno dei nodi v_i è adiacente ad almeno uno tra u_0, u_1, \dots, u_{i-1} . Come nel caso $r = 2$ si ha un cammino $v_r, u_{r-1}, v_{r-1}, u_{r-2}, v_{r-2}, \dots, v_1, u_0$, nel quale gli archi $e_i = (u_i, v_i)$ appartengono ad M , mentre gli $e_j = (v_j, u_{j-1})$ no. Costruiamo allora un nuovo accoppiamento M' togliendo da M gli e_i e aggiungendo gli e_j . Poiché i due archi esterni (v_r, u_{r-1}) e (v_1, u_0) sono in M' , questo accoppiamento contiene un arco in più di M .

CVD

Corollario. Sia $G=(V_1, V_2, E)$ un grafo bipartito k -regolare con $|V_1|=|V_2|$. Allora G contiene k accoppiamenti completi.

Dimostrazione. Sia S un sottinsieme di V_1 . L'insieme $adj(S)$ avrà al più $k|S|$ nodi (se ciascun nodo in $adj(S)$ ha grado 1 nel sottografo indotto da $S \cup adj(S)$) ed almeno $|S|$ nodi (se ciascun nodo in $adj(S)$ ha grado k nel sottografo indotto da $S \cup adj(S)$). In tutti i casi la condizione di Hall è verificata, e quindi esiste un accoppiamento completo, che può essere rimosso dal grafo dando luogo ad un nuovo grafo $(k-1)$ -regolare. Per esso possiamo ripetere il ragionamento, giungendo all'asserto. CVD

Teorema. Sia $G=(V, E)$ un grafo non orientato di massimo grado 6. Allora esiste un grafo orientato $G'=(V', E')$ con $V = V'$ tale che:

- per ogni vertice v di V' si ha $indeg(v) = outdeg(v) = 3$;
- G è un sottografo di G' senza l'orientamento;
- gli archi di G' possono essere partizionati in 3 coperture di cicli disgiunte fra loro.

Inoltre, dato $G=(V, E)$ il grafo orientato G' e i sue 3 cicli di copertura possono essere calcolati in $O(n^{3/2})$ tempo.

Dimostrazione. La dimostrazione è costruttiva, quindi la tecnica che seguiremo è quella di costruire il grafo G' e verificarne le proprietà.

Passo 1. Per ogni coppia di nodi a, b in V , tale che $deg(a)$ e $deg(b)$ siano entrambi dispari, si aggiunga un arco che abbia come estremi a e b . Questo può sempre essere fatto per il corollario al lemma della stretta di mano.

Passo 2. Poiché al termine del passo 1. ogni nodo ha grado pari e al massimo 6, si può procedere come segue:

- per ogni vertice v in V , di grado 4, aggiungi un cappio (v, v) ;

- per ogni vertice v in V , di grado 2, aggiungi 2 cappi (v,v) .

In questo modo avremo creato un grafo regolare (con eventuali cappi ed archi multipli) di grado 6 sovragrafo del grafo originario G . Poiché ogni nodo del grafo ha grado pari, il grafo è euleriano, ed un ciclo euleriano può essere trovato in $O(m) = O(n)$ tempo. Diamo agli archi l'orientamento indotto dal ciclo euleriano.

Osserviamo che tutte le operazioni fin qui eseguite hanno complessità $O(n)$, in quanto $|E| = 3|V|$. Inoltre, per ogni vertice v , il numero di archi entranti in v è uguale al numero di archi uscenti da v , cioè $\text{indeg}(v) = \text{outdeg}(v)$. Sia G' il grafo fin qui costruito. Ora dobbiamo determinarne la copertura di cicli.

Passo 3. Da G' costruiamo un grafo $G'' = ((V_{OUT} \cup V_{IN}), E'')$ bipartito, non orientato, 3-regolare (esattamente 3 archi per nodo), dove:

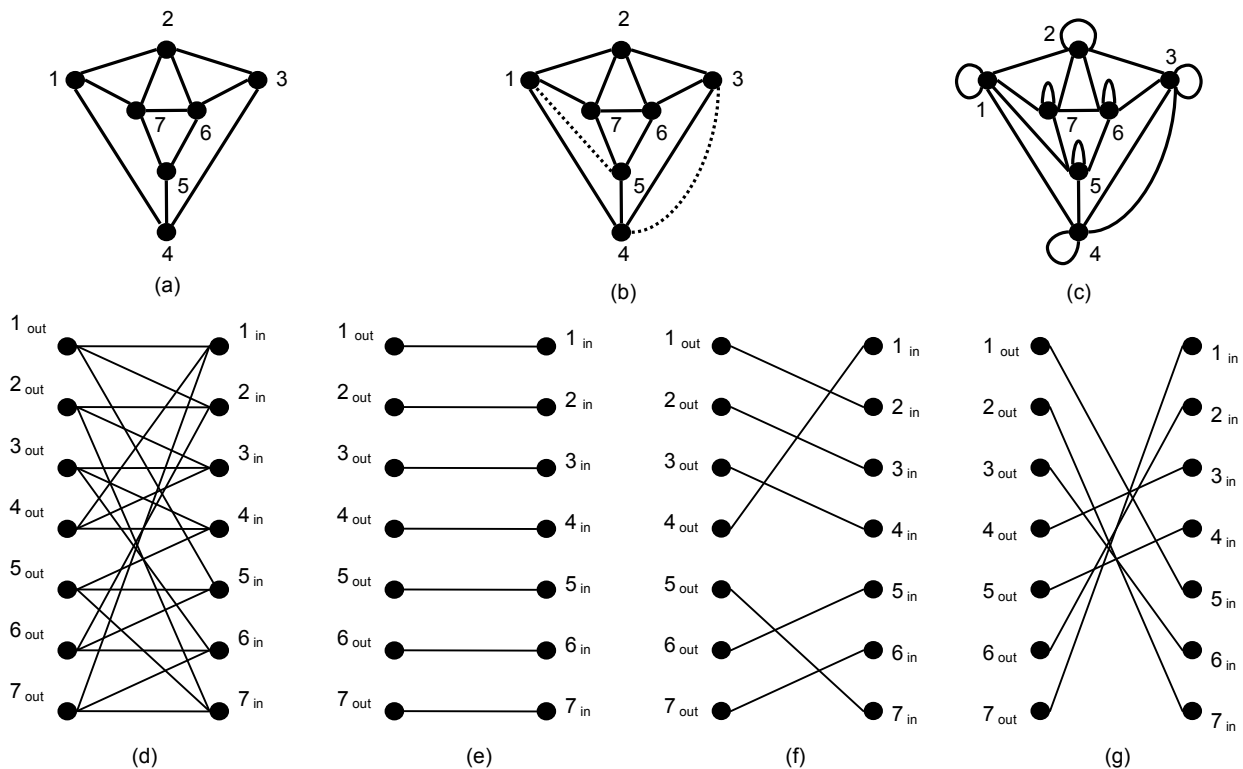
- $V_{out} = \{v_{out} \mid v \text{ in } V\}$
- $V_{in} = \{v_{in} \mid v \text{ in } V\}$
- $E'' = \{(u_{out}, v_{in}) \mid (u,v) \text{ in } E'\}$.

In pratica, la costruzione di G'' consiste nello sdoppiare ciascun vertice v in due vertici v_{in} e v_{out} e nel collegare ciascun v_{out} con gli archi uscenti da v e ciascun v_{in} con gli archi entranti in v .

G'' soddisfa le ipotesi del corollario al teorema di Hall e pertanto contiene 3 accoppiamenti completi. Individuiamo un accoppiamento completo, coloriamo i suoi archi di rosso e rimuoviamoli dal grafo. Il grafo rimanente è un grafo bipartito 2-regolare, quindi esso contiene ancora un accoppiamento completo; coloriamo i suoi archi di verde e rimuoviamoli. I rimanenti archi formano un accoppiamento completo, che coloriamo di blu.

Passo 4. L'algoritmo colora ogni arco diretto (u,v) di G' con il colore assegnato all'arco (u_{out}, v_{in}) di G'' . Questo fornisce ad ogni nodo di G' esattamente un arco in entrata e un arco in uscita di ciascun colore e gli archi di G' sono partizionati in 3 sottografi colorati $C_{red}, C_{green}, C_{blue}$, ognuno dei quali è una copertura di cicli di G' , e le tre coperture sono disgiunte tra loro per la definizione di accoppiamento.

Figura 4.22



Poiché il massimo accoppiamento di un arbitrario grafo bipartito con n vertici ed m archi può essere calcolato in $O(m\sqrt{n})$ tempo, e poiché per G'' abbiamo $n = 2|V|$ ed $m = 3|V|$, il calcolo delle 3 coperture di cicli può essere fatto in $O(|V|^{3/2})$ tempo, e include il tempo impiegato per calcolare G' da G . **CVD**

Esempio. In figura 4.22 è mostrato un esempio di applicazione dell'algoritmo descritto nella dimostrazione precedente: la figura 4.22(a) rappresenta il grafo di partenza; le figure 4.22(b) e 4.22(c) mostrano la situazione dopo i passi 1 e 2 dell'algoritmo (inserimento archi fittizi per portare il grafo ad essere regolare di grado 6); su questo grafo calcoliamo il ciclo euleriano, che è (1,1) (1,2) (2,2) (2,3) (3,3) (3,4) (4,4) (4,1) (1,5) (5,5) (5,4) (4,3) (3,6) (6,6) (6,2) (2,7) (7,6) (6,5) (5,7) (7,7) (7,1). Da tale ciclo deduciamo il grafo bipartito 3-regolare G'' mostrato in figura 4.22(d); nelle figure 4.22(e), 4.22(f) e 4.22(g) sono, infine, raffigurati i 3 accoppiamenti da cui si potranno dedurre C_{blue} , C_{red} e C_{green} . Ad esempio, C_{red} (figura 4.22(f)) è (1,2) (2,3) (3,4) (4,1), (5,7) (7,6) (6,5).

L'algoritmo appena visto nella dimostrazione sarà usato come fase di preprocessing dai due algoritmi di disegno che seguono, dando in input un grafo G non orientato di massimo grado 6 e ottenendo in output un grafo G' orientato che contiene G e che ha tre coperture di cicli disgiunte, C_{red} , C_{green} , C_{blue} . Entrambi gli algoritmi che seguono specificano il percorso degli archi di ciascuna delle 3 coperture di cicli di G' ; cappi e archi di G' che non sono presenti in G verranno ignorati nella fase di disegno.

4.2.1.2. Il primo algoritmo

Questo algoritmo, dovuto a Eades, Symvonis e Whitesides [ESW96], prende come input un grafo di n vertici e di massimo grado 6, e produce in output un disegno su griglia di volume pari a $O(\sqrt{n}) \times O(\sqrt{n}) \times O(\sqrt{n})$, ovvero $O(n^{3/2})$. Ogni arco ha lunghezza $O(\sqrt{n})$ e contiene al massimo 7 svolte per arco.

L'algoritmo è costituito dai seguenti passi:

1. Eseguire l'algoritmo di preprocessing per costruire un grafo diretto aciclico G' e ottenere una partizione dei suoi archi in tre coperture di cicli disgiunte, C_{red} , C_{blue} e C_{green} .
2. Usare la copertura di cicli C_{red} per posizionare i nodi sul piano $z=0$; disegnare il percorso degli archi in C_{red} sullo stesso piano con al più 6 svolte per arco.
3. Disegnare gli archi della copertura di cicli C_{blue} nel semispazio $z \geq 0$ con al massimo 7 svolte per arco.
4. Disegnare gli archi della copertura di cicli C_{green} nel semispazio $z \leq 0$ con al massimo 7 svolte per arco.
5. Eliminare tutto ciò che non è presente in G .

Dettagliamo ora le varie fasi dell'algoritmo, cominciando con il ricordare che il passo 1. coincide con la fase di preprocessing descritta nel paragrafo precedente.

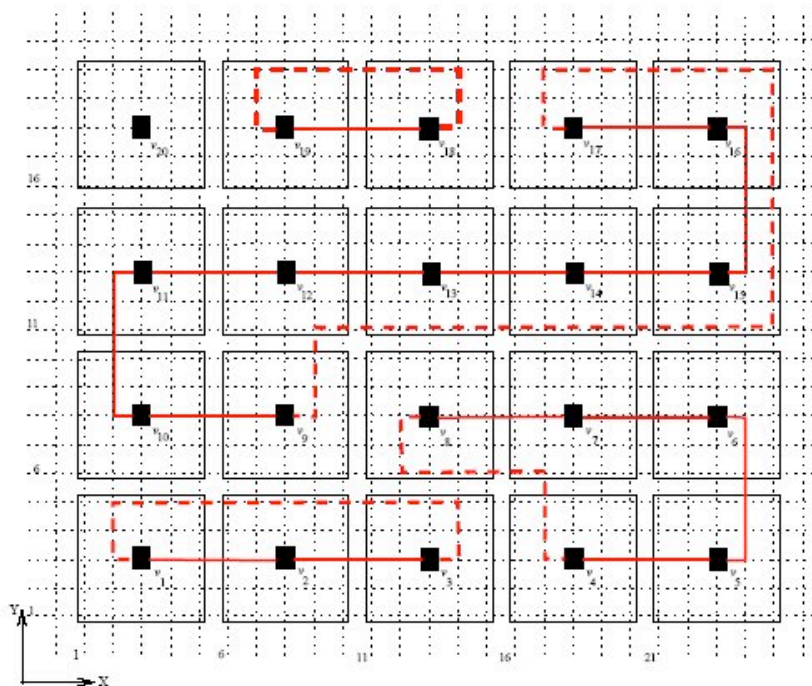
Il passo 2. consiste nello sfruttare C_{red} per posizionare tutti i nodi del grafo e nel disegnare tutti gli archi di C_{red} stesso. Supponiamo che C_{red} consista di k cicli orientati c_1, c_2, \dots, c_k . Assegniamo un ordinamento ai vertici di G' come segue: scegliamo arbitrariamente un vertice da c_1 e ordiniamo i vertici restanti del ciclo a partire dal primo vertice scelto; in seguito ordiniamo i vertici dei rimanenti cicli in modo analogo, così che i vertici di c_i vengano prima di quelli di c_j se $i < j$. Definiamo ora, sul piano $z=0$, una griglia in cui sono individuati dei punti speciali $p_{i,j}$, in modo

tale che $p_{i,j} = (5i + 3, 5j + 3, 0)$ per ogni $0 \leq i, j < \sqrt{n}$. In questo modo, tutti i punti speciali si trovano distanti tra loro di almeno 5 unità e dal bordo della griglia di almeno 3 unità, sia verticalmente che orizzontalmente.

Piuttosto che dare una formula esplicita su come disporre i vertici e il percorso degli archi di C_{red} sulla griglia, mostriamo un esempio dal quale si possano estrapolare poi facilmente le regole generali.

Consideriamo la seguente copertura di cicli C_{red} : $c_1 = \langle v_1, v_2, v_3 \rangle$, $c_2 = \langle v_4, \dots, v_8 \rangle$, $c_3 = \langle v_9, \dots, v_{17} \rangle$, $c_4 = \langle v_{18}, v_{19} \rangle$ e $c_5 = \langle v_{20} \rangle$. Usando l'ordine fissato in precedenza per i cicli, assegniamo ai vertici di G' i punti $p_{i,j}$ sulla griglia seguendo un ordine a serpentina, partendo dal punto speciale in basso a sinistra, come illustrato in Figura 4.23. Successivamente posizioniamo gli archi dei cicli di C_{red} facendo attenzione a non usare mai le colonne su cui giacciono i vertici, usciamo, cioè, sempre lungo la direzione orizzontale. Si possono presentare vari casi:

Figura 4.23.



- i vertici di uno stesso ciclo giacciono sulla stessa riga;
- i vertici di uno stesso ciclo giacciono su due righe consecutive;
- i vertici di uno stesso ciclo giacciono su più di due righe.

Tutti e tre i casi sono presentati in figura 4.23, ad esempio l'arco (v_1, v_3) è del primo tipo, l'arco (v_4, v_8) del secondo e l'arco (v_9, v_{17}) del terzo. Disegnando gli archi come mostrato in figura, avremo al massimo 6 svolte per arco. A questo punto abbiamo un disegno 2D della copertura di cicli rossa; disegniamo ora le altre due coperture sfruttando la terza dimensione.

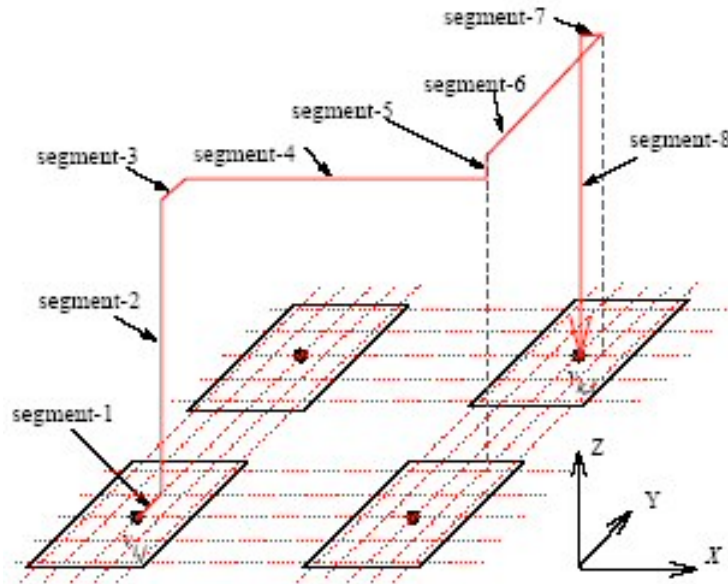
Il passo 3. consiste nel sistemare C_{blue} nel semispazio $z \geq 0$. Gli archi di C_{green} sono posizionati in modo analogo nel semispazio $z \leq 0$. Il percorso di un arbitrario arco di C_{blue} viene mostrato in Figura 4.24.

I vertici v e w sono assegnati, rispettivamente, ai punti speciali $p_{i,j}$ e $p_{k,l}$ aventi coordinate $(x_v, y_v, 0)$ e $(x_w, y_w, 0)$, dove $x_v = 5i + 3$, $y_v = 5j + 3$, $x_w = 5k + 3$ e $y_w = 5l + 3$. La seguente tabella dettaglia

le coordinate di tutti e 8 i segmenti dell'arco mostrato in Figura 4.24, dove rimane da definire il valore di z_{vw} (sempre dispari), di cui parleremo nel seguito.

La seguente tabella definisce formalmente gli estremi degli 8 segmenti di ciascun arco della copertura di cicli blu; le coordinate degli estremi dei segmenti degli archi dei cicli verdi si possono ricavare facilmente.

Figura 4.24.



| Segmento | Punto iniziale | Punto finale |
|----------|----------------------------------|----------------------------------|
| 1 | $(x_v, y_v, 0)$ | $(x_v, y_v + 1, 0)$ |
| 2 | $(x_v, y_v + 1, 0)$ | $(x_v, y_v + 1, z_{vw})$ |
| 3 | $(x_v, y_v + 1, z_{vw})$ | $(x_v, y_v + 2, z_{vw})$ |
| 4 | $(x_v, y_v + 2, z_{vw})$ | $(x_w + 1, y_v + 2, z_{vw})$ |
| 5 | $(x_w + 1, y_v + 2, z_{vw})$ | $(x_w + 1, y_v + 2, z_{vw} + 1)$ |
| 6 | $(x_w + 1, y_v + 2, z_{vw} + 1)$ | $(x_w + 1, y_w, z_{vw} + 1)$ |
| 7 | $(x_w + 1, y_w, z_{vw} + 1)$ | $(x_w, y_w, z_{vw} + 1)$ |
| 8 | $(x_w, y_w, z_{vw} + 1)$ | $(x_w, y_w, 0)$ |

Dimostriamo ora che i percorsi degli archi così definiti non si possono intersecare, dando così luogo ad un disegno ortogonale 3D corretto. Notiamo innanzi tutto che, dato che il disegno è su griglia, gli eventuali incroci che interessano segmenti di lunghezza unitaria possono avvenire solamente negli spigoli, pertanto possiamo trascurare questi segmenti e considerare solo quelli con lunghezza maggiore di 1. Sappiamo, inoltre, che non si possono incrociare tra di loro archi dei cicli di C_{red} perché gli archi interni a C_{red} sono costruiti secondo la Figura 4.23, in cui si vede chiaramente che il percorso degli archi di un ciclo, ricade sempre all'interno dei quadrati centrati sui vertici del ciclo, quindi non è possibile che due cicli abbiano archi che si intersecano dato che non

c'è intersezione tra questi quadrati. Inoltre, non è possibile che archi di C_{red} intersechino archi di C_{green} o C_{blue} in quanto questi svilupperanno la loro costruzione in verticale e, per quanto riguarda il segmento 1, l'unico che giace sul piano di C_{red} , esso esce dal vertice sempre secondo la direzione verticale, lasciata opportunamente libera nella costruzione di C_{red} . Infine, non si possono verificare incroci tra archi di C_{green} con archi di C_{blue} in quanto se una intersezione ci fosse essa dovrebbe essere sul piano $z=0$, ma l'algoritmo posiziona il segmento 1 di ogni arco di C_{blue} verso le ordinate crescenti ("sopra" il vertice), mentre posiziona il segmento 1 di ogni arco di C_{green} verso le ordinate decrescenti ("sotto" il vertice).

Eventuali intersezioni, dunque, si potrebbero verificare solo tra archi di C_{blue} o tra archi di C_{green} . Verifichiamo che anche in questi casi non ve ne sono. Si noti che non si hanno mai intersezioni tra due segmenti di tipo 2 oppure tra due segmenti di tipo 8, poiché ogni punto speciale ha solo un segmento 2 e solo un segmento 8 che fuoriesce dalla propria area. Osserviamo poi che i segmenti 4 si trovano su un piano $y = (5j + 3) + 2$ e i segmenti 6 sul piano $x = (5i + 3) + 1$, per i segmenti 2 avremo $x = (5i + 3)$, $y = (5j + 3) + 1$, mentre per i segmenti 8 $x = (5i + 3)$, $y = (5j + 3)$. Dunque non sono possibili intersezioni tra segmenti 2 con segmenti 4, tra segmenti 2 con segmenti 6, tra segmenti 4 con segmenti 8 e tra segmenti 6 con segmenti 8, in quanto tutte queste coppie giacciono su piani differenti. Per quanto riguarda l'intersezione di segmenti 4 con segmenti 6, abbiamo che essi sono instradati su piani paralleli, uno con coordinata z pari, l'altro dispari, pertanto anche in questi due casi non si avranno intersezioni. L'unico caso rimasto da verificare è che non ci siano intersezioni tra due segmenti 4, o tra due segmenti 6 di archi differenti. Questo equivale ad imporre che due segmenti 2 o due segmenti 8 relativi ad archi che potrebbero causare collisioni non abbiano la stessa altezza. Si noti che i segmenti 4 possono intersecarsi tra loro solo se due punti hanno la stessa ascissa, mentre i segmenti 6 solo se due punti hanno la stessa ordinata.

Allo scopo di imporre questa disuguaglianza di lunghezze, costruiamo il *grafo dei confitti* H , che ha per insieme di vertici gli archi della copertura di cicli C_{blue} . Viene inserito un arco (e, f) tra due nodi del grafo H ogni qual volta e ed f abbiano estremi sulla stessa riga o sulla stessa colonna. H ha massimo grado $2(\sqrt{n} - 1)$, pertanto è possibile trovare una colorazione di vertici con al più $2\sqrt{n} - 1$ colori tramite un semplice algoritmo *greedy*: per ogni vertice v si seleziona il primo colore c diverso da quelli dei suoi adiacenti; banalmente, c non può essere più grande del grado del vertice stesso +1. E' facile vedere che se due vertici hanno colori diversi in H , gli archi corrispondenti possono essere in conflitto nel disegno. Se il colore c , con $1 \leq c \leq 2\sqrt{n} - 1$ è assegnato al vertice di H che corrisponde all'arco (v, w) , impostiamo $z_{vw} = 2c - 1$. In questo modo avremo la garanzia di assegnare valori di z diversi per archi in potenziale conflitto, avendo così la certezza che due segmenti 4 o due segmenti 6 non potranno mai intersecarsi tra loro.

Rimane da calcolare la complessità computazionale dell'algoritmo presentato ed il volume del disegno prodotto.

Teorema. *L'algoritmo ora presentato disegna un grafo di massimo grado 6 in modo ortogonale su griglia 3D con al più 7 svolte per arco in un volume $5\sqrt{n} \times 5\sqrt{n} \times 8\sqrt{n}$ in $O(n\sqrt{n})$ tempo.*

Dimostrazione. Calcoliamo dapprima la complessità. L'algoritmo consiste di 5 passi, evidenziamo la complessità passo per passo, da cui deduciamo la complessità totale:

- | | |
|---------------------------|----------------|
| 1. Fase di preprocessing | $O(n\sqrt{n})$ |
| 2. Disegno di C_{red} | $O(n)$ |
| 3. Disegno di C_{blue} | $O(n\sqrt{n})$ |
| 4. Disegno di C_{green} | $O(n\sqrt{n})$ |

5. Pulitura di G'

$O(n)$

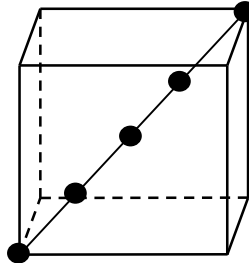
Si osservi che la complessità dei passi 3 e 4 deriva dalla costruzione e colorazione del grafo dei conflitti, che ha un numero di vertici limitato dal numero di archi del grafo G' , ma un numero di archi che può essere $O(n\sqrt{n})$; ne segue che la costruzione e colorazione di H , che si effettuano entrambe in un tempo lineare nella dimensione del grafo, necessitano un tempo $O(n\sqrt{n})$. In totale avremo, quindi una complessità $O(n\sqrt{n})$, cioè $O(n^{3/2})$.

Calcoliamo ora il volume del disegno prodotto. Per quanto riguarda la base, tutti i vertici occupano i punti speciali situati al centro di quadrati 5×5 (vedi Figura 4.23); dato che su ogni lato trovano posto, al più, \sqrt{n} nodi, ogni singolo lato della base avrà lunghezza $5\sqrt{n}$, pertanto avremo un'area di base pari a $5\sqrt{n} \times 5\sqrt{n}$. Per quanto riguarda l'altezza, occorre tener presente come questa venga definita: usiamo la colorazione del grafo dei conflitti che, unitamente alla formula $z = 2c - 1$, stabilisce l'altezza di un dato arco. Per calcolare l'altezza dell'intero disegno, consideriamo il caso peggiore $c = 2\sqrt{n} - 1$. In tal caso $z = 2(2\sqrt{n} - 1) - 1$ quindi $z = 4\sqrt{n} - 5$. Considerando, infine, che questo ragionamento deve essere fatto per il disegno di entrambe le coperture di cicli C_{blue} e C_{green} otteniamo un'altezza complessiva del disegno pari a $2(4\sqrt{n} - 5) = 8\sqrt{n} - 10$. CVD

4.2.1.3. Il secondo algoritmo

Anche l'algoritmo presentato in questa sezione è dovuto a Eades, Symvonis e Whitesides [ESW96]; esso utilizza di nuovo la fase di preprocessing, prende come input un grafo di n vertici e di massimo grado 6, e produce in output un disegno su griglia con al massimo 3 svolte per arco, ma con volume pari a $O(n) \times O(n) \times O(n)$.

Figura 4.25.



L'algoritmo posiziona i vertici di G' (ottenuto dalla fase di preprocessing) sulla diagonale di un cubo di dimensioni $3n \times 3n \times 3n$. Più precisamente, esso numera arbitrariamente i vertici di G' da 1 ad n e posiziona il vertice numerato i nella punto $p_i = (3i, 3i, 3i)$ come illustrato in figura 4.25.

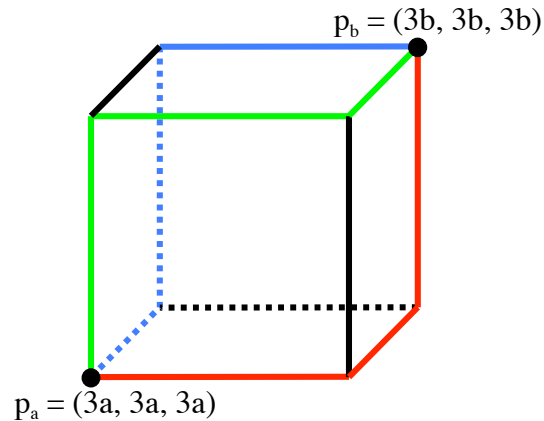
Ogni coppia di nodi a e b in G' determina un cubo $C(a,b)$ con p_a e p_b ai vertici opposti (vedi figura 4.26).

Allo scopo di determinare il percorso del generico arco dal nodo a al nodo b , appartenente alla copertura di cicli rossa, blu o verde, si definiscano i 3 percorsi rosso, blu e verde tra p_a e p_b lungo i lati del cubo $C(a,b)$ come illustrato in figura 4.26. Si osservi che ogni percorso così definito ha solo 2 svolte. L'idea generale alla base dell'algoritmo è quella di instradare l'intero ciclo di copertura rosso (rispettivamente blu o verde) sfruttando il percorso rosso (rispettivamente blu o verde). Nel seguito vedremo che ciò non è possibile in modo così immediato, e quindi alcuni dettagli dovranno essere rifiniti opportunamente. In particolare, introduciamo il concetto di *minimo* o *massimo* in un ciclo.

Definizione. Sia $u_1 \rightarrow u_2 \rightarrow \dots \rightarrow u_k \rightarrow u_1$ un ciclo di G' . Un nodo u_i è un *massimo locale* (rispettivamente, *minimo locale*) rispetto al ciclo se il valore intero ad esso assegnato è più grande (rispettivamente, più piccolo) rispetto ai suoi vicini nel ciclo.

(rispettivamente, più piccolo) del valore assegnato tanto al suo predecessore u_{i-1} quanto al suo successore u_{i+1} .

Figura 4.26.



Si osservi che un vertice può essere un massimo locale rispetto ad un ciclo e un minimo locale rispetto ad un altro ciclo. Per come sono definite le coperture di cicli, su ogni vertice incidono esattamente tre cicli, ciascuno di un differente colore, quindi esso potrà risultare un massimo locale rispetto ad uno o due colori ed un minimo locale rispetto agli altri colori.

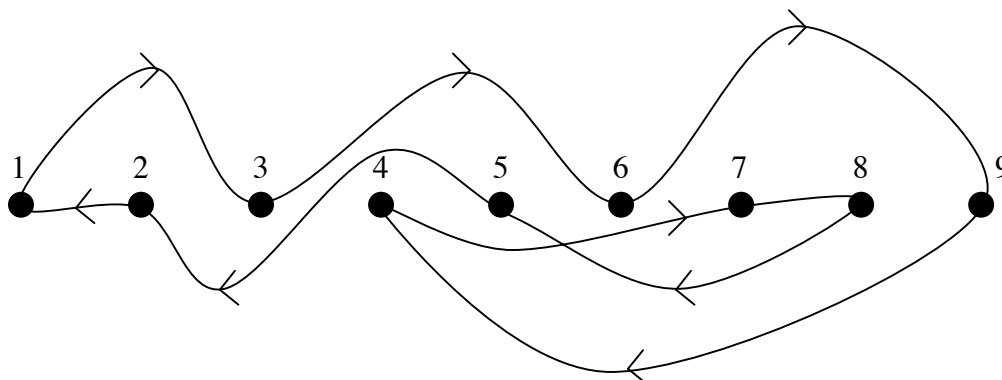
Definizione . Un arco (u_i, u_{i+1}) è detto *crescente* (rispettivamente, *decescente*) se il numero assegnato ad u_i è minore (rispettivamente, maggiore) del numero assegnato ad u_{i+1} .

Definiamo, infine, 4 categorie di archi:

Definizione 5. Un arco (u_i, u_{i+1}) è:

- *crescente normale* se il numero assegnato ad u_i è minore di quello assegnato ad u_{i+1} , che è minore di quello assegnato ad u_{i+2} (archi $(1,3)$, $(3,6)$ e $(4,7)$ in Figura 4.27);
- *decescente normale* se il numero assegnato ad u_i è maggiore di quello assegnato ad u_{i+1} , che è maggiore di quello assegnato ad u_{i+2} (archi $(8,5)$ e $(5,2)$ in Figura 4.27);
- *entrante in un minimo locale* se il numero assegnato ad u_i è maggiore di quello assegnato ad u_{i+1} ma quest'ultimo è minore di quello assegnato ad u_{i+2} (archi $(9,4)$ e $(2,1)$ in Figura 4.27);
- *entrante in un massimo locale* se il numero assegnato ad u_i è minore di quello assegnato ad u_{i+1} ma questo è maggiore di quello assegnato ad u_{i+2} (archi $(6,9)$ e $(7,8)$ in Figura 4.27);

Figura 4.27.

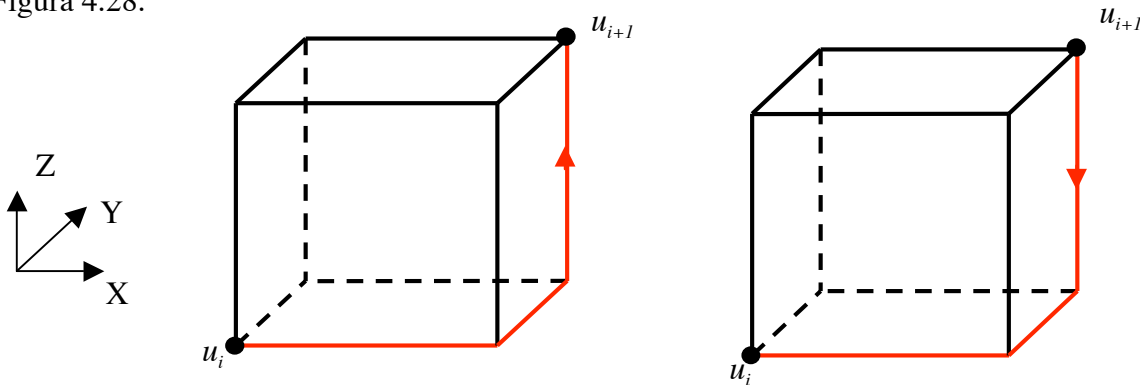


A questo punto, siamo pronti a definire i percorsi che l'algoritmo assegna agli archi delle tre coperture di cicli:

- un arco normale rosso (u_i, u_{i+1}) (crescente o decrescente) è tracciato lungo il percorso rosso degli archi del cubo $C(u_i, u_{i+1})$ con due sole svolte; più precisamente, se è normale crescente esso passa per i punti $(3u_i, 3u_i, 3u_i)$, $(3u_{i+1}, 3u_i, 3u_i)$, $(3u_{i+1}, 3u_{i+1}, 3u_i)$ e $(3u_{i+1}, 3u_{i+1}, 3u_{i+1})$ (vedi figura 4.28 parte sinistra), se è normale decrescente passa per i punti $(3u_i, 3u_i, 3u_i)$, $(3u_i, 3u_i, 3u_{i+1})$, $(3u_i, 3u_{i+1}, 3u_{i+1})$ e $(3u_{i+1}, 3u_{i+1}, 3u_{i+1})$ (vedi figura 4.28 parte destra);
- un arco rosso (u_i, u_{i+1}) entrante in un minimo locale passa per i punti $(3u_i, 3u_i, 3u_i)$, $(3u_i, 3u_i, 3u_{i+1}-1)$, $(3u_i, 3u_{i+1}, 3u_{i+1}-1)$, $(3u_{i+1}, 3u_{i+1}, 3u_{i+1}-1)$ e $(3u_{i+1}, 3u_{i+1}, 3u_{i+1})$ ed ha tre svolte (vedi figura 4.29 parte sinistra);
- un arco rosso (u_i, u_{i+1}) entrante in un massimo locale passa per i punti $(3u_i, 3u_i, 3u_i)$, $(3u_{i+1}+1, 3u_i, 3u_i)$, $(3u_{i+1}+1, 3u_{i+1}, 3u_{i+1})$, $(3u_{i+1}+1, 3u_{i+1}, 3u_{i+1})$ e $(3u_{i+1}, 3u_{i+1}, 3u_{i+1})$ ed ha tre svolte (vedi figura 4.29 parte destra).

Per i cicli verdi e rossi si possono fare discorsi analoghi, usando le porzioni ad essi dedicate.

Figura 4.28.



L'algoritmo, descritto per grandi linee, è dunque il seguente:

ALGORITMO 3BENDS

1. Usa l'algoritmo di preprocessing per calcolare il grafo 6-regolare diretto G' e le sue 3 coperture di cicli C_{red} , C_{green} e C_{blue} ;
2. Numera arbitrariamente i vertici di $V' = V$ da 1 ad n ;
3. FOR $1 \leq i \leq n$, posiziona il nodo u_i nel punto $p_i = (3i, 3i, 3i)$;
4. Disegna i percorsi per ogni arco colorato (u_i, u_{i+1}) di G' secondo le regole descritte in precedenza;
5. FOR ogni arco non diretto di G , disegna il percorso del corrispondente arco diretto colorato di G' .

Figura 4.29.

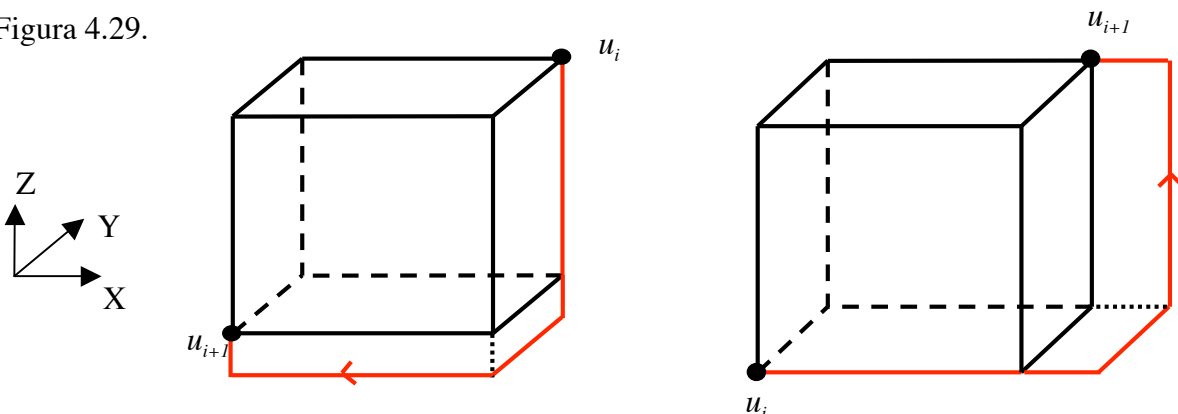
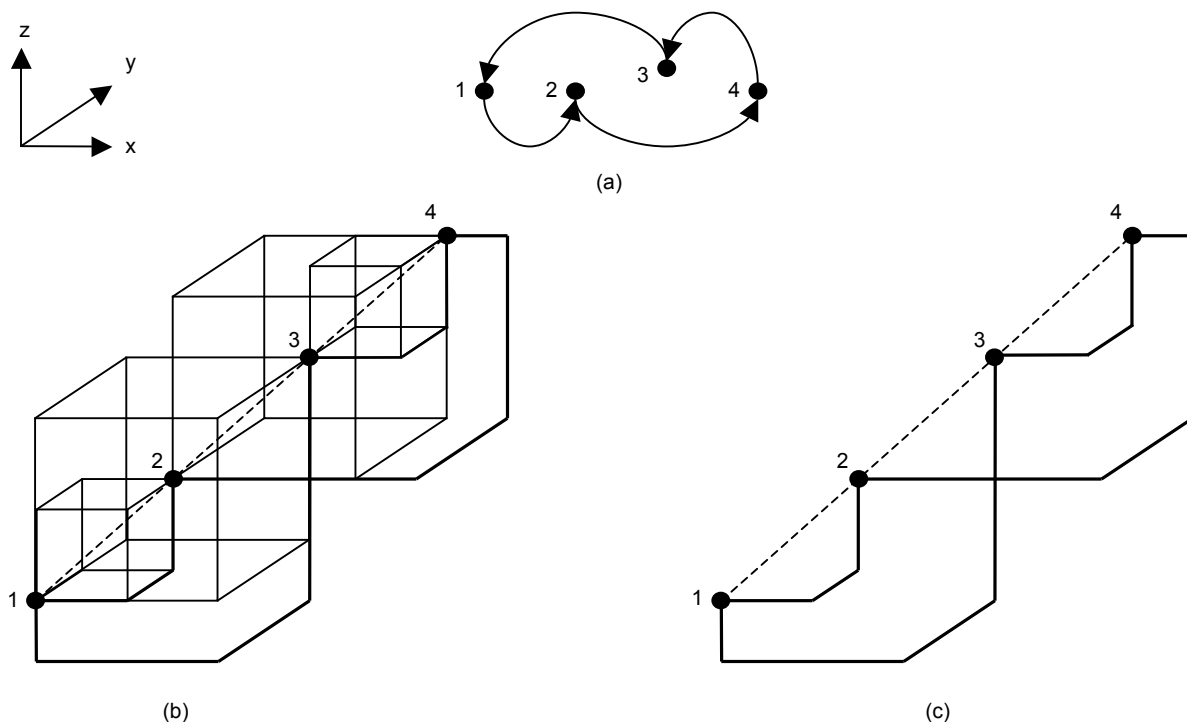


Figura 4.30.



Esempio. In figura 4.30 è mostrato il disegno di un ciclo semplice secondo le regole ora descritte. L'arco (1,2) è normale crescente, l'arco (2,4) entra in un massimo locale, l'arco (4,3) è normale decrescente, e l'arco (3,1) è entrante in un minimo locale.

Teorema . Per ogni grafo G di massimo grado 6, esiste un algoritmo con complessità temporale $O(n^{3/2})$ che produce un disegno ortogonale su griglia 3D avente:

- al massimo 3 svolte per arco;
- massima lunghezza di un arco pari a $9(n-1)+2$;
- un volume di dimensioni al più $(3n-1) \times (3n-1) \times (3n-1)$.

Dimostrazione. Dapprima si osservi che il disegno prodotto è banalmente un disegno ortogonale su griglia, e che non sono presenti incroci, per come sono stati tracciati gli archi, che occupano porzioni disgiunte di volume.

Dimostriamo ora una per una tutte le affermazioni.

Nella descrizione dei percorsi degli archi si è detto che esistono 4 tipi di arco, tracciati in modo tale da avere 2 o 3 svolte, e quindi il disegno ha al massimo 3 svolte per arco.

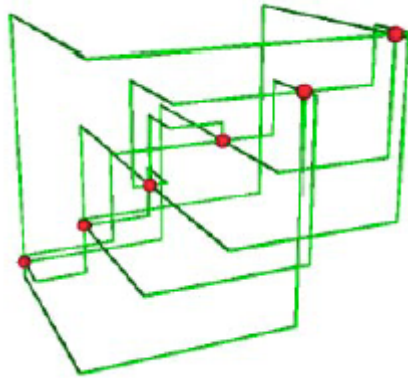
L'arco più lungo è quello che connette i 2 vertici posizionati agli estremi della diagonale del cubo p_1 e p_n . Tale arco percorre $n-1$ unità lungo l'asse X moltiplicato 3 (ovvero la differenza di ascissa tra i due nodi) + $n-1$ unità sull'asse Y moltiplicato 3 (ovvero la differenza di ordinata tra i due nodi) + $n-1$ unità sull'asse Z moltiplicato 3 (ovvero la differenza di quota tra i due nodi). Aggiungiamo ancora due unità perché l'arco potrebbe essere entrante in un massimo locale o in un minimo locale ed otteniamo $9(n-1)+1$.

Il cubo $C(1,n)$ avente p_1 e p_n agli estremi della diagonale ha volume $(3n-3) \times (3n-3) \times (3n-3)$. Anche in questo caso, per ciascun colore di cicli, dobbiamo incrementare due delle tre dimensioni a causa della possibile esistenza di archi entranti in un minimo o un massimo locale. Ne consegue un volume di $(3n-1) \times (3n-1) \times (3n-1)$.

Per quanto riguarda, infine, la complessità computazionale, si osservi che la parte più onerosa dell'algoritmo è la fase di preprocessing, ne consegue un $O(n^{3/2})$ tempo. **CVD**

In Figura 4.30 è mostrato un esempio di output prodotto dall'algoritmo.

Figura 4.30.



4.2.2. Disegno rettilineo 3D

Riferimenti: [Cal94] Solo sezioni 1 e 2, esclusa dimostrazione del teorema 1; [CS97] tutto, escluso Th. 4.

Nel *disegno rettilineo 3D*, i nodi sono posizionati sui punti della griglia (a coordinate intere) e gli archi sono segmenti rettilinei, tra cui non sono permessi incroci. Questo tipo di disegno è l'estensione del disegno rettilineo 2D, ed è stato introdotto solo nel '94 da Cohen, Eades, Lin e Ruskey [Cal94].

Si osservi che il teorema di Steinitz assicura l'esistenza di un disegno bidimensionale rettilineo senza incroci per tutti i grafi planari 3-connessi. Spostandosi sulle 3 dimensioni, tale rappresentazione è sempre possibile per qualunque grafo. L'obiettivo è quindi quello di minimizzare il volume.

Purtroppo, questa visualizzazione è molto difficile da leggere, perché tutti gli archi sono a due a due sghembi ma in ogni proiezione bidimensionale ci sono moltissimi incroci (vedi Fig. 4.31). Pertanto, questa convenzione, ancor più della ortogonale, ha un interesse esclusivamente teorico.

Cohen, Eades, Lin e Ruskey [Cal94] dimostrano che un limite inferiore per il volume di K_n è $\Omega(n \times n \times n)$ e poi danno un algoritmo per disegnare qualunque grafo in volume $O(n^3)$, più precisamente $n \times 2n \times 2n$. Nel seguito dettagliamo alcuni di questi risultati.

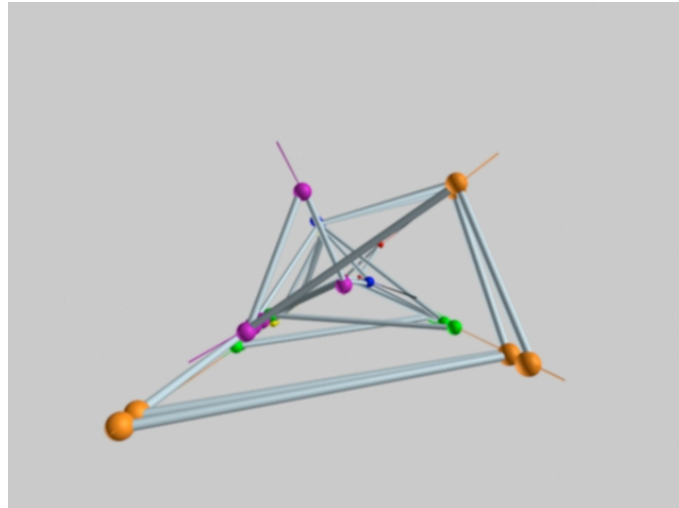
4.2.2.1. Limitazione inferiore per grafi generali

Teorema. K_5 e $K_{3,3}$ sono non planari.

Dimostrazione. Se K_5 fosse planare, dovrebbe valere la formula di Eulero, ed il conseguente risultato $3f \leq 2m$, da cui $m \leq 3n-6$, mentre K_5 ha 5 nodi e 10 archi.

Per dimostrare che $K_{3,3}$ non è planare, si osservi che – essendo $K_{3,3}$ un grafo bipartito – non ha cicli dispari, e quindi ogni faccia è limitata da almeno 4 archi, cioè $4f \leq 2m$. Sostituendo in questa disuguaglianza e nella formula di Eulero i valori $n=6$ ed $m=9$ si ottiene una contraddizione. **CVD**

Figura 4.31.



Teorema. Sia Γ un disegno rettilineo 3D di K_n , e Γ utilizzi volume $X \times Y \times Z$. Allora ciascuna dimensione è $\Omega(n)$.

Dimostrazione. Consideriamo la dimensione X e distinguiamo 2 casi:

- $X \leq n/5$. Considera i piani paralleli al piano yz che intersecano Γ . Deve esistere un tale piano che contenga almeno 5 nodi. Essi inducono un K_5 , che non è planare; ne scaturisce un assurdo.
- $X > n/5$. Questo è possibile, ma allora X è un $\Omega(n)$.

Lo stesso ragionamento può essere replicato per le altre 2 dimensioni. Segue che ogni dimensione deve essere almeno $n/5$. **CVD**

4.2.2.2. Limitazione superiore per grafi generali

Diamo solo un'idea, senza entrare nel dettaglio. Si determinano dei punti, detti *punti universali*, che sono punti su cui si possono posizionare i nodi di qualunque grafo ed hanno la proprietà di essere congiunti a due a due da rette sghembe.

L'idea è di usare la cubica gobba di equazioni parametriche t, t^2, t^3 , che è una curva sghemba, nel senso che corde distinte si intersecano solo ai loro estremi, oppure, in altri termini, non esistono su di essa 4 punti complanari. Tuttavia, se si scelgono come punti universali tutti i punti a coordinate intere della cubica gobba si ottiene un volume di $n \times n^2 \times n^3$.

In [Cal96], gli autori trovano il modo per 'ripiegare' questa cubica restringendo il volume necessario utilizzando le classi resto $\text{mod } p$, dove p è un numero primo con $n < p < 2n$, per cui i punti universali sono quelli a coordinate $t, t^2 \text{ mod } p, t^3 \text{ mod } p$. Essi poi dimostrano che i punti così definiti sono comunque tutti a quattro a quattro non complanari tra loro.

Nello stesso lavoro, Cohen, Eades, Lin e Ruskey dimostrano che alcune classi di grafi (alberi e grafi planari) hanno bisogno di un volume inferiore ($O(n \log n)$ per gli alberi ed $O(n^2)$ per i planari). Essi congetturano poi l'esistenza di altre classi di grafi per cui sia necessario un volume inferiore a $O(n^3)$.

Gli autori di [CS97] determinano delle altre classi che richiedono volume minore (bipartiti, 3-partiti e 4-partiti), e Pach e Toth [PT98] dimostrano che tale volume è ottimo, generalizzando il risultato a grafi k -partiti, per k costante.

4.2.2.3. Limitazione inferiore per grafi bipartiti

Sia $G = (V, E)$ un grafo; un sottoinsieme di nodi A si dice *indipendente* se il sottografo indotto da A non ha archi, ovvero se $\forall x, y \in A$ si ha $\{x, y\} \notin E$.

Un grafo $G = (V, E)$ si dice *bipartito* se esiste una partizione dell'insieme dei nodi $V = V_1 \cup V_2$, tale che $V_1 \cap V_2 = \emptyset$, e V_1 e V_2 sono insiemi non vuoti e indipendenti.

Un grafo $G = (V, E)$ si dice *k-partito* se esiste una partizione dell'insieme dei nodi $V = V_1 \cup V_2 \cup \dots \cup V_k$, tale che $\forall i, j$, con $i, j = 1, 2, \dots, k$ e $i \neq j$, $V_i \cap V_j = \emptyset$, e gli insiemi V_1, V_2, \dots, V_k sono non vuoti e indipendenti.

Siano V_1 e V_2 due insiemi finiti non vuoti e disgiunti; il grafo *completo bipartito* su V_1 e V_2 è il grafo bipartito $G = (V_1, V_2, E)$, dove $E = \{ \{x, y\} \mid x \in V_1, y \in V_2 \}$. In altre parole, in un grafo completo bipartito esistono tutti i possibili archi tra ciascun nodo di V_1 e ciascun nodo di V_2 .

Siano n e m due interi positivi; il grafo *completo bipartito di ordine (n, m)* , è il grafo completo bipartito $K_{n,m} = (\{1, 2, \dots, n\}, \{n+1, n+2, \dots, n+m\}, E)$.

Nel seguito chiameremo i nodi delle due partizioni nodi *rossi* e nodi *blu*, e perciò le due partizioni saranno denominate con R e B anziché con V_1 e V_2 . Le quantità r e b sono le cardinalità degli insiemi R e B , ed $r + b = n$.

Teorema. Sia Γ un disegno rettilineo 3D di $K_{r,b}$, con $r, b \geq 3$. Allora il volume di Γ è un $\Omega(n^{3/2})$.

Dimostrazione. Si proiettino tutti i nodi rossi sul piano yz . Due casi sono possibili:

- tutti i nodi hanno proiezioni distinte, allora l'area della sezione perpendicolare all'asse x è almeno r ;
- qualche coppia di punti va nello stesso punto del piano yz . Allora, non può esistere una coppia di punti blu che hanno la stessa proiezione sul piano yz , altrimenti i quattro punti formerebbero un $K_{2,2}$ sullo stesso piano determinato da due rette parallele, e quindi un incrocio. Segue che l'area della sezione perpendicolare all'asse x è almeno b .

Questo ragionamento può essere ripetuto per tutte e tre le direzioni. È facile vedere che il minor volume si ottiene quando $r=b=n/2$ e, in tal caso, $V = \Omega(n^{3/2})$. **CVD**

Corollario. Sia Γ un disegno rettilineo 3D di un grafo completo *k-partito* in cui almeno due delle *k* partizioni contengono ≥ 3 nodi. Allora il volume di Γ è un $\Omega(n^{3/2})$.

Dimostrazione. L'asserto segue dal fatto che ogni grafo *k-partito* contiene un sottografo bipartito, e quindi la stessa limitazione inferiore vale per tutta la classe di grafi *k-partiti*. **CVD**

4.2.2.4. Limitazione superiore per grafi bipartiti

Due rette dello spazio si dicono *sghembe* se non giacciono sullo stesso piano. In altre parole, date le rette dello spazio a e b , siano A e B i fasci di piani centrati rispettivamente in a e b , allora a e b sono sghembe se e solo se $A \cap B = \emptyset$, cioè se non esiste un piano appartenente contemporaneamente sia ad A che a B .

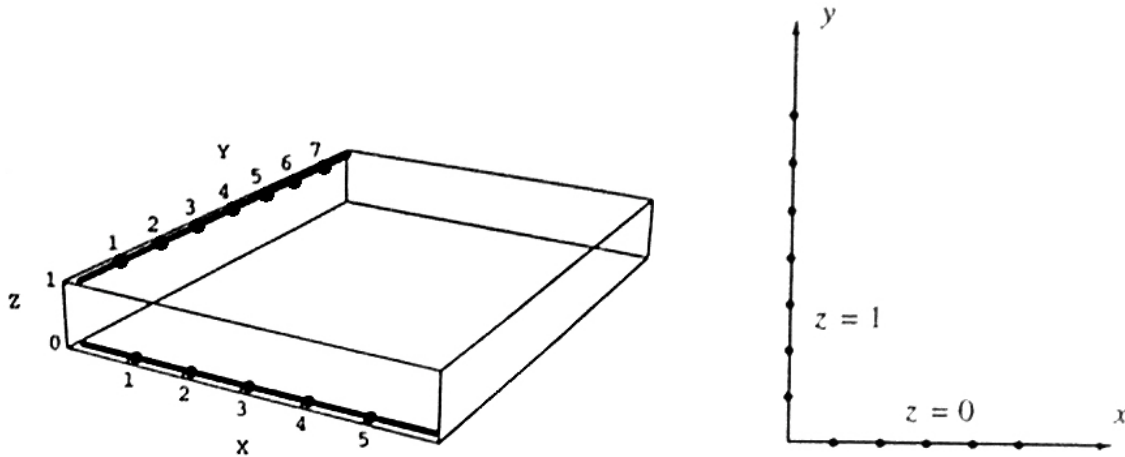
In questa sezione mostreremo come sia possibile costruire un disegno rettilineo 3D su griglia di un grafo bipartito completo in volume $O(n^2)$. Inoltre mostriamo come questa tecnica, basata sulla sistemazione di ogni colore su una di alcune rette sghembe possa essere estesa al disegno di grafi 3-partiti in volume $O(n^2)$.

Teorema. È possibile disegnare un grafo completo bipartito in modo rettilineo 3D su griglia in un volume $O(n^2)$.

Dimostrazione. È sufficiente collocare i nodi su due rette sghembe come mostrato in figura 4.32. Poiché le rette sono sghembe, quattro nodi $r_1, r_2 \in R$ e $b_1, b_2 \in B$ non possono giacere sullo

stesso piano. E' facile vedere che un volume $O(n^2)$ è sufficiente a contenere il disegno, infatti le 3 dimensioni del parallelepipedo ad esso circoscritto sono r, b ed un valore costante. **CVD**

Figura 4.32.



Corollario. E' possibile disegnare un qualunque grafo bipartito in modo rettilineo 3D su griglia in un volume $O(n^2)$.

Dimostrazione. Basta osservare che un grafo completo bipartito contiene come sottografo un qualunque bipartito che abbia la stessa partizione dei nodi. Pertanto un volume $O(n^2)$ è sufficiente a qualsiasi bipartito di n nodi. **CVD**

Teorema. E' possibile disegnare un grafo completo 3-partito in modo rettilineo 3D su griglia in un volume $O(n^2)$.

Dimostrazione. Mostriamo due modi per costruire un disegno rettilineo 3D su griglia con volume $O(n^2)$ di un grafo completo 3-partito.

Sia $K_{r,b,g} = (R \cup B \cup G, E)$ un grafo completo 3-partito, dove r, b, g sono le cardinalità rispettivamente degli insiemi R, B, G , ed $r + b + g = n$.

$K_{r,b,g}$ può essere disegnato in volume $O(n^2)$ usando una delle due seguenti tecniche, entrambe consistenti nel collocare i nodi lungo tre rette sghembe.

PRIMO MODO. Le tre rette su cui giacciono i nodi, come mostrato in figura 4.33, hanno equazioni parametriche $(x = p, y = 0, z = 0)$, $(x = -q, y = q, z = 1)$ e $(x = s, y = s, z = 2)$. I nodi sono collocati a coordinate $R_i = (i, 0, 0)$, $i = 1, \dots, r$, $B_j = (-j, j, 1)$, $j = 1, \dots, b$ e $G_k = (-k, -k, 2)$, $k = 1, \dots, g$. Si osservi che ai tre sottoinsiemi di archi che connettono tutte le possibili coppie di colori (cioè i tre sottoinsiemi di archi che connettono nodi *blu* e *rossi*, *rossi* e *gialli*, *gialli* e *blu*) sono assegnati porzioni differenti di spazio ("proprietà di separazione"). Si osservi che nessun incrocio di archi può sorgere grazie a tale proprietà.

Il volume di tale disegno è $(b+r) \times (b+g) \times 2$ che, nel caso in cui le tre partizioni siano bilanciate (cioè $b=r=g=n/3$) è pari a $2/3 n \times 2/3 n \times 2 = 8/9 n^2$. In ogni caso, il volume è un $O(n^2)$.

Nel tentativo di rendere il volume inferiore, possiamo definire un secondo metodo di costruzione.

SECONDO MODO. Le tre rette su cui giacciono i nodi, come mostrato in figura 4.34, hanno equazioni parametriche $(x = p, y = 0, z = 0)$, $(x = 0, y = s, z = 2)$, $(x = q, y = 2g - q, z = 1)$. Si assuma, senza perdita di generalità, che su esse giacciono rispettivamente i nodi di G, R, B e $g \geq r$ (se invece fosse stato $g < r$ il valore della coordinata y della terza equazione parametrica sarebbe stata $y = 2r - q$). In questa seconda configurazione tutti gli archi giacciono nella stessa porzione di spazio. In questa costruzione, alcuni incroci possono verificarsi. Per rendersene conto è sufficiente

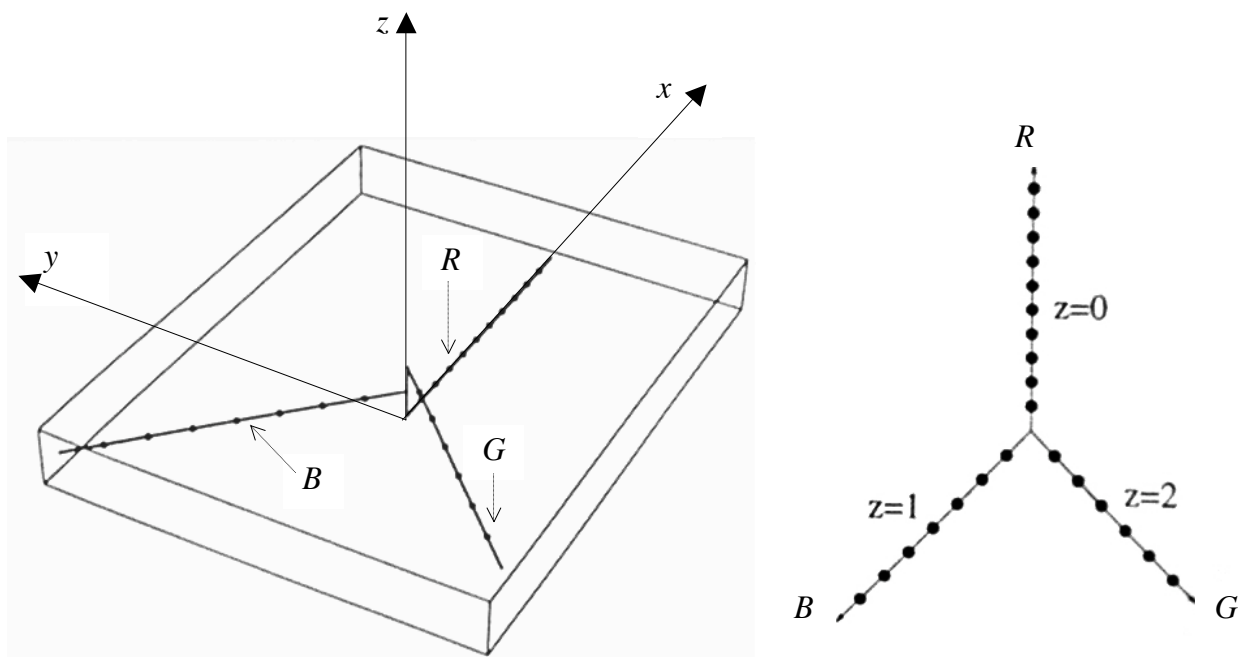
proiettare il grafo sul piano yz (o equivalentemente sul piano xz) come mostrato in figura 4.35. L'asse x nella proiezione è un punto e di conseguenza i vertici *verdi* (che per ipotesi giacciono sull'asse x) hanno tutti la stessa proiezione in questo punto. Il fascio di piani centrati in x è rappresentato nella proiezione dall'insieme di rette incidenti nel punto proiezione di x . Nella figura si vede che alcune di queste rette intersecano le rette su cui giacciono i nodi *blu* ed i nodi *rossi*; in particolare, se intersecano un nodo *blu*, intersecano anche un nodo *rosso* e questo vuol dire che su uno stesso piano contenente TUTTI i nodi *gialli* giacciono anche un nodo *rosso* ED uno *blu*, dando luogo ad un incrocio. Per ovviare a questo inconveniente, è necessario non disporre i nodi appartenenti alla retta R in ogni incrocio della griglia (coordinata intera) ma “vietare” alcune posizioni. I piani contenenti la retta G e passanti attraverso un punto della retta B incontrano anche la retta R sempre a coordinata y di R doppia rispetto a quella di B . Allora bisogna posizionare i nodi *rossi* solo su quegli incroci della griglia dove R ha coordinata y dispari. Un ragionamento simile può essere fatto per i nodi *gialli* considerando tutti i piani contenenti la retta R . Nessun piano contenente la retta B può invece intersecare contemporaneamente sia R che G , e quindi non è necessario vietare alcuna posizione su B .

Il volume di questo disegno è $r \times g \times 2$, e quindi sempre inferiore al volume del disegno scaturito dal disegno precedente, ma pur sempre dell'ordine di n^2 .

In entrambe le costruzioni, la massima lunghezza di ogni linea è $O(n)$.

CVD

Figura 4.33.



Si osservi che queste rappresentazioni hanno altezza costante, quindi pessima aspect ratio; è un problema aperto capire se si possa ottenere lo stesso volume con le 3 dimensioni comparabili.

E' possibile rappresentare anche i 4-partiti posizionando i nodi di ciascun colore (partizione) su una di quattro rette sghembe in modo simile alla seconda costruzione vista sopra per i 3-partiti, ma è necessario vietare più posizione che al passo precedente, perché le combinazioni di incroci aumentano.

Aumentando il numero delle partizioni le cose si complicano, infatti non si riesce ad estendere questa tecnica ai 5-partiti: in una retta si dovrebbero vietare troppe posizioni per impedire incroci,

Figura 4.34.

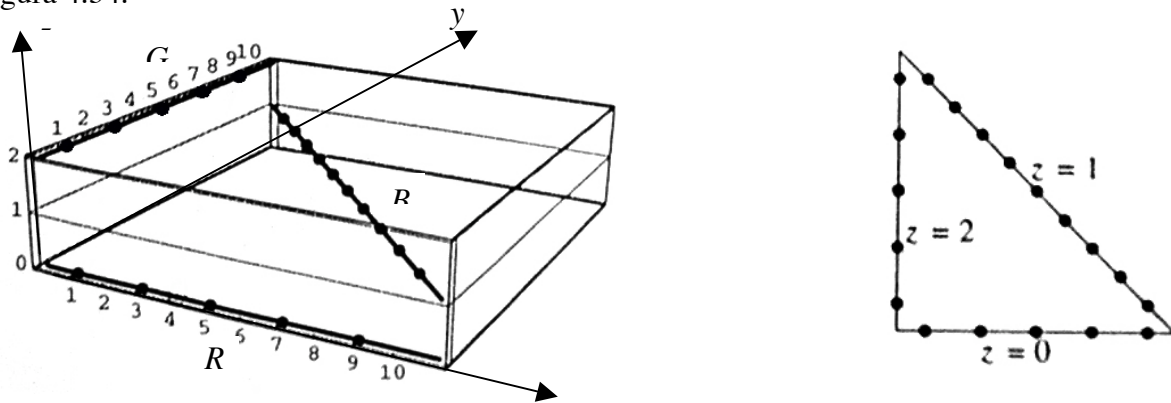
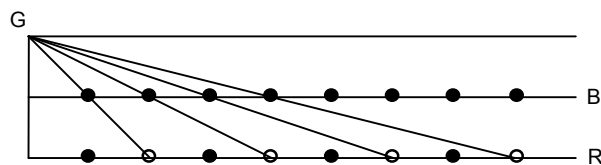


Figura 4.35.



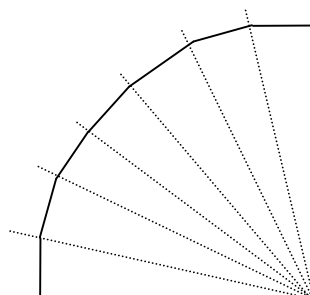
ma più aumenta il numero di posizioni da vietare su una retta e più lunga questa dovrà essere per contenere tutti i nodi, di conseguenza il volume non è più limitato dal quadrato del numero dei nodi; un altro problema legato a questa tecnica consiste nel fatto che le rette usate devono avere tutti coefficienti angolari diversi tra loro, ed aumentando il numero delle rette si devono cominciare ad usare coefficienti angolari le cui rette corrispondenti ammettono sempre meno punti a coordinate intere; anche questo contribuisce ad aumentare la dimensione del disegno.

Nello stesso lavoro, gli autori congetturano che la limitazione inferiore possa essere migliorata ad $\Omega(n^2)$ e che, anche per $k > 4$, il volume di un grafo k -partito sia $O(n^2)$. Questa congettura viene dimostrata da Pach, Thiele e Toth [PTT98] tramite il seguente teorema:

Teorema. *Per qualunque costante $k \geq 2$, ogni grafo k -partito ammette un disegno rettilineo su griglia 3D il cui volume è $O(k^2 n^2)$. L'ordine di grandezza di questo limite non può essere migliorato.*

Idea di dimostrazione. Per visualizzare in modo 3D rettilineo grafi k -colorabili con $k > 4$ è possibile usare la seguente tecnica: si utilizzano k piani tali che le loro proiezioni sul piano xy formino un k -latero simile a quello mostrato in figura 4.36. Il volume risultante è in funzione di n^2 ma anche di k . Ne consegue che questa tecnica ha senso solo se k è una costante.

Figura 4.36.



Si osservi che molte famiglie di grafi ricadono all'interno dei grafi k -partiti, per k costante, ad esempio grafi planari, grafi di genere limitato, e grafi di treewidth limitata. Dal risultato precedente, ne segue che tutte queste famiglie di grafi ammettono un disegno rettilineo in volume quadratico. Questo solleva naturalmente la questione di capire quali siano le famiglie di grafi che ammettono un volume subquadratico, o addirittura lineare. Ovviamente, il volume lineare è il meglio che si possa sperare, visto che un grafo con n nodi non può essere rappresentato in volume inferiore ad $O(n)$.

In questa direzione va il seguente teorema [BCMW04]:

Teorema. *Ogni disegno rettilineo di un grafo con n nodi ed m archi ha un volume di almeno $1/8(m+n)$.*

Dimostrazione. Per provare l'asserto, dobbiamo dimostrare la seguente affermazione: *il massimo numero di archi di un grafo in un suo disegno rettilineo di volume $X \times Y \times Z$ è esattamente $(2X-1)(2Y-1)(2Z-1)-XYZ$.*

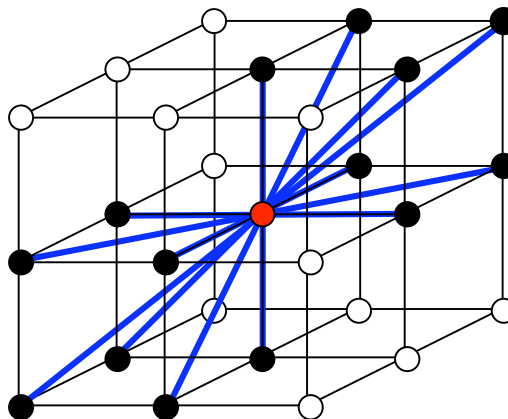
Sia B la scatola che racchiude il disegno rettilineo di un grafo con n nodi ed m archi, di volume $X \times Y \times Z$. Consideriamo l'insieme $P = \{(x,y,z) \in B : 2x, 2y, 2z \text{ interi}\}$, cioè l'insieme di punti nella scatola le cui coordinate siano o intere o del tipo $a/2$, per qualche a . Ovviamente, $|P| = (2X-1)(2Y-1)(2Z-1)$. Si osservi che il punto medio di ciascun arco del grafo deve appartenere a P , perché i nodi sono sempre posizionati a coordinate intere; inoltre, poiché gli archi non si incrociano, i punti medi degli archi sono tutti distinti. Ne segue che $m \leq |P|$. Inoltre, poiché i punti medi degli archi non intersecano mai i nodi, si ha $m \leq |P| - n$.

Un disegno col massimo numero di archi non può avere alcun arco che passa per un punto della griglia, altrimenti si potrebbe inserire un nuovo nodo in corrispondenza di quel punto ed aumentare il numero degli archi. Perciò si deve necessariamente avere che $n = XYZ$ ed $m \leq |P| - n$.

Dimostriamo la disuguaglianza opposta tramite la seguente costruzione: si associ ad ogni punto di B un nodo del grafo ed ogni nodo a coordinate (x,y,z) sia adiacente a tutti i nodi a coordinate $(x+1,y,z), (x-1,y,z), (x,y+1,z), (x,y-1,z), (x,y,z+1), (x,y,z-1), (x+1,y+1,z), (x-1,y-1,z), (x+1,y,z+1), (x-1,y,z-1), (x,y+1,z+1), (x,y-1,z-1), (x+1,y+1,z+1), (x-1,y-1,z-1)$ – vedi figura 4.37. E' facile vedere che non vi sono archi che intersecano (se non nei loro estremi) ed inoltre ogni punto dell'insieme P è o un nodo o il punto medio di un arco; perciò il numero di archi è $m = |P| - n$.

Per concludere la dimostrazione, sia ora $V = X \times Y \times Z$ il volume necessario per rappresentare un grafo con n nodi ed m archi. Sappiamo che $m = |P| - n < 8V - n$, da cui $V > (m+n)/8$. **CVD**

Figura 4.37.



Per concludere, osserviamo che, come conseguenza, questo teorema comporta il fatto che grafi che hanno $m = O(n)$ possono ammettere un disegno rettilineo con volume lineare; i grafi planari sono

dei buoni candidati, in questo senso: essi sono grafi 4-partiti, e quindi ammettono un disegno rettilineo con volume $O(n^2)$, ma la limitazione inferiore per il volume per questa classe di grafi è $O(n)$. Un importante problema, tuttora aperto, è quello di stabilire se i grafi planari ammettano un disegno rettilineo di volume lineare. In questa direzione è stato fatto un piccolo passo avanti, dimostrando che i grafi planari ammettono un disegno rettilineo in volume $O(n^{3/2})$ [DW04b].

si può aggiungere qui il track drawing

RIFERIMENTI BIBLIOGRAFICI

- [BCMWO4] P. Bose, J. Czyzowicz, P. Morin, D.R. Wood: "The Maximum Number of Edges in a Three-Dimensional Grid-Drawing", *Journal of Graph Algorithms and Applications*, **8**(1), pp. 21-26, 2004.
- [CS97] T. Calamoneri, A. Sterbini: 3D straight-line drawing of 4-colorable graphs, *Information Proc. Letters* **63**, pp. 97-102, 1997.
- [Cal94] Cohen, Eades, Lin, Ruskey: Three-Dimensional Graph Drawing, Proc. *Graph Drawing '94, LNCS 894*, pp. 1-11, 1994.
- [DW04] V. Dujmovic, S. Whitesides: "Three-dimensional drawings", in ???, 2004.
- [DW04b] V. Dujmovic, D.R. Wood: "Three-dimensional grid drawings with sub-quadratic volume", in J. Pach editor, *Towards a Theory of Geometric Graphs*, vol. 342 of *Contemporary Mathematics*, pp. 55-66, 2004.
- [ESW96] P. Eades, A. Symvonis, S. Whitesides: "Two Algorithms for Three Dimensional Orthogonal Graph Drawing", Proc. *Graph Drawing (GD'96) LNCS 1190*, pp. 139-154, 1996. Anche come: "Three Dimensional orthogonal graph Drawing algorithms", *Discrete Applied Mathematics*, **103**, pp. 55-87, 2000.
- [KB67] A.N. Kolmogorov, Y.M. Bardzin: About realization of sets in 3-dimensional space, *Problems in Cybernetics*, 261-268, 1967.
- [LW92] J. H. van Lint, R. M. Wilson: *A Course in Combinatorics*, Cambridge University Press, 1992.
- [PTT98] J. Pach, T. Thiele, G. Toth: Three-Dimensional Grid Drawings of Graphs. Proc. *6th Int.l Symp. On Graph Drawing (GD '97)*, Lecture Notes in Computer Science 1353, 1998. Anche in B. Chazelle, J.E. Goodman, R. Pollack editors, *Advances in discrete and computational geometry*, 223, 251-255, 1999.
- [W97] R. Webber: Finding the best viewpoints for three-dimensional graph drawings. Proc. *5th Int.l Symp. On Graph Drawing (GD '97)*. Lecture Notes in Computer Science 1353, 87-98, 1997.
- [W98] D. Wood: An algorithm for three-dimensional orthogonal graph drawing. Proc. *6th Int.l Symp. On Graph Drawing (GD '98)*, Lecture Notes in Computer Science 1547, 332-346, 1998.
- [W98b] D. Wood: Two-bend three-dimensional orthogonal grid drawing of maximum degree five graphs. *Tech. Rep. 98/03 Monash Univ.*, 1998.
- [W03] D.R. Wood: "Lower Bounds for the Number of bends in Three-Dimensional Orthogonal graph Drawings", *Journal of Graph Algorithms and Applications*, **7**(1), pp. 33-77, 2003.