

Capitolo 10

Assegnamento e accoppiamento

Si definiscono come problemi di ottimizzazione combinatoria quei problemi che sono originati da strutture discrete e che hanno come soluzioni ammissibili tutte le configurazioni della struttura che siano ammissibili in base ad un prefissato criterio. Normalmente il numero di soluzioni ammissibili cresce in modo combinatorio, quindi esponenziale, rispetto alle dimensioni dell'istanza ed è indispensabile uno studio delle proprietà strutturali del problema per poter trovare l'ottimo. Nei capitoli precedenti si sono già analizzati alcuni problemi che rientrano naturalmente nell'ottimizzazione combinatoria, come i problemi del cammino minimo, del taglio di minima capacità e dell'assegnamento. In questi problemi una fortunata circostanza, cioè la totale unimodularità della matrice dei vincoli, permette di affrontarli come se fossero normali problemi continui di flusso o, più in generale, di programmazione lineare. Per gli altri problemi invece sono richieste analisi più approfondite per pervenire ad algoritmi efficienti o semplicemente accettabili.

Iniziamo questo studio da quei problemi (pochi purtroppo) per i quali esistono algoritmi polinomiali. In questa categoria rientrano ad esempio, oltre ai problemi già visti, i problemi d'accoppiamento e i problemi definiti su matroidi. Data l'efficienza dei loro algoritmi risolutori, è naturale che questi problemi vengano invocati spesso nella costruzione di algoritmi complessi per problemi più difficili. Quindi una loro trattazione abbastanza approfondita è giustificata da questo fatto e anche dall'osservazione che alcuni metodi di analisi si possono poi applicare ed estendere ad altre classi di problemi.

In questo capitolo viene studiato il problema dell'accoppiamento sia nel caso generale sia in quello particolare su grafi bipartiti, che corrisponde al problema dell'assegnamento. Questo secondo problema verrà affrontato con tecniche diverse da quelle già viste. Il problema dell'assegnamento fu studiato da Kuhn [1955], il quale elaborò un particolare algoritmo polinomiale, denominato 'algoritmo ungherese', che sfrutta la totale unimodularità della matrice dei vincoli. Questa proprietà non è valida per il problema dell'accoppiamento su un grafo generico, per cui lo sviluppo di un algoritmo polinomiale per il problema dell'accoppiamento fu più difficile e giunse molto più tardi con il fondamentale lavoro di Edmonds [1965b].

All'interno dei problemi polinomiali di ottimizzazione combinatoria, il problema dell'accoppiamento è certamente uno dei più complessi e la sua analisi richiede quindi un intero capitolo. Ripetiamo la definizione già fornita nel secondo capitolo (vedi definizione 2.7).

10.1 DEFINIZIONE. *Dato un grafo (non orientato) $G = (N, E)$, un sottoinsieme M di archi, tale che ogni nodo del grafo è incidente in al più un arco di M , viene detto accoppiamento, (matching). I nodi incidenti in M vengono detti nodi accoppiati, mentre gli altri nodi vengono detti esposti.* ■

10.2 DEFINIZIONE. *Se il grafo è bipartito, un accoppiamento viene anche detto assegnamento.* ■

Il motivo per cui l'accoppiamento viene detto anche assegnamento nel caso bipartito è ovviamente dovuto al fatto che un accoppiamento realizza una corrispondenza biunivoca fra due sottoinsiemi di due insiemi dati.

10.3 DEFINIZIONE. *Un accoppiamento in cui tutti i nodi sono accoppiati viene detto perfetto.* ■

10.4 DEFINIZIONE. *Il problema dell'accoppiamento di cardinalità consiste nel trovare un accoppiamento di massima cardinalità.* ■

Se per ogni arco è assegnato un costo c_e , $e \in E$, e se il peso di un insieme M di archi viene semplicemente definito come $c(M) := \sum_{e \in M} c_e$, allora si può definire:

10.5 DEFINIZIONE. *Dato un grafo completo, il problema dell'accoppiamento pesato consiste nel trovare un accoppiamento perfetto di costo minimo.* ■

10.6 DEFINIZIONE. *Dato un grafo bipartito completo, il problema dell'assegnamento pesato consiste nel trovare un accoppiamento perfetto di costo minimo.* ■

Ovviamente nei problemi pesati il grafo deve avere un numero pari di nodi e nel caso bipartito i due insiemi di nodi devono avere la stessa cardinalità.

Anche se i problemi di accoppiamento bipartito sono un caso particolare di quelli definiti su un grafo generale, è tuttavia utile distinguere l'accoppiamento bipartito da quello non bipartito. Non solo un accoppiamento bipartito corrisponde ad un assegnamento e quindi riveste una speciale importanza in alcuni problemi di ricerca operativa, ma avviene anche che la particolare struttura semplifichi in modo essenziale il problema.

10.1. Cammini aumentanti

Nello sviluppo di algoritmi per problemi di accoppiamento uno dei concetti chiave è quello di cammino aumentante. Sia $G = (N, E)$ un grafo assegnato (indifferentemente se completo o non, bipartito o generico) e sia $M \subset E$ un accoppiamento assegnato.

10.7 DEFINIZIONE.

- *Un cammino semplice che parte da un nodo esposto e consiste alternativamente di archi accoppiati e archi non accoppiati viene detto cammino alternante.*
- *I nodi di un cammino alternante che distano dal nodo esposto un numero pari di archi vengono detti nodi esterni, mentre quelli distanti un numero dispari di archi nodi interni.*
- *Un albero con radice in un nodo esposto e in cui tutti i cammini dalla radice ad ogni altro nodo siano alternanti viene detto albero alternante;*
- *Un albero alternante è massimale se non vi si possono aggiungere ulteriori nodi;*
- *Un cammino alternante viene detto cammino aumentante se termina con un nodo esposto.* ■

È opportuno notare che le definizioni dipendono dal particolare accoppiamento M . L'importanza dei cammini aumentanti viene espressa dal seguente teorema:

10.8 TEOREMA. *Un accoppiamento è di cardinalità massima se e solo se non esistono cammini aumentanti.*

DIMOSTRAZIONE. È evidente che se esiste un cammino aumentante è possibile ottenere un nuovo accoppiamento M' tale che $|M'| = |M| + 1$, semplicemente rendendo liberi gli archi accoppiati del cammino e viceversa, e quindi M non può essere ottimo.

Viceversa sia M un accoppiamento per il quale non esistono cammini aumentanti e sia M' un altro accoppiamento. Consideriamo l'insieme di archi $E' := M' \setminus M \cup M \setminus M'$. In generale E' è un insieme sconnesso di archi e, siccome il grado di ogni nodo rispetto a E' è al massimo due, consiste di circuiti e/o cammini disgiunti. Due archi di E' incidenti nello stesso nodo devono necessariamente appartenere l'uno a M e l'altro a M' . Quindi in tutti i cammini e i circuiti di E' gli archi di M e M' si alternano; inoltre i circuiti devono contenere lo stesso numero di archi di M e di M' . Siccome M non ha cammini aumentanti ogni cammino di E' deve contenere alle sue estremità archi di M che quindi risultano in numero maggiore di quelli di M' da cui $|M| \geq |M'|$. Dato che M' è un qualsiasi accoppiamento, M è ottimo. ■

Il teorema fornisce pertanto un metodo per ottenere un accoppiamento di cardinalità massima: dato un accoppiamento corrente, si tratta di trovare un cammino aumentante, aggiornare l'accoppiamento corrente che chiameremo *accoppiamento aumentato da M* e proseguire iterativamente fino a quando non esistono più cammini aumentanti. Il seguente teorema facilita la ricerca di cammini aumentanti:

10.9 TEOREMA. *Sia s un nodo esposto in un accoppiamento M e non esista nessun cammino aumentante da s relativamente all'accoppiamento M . Allora non esistono cammini aumentanti da s neppure relativamente ad accoppiamenti aumentati da M .*

DIMOSTRAZIONE. Supponiamo che in M aumentato da M' tramite il cammino aumentante (in M') $P' : s' \rightarrow t'$ esista un cammino aumentante $P : s \rightarrow t$. Se P e P' non hanno archi in comune si ha immediatamente la contraddizione che P è un cammino aumentante anche in M' . Se P e P' hanno archi in comune sia k il primo nodo di P appartenente anche a P' (vedi figura 10.1). Si costruisca ora un cammino \hat{P} partendo da s fino a k su P e poi proseguendo da k su P' cominciando con l'arco libero incidente a k fino all'estremo s' oppure t' (sia ad esempio s'). Il cammino $\hat{P} : s \rightarrow k \rightarrow s'$ è semplice per la scelta particolare di k . Inoltre, tenendo conto che gli archi da k a s' si presentano in M' con gli archi liberi e accoppiati invertiti, si è trovato un cammino aumentante $\hat{P} : s \rightarrow s'$ (oppure t') rispetto a M' . ■

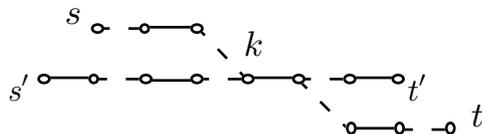


FIGURA 10.1

10.10 ESERCIZIO. Sia \hat{m} la cardinalità di un accoppiamento massimo e sia M un accoppiamento. Si dimostri che esistono $\hat{m} - |M|$ cammini aumentanti disgiunti nei nodi. ■

In base al teorema la ricerca di cammini aumentanti può essere fatta in modo sistematico a partire da un nodo esposto alla volta. Un nodo dal quale non esistono cammini aumentanti può non essere più preso in considerazione.

Il problema si sposta allora sulla determinazione di un cammino aumentante da un nodo esposto assegnato. Siccome l'esistenza di un cammino (qualsiasi) fra un nodo ed un insieme assegnato di nodi può essere determinata tramite una ricerca sul grafo, possiamo pensare di usare la stessa tecnica limitata però ai cammini alternanti. Questo è equivalente a generare un albero alternante $T(s)$ con radice nel nodo esposto s . Nel caso di grafi bipartiti vale il seguente teorema:

10.11 TEOREMA. *Sia il grafo G bipartito e sia M un accoppiamento. Allora il nodo esposto t appartiene a tutti gli alberi alternanti massimali $T(s)$ se e solo se esiste un cammino aumentante $s \rightarrow t$.*

DIMOSTRAZIONE. Se il nodo esposto t appartiene a tutti gli alberi alternanti massimali che si possono generare a partire da s è ovvio che esiste un cammino aumentante $s \rightarrow t$.

Supponiamo allora che esista un cammino aumentante $P : s \rightarrow t$ e che esista un albero massimale $T(s)$ per cui $t \notin T(s)$. Sia $t'' \rightarrow t' \rightarrow t$ la parte terminale di P , con t'' accoppiato con t' . Supponiamo $t' \in T(s)$. Siccome $T(s)$ è massimale, t' deve essere interno in $T(s)$. Però t' è esterno su P . Allora esiste un circuito dispari nel grafo, cosa impossibile se il grafo è bipartito. Quindi $t' \notin T(s)$. Supponiamo $t'' \in T(s)$. Però $T(s)$ è massimale ed essendo (t', t'') accoppiato la condizione $t'' \in T(s)$ e $t' \notin T(s)$ è impossibile. Il ragionamento si ripete analogo percorrendo a ritroso tutti gli altri nodi di P fino ad arrivare alla contraddizione $s \notin T(s)$. ■

La conseguenza importante del precedente teorema è che basta eseguire una ricerca ad albero di cammini alternanti (di costo $O(m)$) per trovare un eventuale cammino aumentante in un grafo bipartito $G = (N_1, N_2, E)$. Tenuto conto che tale ricerca viene ripetuta al massimo $\min\{|N_1|, |N_2|\} \in O(n)$ volte in base al teorema 10.9 abbiamo il seguente risultato:

10.12 TEOREMA. *Un assegnamento di massima cardinalità si può trovare, eseguendo ricerche di cammini aumentanti, con complessità $O(nm)$.* ■

Si può tuttavia far meglio di così trasformando il problema in un problema di massimo flusso. Si aggiungano al grafo bipartito $G = (N_1, N_2, E)$ un nodo sorgente s ed un nodo pozzo t , si aggiungano archi da s a ogni nodo di N_1 e da ogni nodo di N_2 a t . Tutti gli archi abbiano intervallo di capacità $[0, 1]$. È immediato verificare che la massimizzazione del flusso da s a t è equivalente a realizzare un assegnamento di cardinalità massima in G . Gli archi accoppiati sono esattamente quelli con flusso unitario.

L'algoritmo del massimo flusso delineato nel capitolo 8 ha in generale complessità $O(n^3)$. Ma nel caso particolare di capacità $[0, 1]$ si vede facilmente, ripercorrendo le varie fasi dell'algoritmo, che vi sono solo operazioni di saturazione e quindi la complessità è $O(mn)$ (un'analisi più approfondita porta a $O(mn^{2/3})$, si veda Jungnickel [1999] pp. 188-190). La seguente analisi (Hopcroft e Karp [1973]) permette però di concludere che bastano meno di n cicli aumentanti per il particolare problema di massimo flusso derivato dal problema dell'assegnamento.

10.13 TEOREMA. *Un assegnamento di massima cardinalità si può trovare, risolvendo un problema di massimo flusso, con complessità $O(m\sqrt{n})$.*

DIMOSTRAZIONE. Sia \hat{M} l'accoppiamento di cardinalità massima. Se $|\hat{M}| \leq \sqrt{n}$ la tesi è banalmente provata dato che ad ogni passo l'accoppiamento aumenta di almeno un'unità.

Sia allora $|\hat{M}| > \sqrt{n}$. Data la particolare struttura del grafo di massimo flusso, la rete a livelli è costituita da un primo livello che comprende archi da s a N_1 , da un secondo livello formato da archi non accoppiati (del grafo bipartito), poi da un certo numero (anche nullo) di coppie di livelli che alternano archi accoppiati e archi non accoppiati, e infine da un livello di archi da N_2 a t . Se indichiamo con L_k il numero di livelli contenenti archi accoppiati al k -esimo ciclo aumentante, si ha quindi che il numero di livelli è $2L_k + 3$.

Dall'esercizio 10.10 si ha che, se M_k è l'accoppiamento al passo k -esimo, esistono $|\hat{M}| - |M_k|$ cammini aumentanti disgiunti nei nodi. La rete a livelli, per come viene costruita, ha un numero di livelli pari al più corto cammino aumentante e questo non può usare più di

$|M_k|/(|\hat{M}| - |M_k|)$ archi accoppiati. Cioè

$$L_k \leq \frac{|M_k|}{|\hat{M}| - |M_k|}$$

Si noti ora che $k - 1 \leq L_k$ perché ad ogni iterazione il numero di livelli deve crescere e che, banalmente, $|M_k| \leq n$. Se si sceglie quel particolare valore di k per cui $|M_k| < |\hat{M}| - \sqrt{n}$ e $|M_{k+1}| \geq |\hat{M}| - \sqrt{n}$ abbiamo

$$k - 1 \leq L_k \leq \frac{|M_k|}{|\hat{M}| - |M_k|} < \frac{n}{\sqrt{n}} = \sqrt{n}$$

Le successive iterazioni sono limitate da $|\hat{M}| - |M_{k+1}| \leq \sqrt{n}$ e quindi la tesi è provata. ■

10.14 ESERCIZIO. Sia assegnato un grafo bipartito completo con costo c_e per ogni arco. Il problema dell'assegnamento di *costo estremo minimo* (o anche *costo collo di bottiglia*, *bottleneck cost*) consiste nel determinare un accoppiamento M che minimizza $\max_{e \in M} c_e$ (anziché $\sum_{e \in M} c_e$). Si sviluppi un algoritmo polinomiale per il problema. ■

Il fatto che il grafo sia bipartito costituisce un'ipotesi cruciale del teorema 10.11. Infatti la presenza di circuiti dispari fa sì che il teorema non sia valido in generale. Si veda l'accoppiamento in figura 10.2, per il quale esiste un cammino aumentante, ma esiste anche l'albero alternante massimale in figura 10.3, che *non* contiene il cammino aumentante.

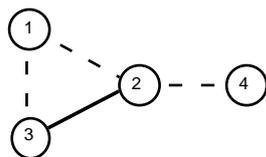


FIGURA 10.2

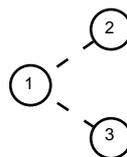


FIGURA 10.3

Nel caso generale si deve quindi adottare una strategia diversa nella ricerca di cammini alternanti. Tenendo conto che i nodi esterni di ogni cammino alternante sono nodi ‘obbligati’ una volta scelto il nodo interno, si può costruire un grafo orientato ausiliario $\hat{G} = (N, \hat{E})$, nel seguente modo:

$$(h, k) \in \hat{E} \iff k \text{ è accoppiato con un nodo adiacente ad } h.$$

Inoltre sia W_t l'insieme di nodi accoppiati adiacenti ad un nodo esposto t e sia $W := \cup_{t \neq s} W_t$ (*nodi bersaglio*). Si può quindi pensare di cercare un percorso da s a W tramite una ricerca ad albero su \hat{G} .

La ricerca ad albero condotta su \hat{G} è più ‘ampia’ di quella alternante su G . Infatti i percorsi su \hat{G} sono identificati solo dai nodi esterni dei cammini alternanti su G e quindi i nodi interni dei cammini possono comparire come nodi esterni di altri cammini alternanti. Indichiamo con $\hat{T}(s)$ un albero massimale su \hat{G} generato in questo modo a partire da s .

Con questo nuovo tipo di ricerca la situazione si è ribaltata rispetto al teorema 10.11. Infatti è immediato verificare che, se esiste un cammino aumentante da s a t in G , allora esiste un cammino da s a W_t in \hat{G} e quindi i nodi in W_t devono appartenere a qualsiasi albero $\hat{T}(s)$. Il problema consiste nel verificare se i cammini su G corrispondenti a cammini su $\hat{T}(s)$ sono semplici e quindi effettivamente aumentanti. Si consideri l'esempio in figura 10.4 che

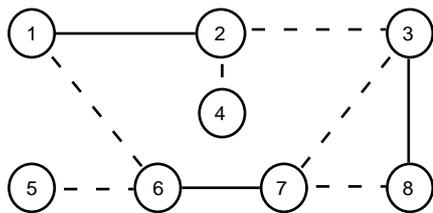


FIGURA 10.4

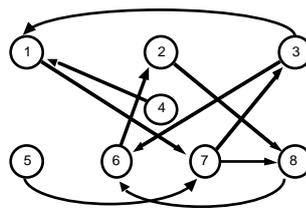


FIGURA 10.5

genera il grafo ausiliario in figura 10.5. Partendo dal nodo esposto 5 e cercando di raggiungere il nodo bersaglio 2, si trova il percorso $5 \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow 2$ che *non* è in realtà un cammino aumentante su G dato che corrisponde al cammino $5 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 4$ cioè ad un cammino che ritorna su se stesso (in senso opposto) dopo aver percorso un circuito dispari.

Quando un cammino ritorna su se stesso deve necessariamente percorrere un circuito dispari ‘pieno’, cioè contenente il massimo accoppiamento possibile. Tale circuito viene chiamato *germoglio* (in inglese *blossom*). Inoltre il cammino entra nel germoglio e ne esce dall’unico nodo non accoppiato con altri nodi del germoglio, che viene detto *base* del germoglio.

Il modo per aggirare l’ostacolo dovuto alla presenza dei germogli consiste, non appena viene generato un germoglio, nel costruire un altro grafo ottenuto per contrazione dei nodi del germoglio in uno pseudonodo e proseguire la ricerca del cammino aumentante su questo nuovo grafo. Affinché tale idea si traduca in un metodo valido occorre che sia vera la seguente proprietà:

10.15 TEOREMA. *Sia B un germoglio ottenuto nella ricerca di un cammino aumentante da s . Allora esiste un cammino aumentante da s su G se e solo se esiste un cammino aumentante da s sul grafo G' ottenuto per contrazione dei nodi di B .*

DIMOSTRAZIONE. La dimostrazione di questo teorema non è concettualmente difficile e viene perciò lasciata come esercizio. La dimostrazione del ‘se’ è quasi immediata. Per quel che riguarda il ‘solo se’ si tratta di prendere in esame i possibili modi in cui il (vero) cammino aumentante su G si intreccia con il (falso) cammino aumentante che ha creato il germoglio e quindi il grafo G' , e per ogni caso verificare l’esistenza di un circuito aumentante su G' . ■

Si noti che il germoglio deve essere generato durante una ricerca di cammini aumentanti affinché il teorema sia valido. Si consideri l’esempio in figura 10.6 dove esiste un cammino aumentante. I nodi 2, 3 e 4 formano un germoglio. Operata la contrazione si ottiene un grafo (figura 10.7) dove non ci sono cammini aumentanti. Infatti un tale germoglio non può essere trovata nella ricerca di un cammino aumentante.

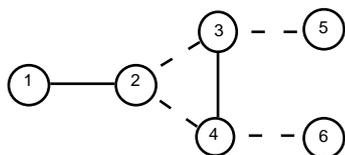


FIGURA 10.6

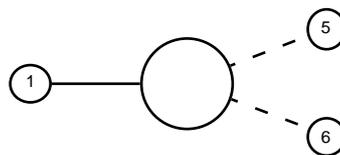


FIGURA 10.7

La procedura per la ricerca di un cammino aumentante può richiedere ulteriori contrazioni in modo ricorsivo fino a pervenire alla conclusione che non esistono cammini aumentanti, oppure alla determinazione di un cammino aumentante (nel grafo ottenuto con le contrazioni successive).

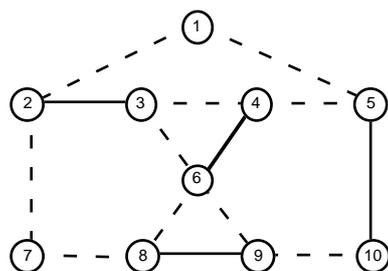


FIGURA 10.8

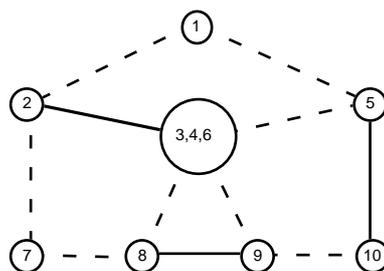


FIGURA 10.9

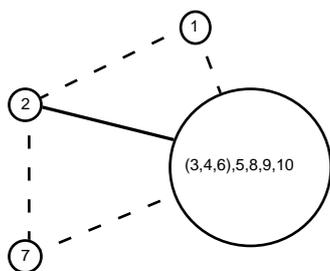


FIGURA 10.10

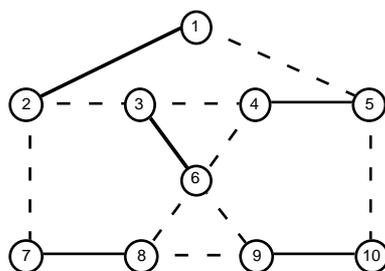


FIGURA 10.11

Nel primo caso il teorema 10.15 permette di concludere che non esistono cammini aumentanti dal nodo dato nel grafo originale. Nel secondo caso bisogna ricostruire il cammino aumentante nel grafo originale a partire da quello del grafo contratto. Questo si effettua espandendo a ritroso gli pseudonodi nei loro germogli e facendo attraversare il germoglio dal cammino aumentante nell'unico modo permesso. A titolo di esempio si consideri il grafo in figura 10.8 che presenta due nodi esposti. Possiamo supporre che la procedura che cerca il cammino aumentante generi sul grafo ausiliario il cammino $1 \rightarrow 3 \rightarrow 4 \rightarrow 2$ che corrisponde al falso cammino aumentante $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 7$. Si è quindi generato il germoglio $(3, 4, 6)$ da cui si ottiene il grafo contratto in figura 10.9. Supponiamo che sul grafo ausiliario del grafo contratto si generi questa volta il cammino $1 \rightarrow (3, 4, 6) \rightarrow 10 \rightarrow 8 \rightarrow 2$ che corrisponde al falso cammino aumentante $1 \rightarrow 2 \rightarrow (3, 4, 6) \rightarrow 5 \rightarrow 10 \rightarrow 9 \rightarrow 8 \rightarrow (3, 4, 6) \rightarrow 2 \rightarrow 7$. Il nuovo germoglio è $((3, 4, 6), 5, 8, 9, 10)$ che, contratto, dà luogo al grafo in figura 10.10. Questa volta si ottiene senza problemi il cammino aumentante $1 \rightarrow 2 \rightarrow ((3, 4, 6), 5, 8, 9, 10) \rightarrow 7$. Bisogna ora espandere questo cammino aumentante a ritroso nei vari grafi. Il germoglio $((3, 4, 6), 5, 8, 9, 10)$ può essere attraversato da due cammini nel grafo di figura 10.9, e cioè da $1 \rightarrow 2 \rightarrow (3, 4, 6) \rightarrow 8 \rightarrow 7$ oppure da $1 \rightarrow 2 \rightarrow (3, 4, 6) \rightarrow 5 \rightarrow 10 \rightarrow 9 \rightarrow 8 \rightarrow 7$, ma solo il secondo è un cammino alternante. Ora bisogna espandere questo cammino sul grafo originario e si ottiene il cammino $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 9 \rightarrow 8 \rightarrow 7$ dal quale si ottiene l'accoppiamento aumentato di figura 10.11.

L'algoritmo appena delineato ha una complessità computazionale che dipende dalla sua implementazione. Certamente bisogna operare $O(n)$ ricerche di cammini aumentanti (anche se è conveniente iniziare da un accoppiamento massimale ottenuto con una euristica qualsiasi, pur tuttavia il numero di nodi esposti può sempre essere $O(n)$). Cruciale è il modo con cui si realizza la ricerca del cammino aumentante. Un modo semplicistico di procedere richiederebbe una generazione del grafo ausiliario con conseguente ricerca per ogni germoglio trovato. Siccome il numero di germogli trovati è $O(n)$ (il numero di contrazioni non può essere superiore al numero dei nodi) ed una ricerca sul grafo ha complessità $O(m)$, si ottiene una complessità globale $O(n^2 m)$.

Si può tuttavia migliorare questa complessità senza ricominciare daccapo la ricerca quando si trova un germoglio e quindi utilizzando l'informazione già ottenuta. A questo scopo bisogna realizzare una struttura dati che permetta di: 1) identificare un insieme di nodi corrispondente ad un germoglio, cioè uno pseudonodo, 2) determinare rapidamente a quale pseudonodo appartiene un nodo, 3) creare un nuovo pseudonodo da più nodi o pseudonodi.

Questo si ottiene rappresentando ogni sottoinsieme di nodi come un albero con radice in cui ogni nodo dell'albero è un elemento del sottoinsieme e la radice è un elemento del sottoinsieme che funge da 'rappresentante' e che identifica il sottoinsieme (si veda anche il capitolo 2 di Tarjan [1983]). Ogni nodo dell'albero punta al suo nodo genitore mentre la radice punta a se stessa. Quindi la determinazione del sottoinsieme cui appartiene un nodo assegnato richiede un numero di passi pari alla profondità dell'albero.

Fondere assieme due sottoinsiemi disgiunti, cioè due alberi, richiede una complessità costante, in quanto basta 'dirottare' il puntatore di una radice verso l'altra radice. Nel caso gli alberi abbiano profondità diversa è ovviamente conveniente mantenere come radice quella dell'albero più profondo e quindi la profondità del nuovo albero è uguale alla maggiore delle due profondità. Nel caso le profondità siano uguali è indifferente quale radice mantenere come tale, però questa volta la profondità del nuovo albero è di una unità più elevata di quella degli alberi originari. Dimostriamo ora per induzione che, partendo da alberi costituiti da un singolo elemento e operando in questo modo, la profondità di un albero qualsiasi è $O(\log_2 n)$.

Sia $p(T)$ la profondità di un albero T . Se T è costituito da un singolo elemento allora $p(T) = 0$ e si ha $p(T) = \log_2 |T| = 0$. Quindi la proprietà è verificata se $|T| = 1$. Dati due alberi T_1 e T_2 per i quali la proprietà sia verificata, la loro fusione genera un albero T_3 per il quale si ha, se $p(T_1) \neq p(T_2)$ (sia $p(T_1) > p(T_2)$),

$$p(T_3) = p(T_1) \leq \log_2 |T_1| \leq \log_2 (|T_1| + |T_2|) = \log_2 |T_3|$$

e, se $p(T_1) = p(T_2)$ (sia $|T_1| \leq |T_2|$),

$$p(T_3) = p(T_1) + 1 \leq \log_2 |T_1| + \log_2 2 = \log_2 2 |T_1| \leq \log_2 (|T_1| + |T_2|) = \log_2 |T_3|$$

e quindi la proprietà è ancora verificata. È possibile abbassare ancora la limitazione sulla profondità degli alberi con opportune operazioni sugli stessi. Per un'analisi di questo genere si rinvia a Tarjan [1983].

Per la ricerca dei cammini aumentanti è sufficiente descrivere il grafo assegnando, per ogni nodo, la lista degli archi incidenti (e dei rispettivi nodi adiacenti) da scandire con complessità pari al numero degli archi incidenti. Inoltre l'accoppiamento può essere rappresentato indicando per ogni nodo il suo nodo accoppiato (e ponendo ad esempio 0 per i nodi esposti). Per generare l'albero di ricerca (che consiste di nodi o pseudonodi esterni di cammini aumentanti nel più recente grafo contratto) si mantiene un insieme di archi attivi, cioè di archi incidenti in nodi già inseriti nell'albero e che devono essere ancora verificati per una possibile estensione dei cammini aumentanti. L'insieme degli archi attivi viene aggiornato quando: 1) un nodo esterno di un cammino alternante viene aggiunto all'albero, 2) si genera uno pseudonodo. Nel primo caso vengono inseriti tutti gli archi incidenti al nodo aggiunto (tranne quello accoppiato). Nel secondo caso vengono inseriti tutti gli archi incidenti ai nodi interni del germoglio (quelli ai nodi esterni risultano già inseriti in fasi precedenti).

Potrebbe sorgere il problema di non riuscire ad identificare gli archi incidenti in uno pseudonodo, nel momento in cui andrebbero inclusi fra gli archi attivi. Infatti non è disponibile la lista di incidenza di uno pseudonodo, in quanto le liste di incidenza rimangono espresse

Algoritmo cammini aumentanti

(* A è l'insieme di archi attivi. T è l'albero di ricerca

$find(i)$ trova il rappresentante dello pseudonodo cui appartiene i .

$mate(i)$ fornisce il nodo accoppiato a i . *)

```

input  $(s, M)$ ;
  inizializza  $A$ ;
  while  $\nexists$  cammino aumentante  $\wedge A \neq \emptyset$ 
  do begin
     $(i, j) \in A$ ;  $A := A \setminus (i, j)$ ;
     $ri := find(i)$ ,  $rj := find(j)$ ;
     $k := mate(j)$ ;
    if  $ri \neq rj$ 
    then
      if  $rj$  esposto
      then  $\exists$  cammino aumentante
      else
        if  $rj \in T$ 
        then contrai il germoglio e aggiorna  $A$ 
        else
          if  $k \notin T$ 
          then  $T := T \cup rj$  e aggiorna  $A$ ;
    end;
    if  $\exists$  cammino aumentante
    then aggiorna  $M$ 
  output  $(M)$ .

```

secondo i nodi originali. Tuttavia non serve calcolare la lista d'incidenza di uno pseudonodo perché, né i nodi interni di un germoglio, né i nodi esterni 'nuovi' possono essere pseudonodi. Quindi l'aggiornamento della lista degli archi attivi può essere fatto senza rigenerare la lista d'incidenza del grafo contratto.

Gli archi vengono successivamente prelevati uno alla volta dall'insieme degli archi attivi per la verifica. Quando un arco viene verificato può avvenire che i suoi estremi siano pseudonodi e quindi va fatta, con complessità $O(\log n)$, la ricerca del rappresentante dello pseudonodo. La verifica di un arco può portare alle seguenti conclusioni (sia (i, j) l'arco in esame con i nodo o pseudonodo esterno sull'albero di ricerca e j nodo o pseudonodo): 1) i e j appartengono allo stesso pseudonodo, 2) j è un nodo esposto, 3) j è accoppiato ed è già presente sull'albero di ricerca, 4) j non è presente sull'albero di ricerca ed è accoppiato con un nodo o pseudonodo k già presente sull'albero di ricerca, 5) j non è presente sull'albero di ricerca ed è accoppiato con un nodo k non presente sull'albero di ricerca.

In ogni caso l'arco in esame viene tolto dall'insieme degli archi attivi. Inoltre a seconda dell'esito della verifica si opera nel seguente modo: 1) si passa ad esaminare un altro arco attivo, 2) si termina perché si è trovato un cammino aumentante, 3) si è trovato un germoglio e quindi bisogna procedere alla contrazione del germoglio, 4) si passa ad esaminare un altro arco attivo, 5) si aggiunge il nodo esterno k all'albero di ricerca.

Per quanto già detto, l'aggiornamento degli archi attivi viene fatto nei casi 3) e 5). Si noti ancora che, per poter stabilire per ogni pseudonodo qual è il suo nodo o pseudonodo accoppiato, è utile avere a disposizione per ogni rappresentante di pseudonodo anche la base del germoglio relativo.

Una volta trovato nel grafo contratto il cammino aumentante, bisogna ritrovare il cammino

aumentante nel grafo originale e a questo fine bisogna mantenere una tabella delle contrazioni successive. La ricostruzione del cammino aumentante può essere eseguita con complessità $O(n)$ (dettaglio come esercizio per il lettore).

Quindi per la ricerca di un cammino aumentante bisogna verificare al più tutti gli archi. L'operazione più costosa di ogni verifica è la ricerca dei rappresentanti degli estremi dell'arco con complessità $O(\log n)$. Per quel che riguarda le varie operazioni di contrazione ve ne sono al più n e siccome il costo di ognuna è costante si ha un costo complessivo di $O(n)$. Tenendo conto che si eseguono al più n ricerche di cammini aumentanti si perviene quindi al seguente risultato:

10.16 TEOREMA. *Un accoppiamento di cardinalità massima su un grafo generico si trova con complessità $O(nm \log n)$.* ■

Questo risultato non è il migliore possibile. Si può ridurre il fattore $\log n$ ad una funzione a crescita lentissima in n sfruttando i risultati esposti in Tarjan [1983] oppure si può adottare una strategia diversa nella ricerca dei cammini aumentanti pervenendo alla limitazione $O(m\sqrt{n})$ (Micali e Vazirani [1980]). Questo risultato è a tutt'oggi la migliore limitazione ottenuta. È sorprendente che i migliori valori noti di complessità per grafi bipartiti e per grafi generici coincidano, nonostante la particolarità dei grafi bipartiti.

10.2. Descrizione poliedrale

Dato un grafo G sia $\mathcal{M}(G)$ l'insieme di tutti i possibili accoppiamenti su G . Per ogni accoppiamento M sia $e(M) \in \mathbb{R}^m$ il suo vettore d'incidenza. Il poliedro

$$P_M(G) := \text{conv} \{e(M) : M \in \mathcal{M}(G)\}$$

prende il nome di *poliedro dell'accoppiamento* (*matching polyhedron*). Per quanto visto, minimizzare una funzione lineare su $P_M(G)$ dà come risultato un vertice e cioè un accoppiamento. Il problema è che disponiamo di algoritmi risolutivi per la minimizzazione di una funzione lineare su un poliedro quando questo è descritto da disequazioni lineari. Siamo quindi interessati a trovare le disequazioni che definiscono $P_M(G)$. Vedremo subito che il problema è molto più semplice se il grafo è bipartito.

10.17 ESERCIZIO. Dimostrare che $P_M(G)$ ha dimensione piena. ■

Quando la ricerca si limita agli accoppiamenti perfetti $\overline{\mathcal{M}}(G)$ di G , possiamo definire il *poliedro dell'accoppiamento perfetto* (*perfect matching polyhedron*)

$$P_{\overline{M}}(G) := \text{conv} \{e(M) : M \in \overline{\mathcal{M}}(G)\}$$

Definiamo ora i seguenti poliedri:

$$Q_M^0(G) := \{x \in \mathbb{R}^m : A(G)x \leq \mathbf{1}, x \geq 0\}$$

$$Q_{\overline{M}}^0(G) := \{x \in \mathbb{R}^m : A(G)x = \mathbf{1}, x \geq 0\}$$

dove $A(G)$ è la matrice d'incidenza nodi-archi del grafo G . È immediato che $P_M(G) \subset Q_M^0(G)$ e $P_{\overline{M}}(G) \subset Q_{\overline{M}}^0(G)$, in quanto il vincolo espresso dalla matrice d'incidenza impone proprio che il grado in ogni nodo sia al più uno oppure esattamente uno. Tuttavia l'inclusione potrebbe essere propria, nel qual caso alcuni vertici di $Q_M^0(G)$ o $Q_{\overline{M}}^0(G)$ non

avrebbero coordinate intere e quindi non sarebbero interpretabili come accoppiamenti (se sono interi non possono che avere coordinate 0 o 1) e in definitiva non avrebbe senso risolvere il problema dell'accoppiamento pesato come $\min \{cx : x \in Q_M^0(G)\}$. Lo studio poliedrale dei problemi combinatori consiste proprio nel raffinare la descrizione del poliedro con ulteriori diseuguaglianze quando si trovino vertici con coordinate frazionarie. La situazione è estremamente semplice se il grafo è bipartito.

10.18 TEOREMA. *Se il grafo è bipartito $P_M(G) = Q_M^0(G)$ e $P_{\overline{M}}(G) = Q_{\overline{M}}^0(G)$.*

DIMOSTRAZIONE. Un vertice di $Q_M^0(G)$ ($Q_{\overline{M}}^0(G)$) è associato ad una soluzione di base del sistema $A(G)x \leq \mathbf{1}$ ($A(G)x = \mathbf{1}$), $x \geq 0$. Siccome la matrice d'incidenza nodi-archi di un grafo bipartito è totalmente unimodulare (vedi corollario 2.40) le soluzioni di base devono essere intere. ■

Pertanto i problemi d'accoppiamento bipartito (normale o perfetto) possono essere riformulati come i due seguenti problemi di programmazione lineare:

$$\begin{array}{ll} \max & cx \\ & A(G)x \leq \mathbf{1} \\ & x \geq 0 \end{array} \quad \text{e} \quad \begin{array}{ll} \min & cx \\ & A(G)x = \mathbf{1} \\ & x \geq 0 \end{array}$$

Per ciò che riguarda il problema generale si deve invece notare che la matrice d'incidenza nodi-archi di un grafo non orientato non è totalmente unimodulare e infatti avviene che le inclusioni $P_M(G) \subset Q_M^0(G)$ e $P_{\overline{M}}(G) \subset Q_{\overline{M}}^0(G)$ siano strette. Dal corollario 2.38 si ha che le componenti dell'inversa di una qualsiasi sottomatrice quadrata non singolare di $A(G)$ possono assumere soltanto i valori $\{0, \pm 1/2, \pm 1, \pm 2\}$ e quindi le soluzioni ammissibili di base per il vincolo $A(G)x = \mathbf{1}$ (oppure $A(G)x \leq \mathbf{1}$), $x \geq 0$, possono assumere soltanto i valori $\{0, 1/2, 1\}$. Inoltre il valore $1/2$ può essere assunto solo dagli archi di circuito dei quasi-alberi dispari del grafo associato alla matrice di base.

10.19 ESERCIZIO. Si dimostri che $P_{\overline{M}}(G) = Q_{\overline{M}}^0(G)$ per K_4 (suggerimento: si dimostri che in generale i quasi alberi con componenti frazionarie di una soluzione di base devono essere in numero pari). ■

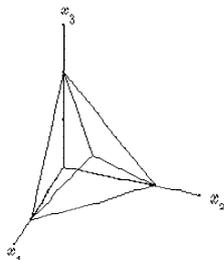


FIGURA 10.12

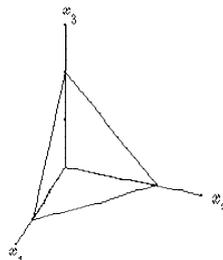


FIGURA 10.13

La situazione può essere illustrata per $Q_M^0(G)$ applicato al grafo K_3 . In questo caso $Q_M^0(G)$, raffigurato in figura 10.12, è il poliedro definito da

$$\begin{array}{ll} x_1 + x_2 & \leq 1 \\ x_1 & + x_3 \leq 1 \\ & x_2 + x_3 \leq 1 \\ x_1 \geq 0, & x_2 \geq 0, x_3 \geq 0 \end{array}$$

i cui vertici sono $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, $(1/2, 1/2, 1/2)$.

È chiaro che se S è un insieme dispari di nodi non vi possono essere più di $(|S| - 1)/2$ archi accoppiati fra gli archi di $E(S)$. Quindi la disequaglianza

$$\sum_{e \in E(S)} x_e \leq \frac{|S| - 1}{2} \tag{10.1}$$

può essere aggiunta a quelle definiti $Q_M^0(G)$ senza perdere soluzioni intere. D'altro lato ogni soluzione di valore $1/2$ su un circuito in S non è ammissibile per il nuovo vincolo. Quindi l'effetto di aggiungere a $Q_M^0(G)$ il vincolo (10.1) è quello di 'tagliare' i vertici indesiderati. Per l'esempio precedente si tratterebbe di aggiungere il vincolo $x_1 + x_2 + x_3 \leq 1$ e il poliedro che si ottiene è quello rappresentato in figura 10.13 i cui vertici sono effettivamente tutti i possibili accoppiamenti.

Possiamo allora pensare di aggiungere i seguenti vincoli: sia \mathcal{S} l'insieme di tutti gli insiemi maggiori di uno e dispari di nodi (si noti che $|\mathcal{S}| = 2^{n-1} - n$). Sia D la matrice $|\mathcal{S}| \times m$ definita come:

$$D_S^e := \begin{cases} 1 & \text{se } e \in E(S) \\ 0 & \text{altrimenti} \end{cases}$$

e sia $d_S = (|S| - 1)/2$. L'aggiunta dei vincoli $Dx \leq d$ rende quindi inammissibili le soluzioni frazionarie di valore $1/2$. Però potrebbe aggiungere vertici frazionari di altro tipo. Per fortuna ciò non succede. Il notevole risultato di Edmonds [1965b] è che i poliedri

$$\begin{aligned} Q_M^1(G) &:= \{x \in \mathbb{R}^m : A(G)x \leq \mathbf{1}, Dx \leq d, x \geq 0\} \\ Q_{\overline{M}}^1(G) &:= \{x \in \mathbb{R}^m : A(G)x = \mathbf{1}, Dx \leq d, x \geq 0\} \end{aligned}$$

sono uguali rispettivamente a $P_M(G)$ e $P_{\overline{M}}(G)$. Questo risultato può essere dimostrato in vari modi alternativi. Qui lo dimostreremo seguendo il metodo induttivo di Schrijver (vedi Schrijver [1986] p. 109) e successivamente lo dimostreremo in modo costruttivo provando la correttezza dell'algoritmo di Edmonds. Una terza dimostrazione (che non viene qui esposta) è dovuta a Lovász [1979] e si basa sulla tecnica di dimostrare che ogni disequaglianza che definisce una faccetta di $P_M(G)$ è necessariamente inclusa fra le disequaglianze che definiscono $Q_M^1(G)$.

Dimostreremo il teorema prima per il poliedro dell'accoppiamento perfetto e poi da questo dedurremo il risultato analogo per il poliedro dell'accoppiamento. Nel seguito useremo la notazione compatta $x(S) := \sum_{e \in E(S)} x_e$, per cui una singola disequaglianza del vincolo $Dx \leq d$ può anche essere espressa come $x(S) \leq (|S| - 1)/2$, e $x(Q(S)) := \sum_{e \in Q(S)} x_e$. Preliminarmente serve il seguente lemma:

10.20 LEMMA. *Sia $x \in Q_M^0(G)$ e sia S un insieme dispari. Allora $x(S) \leq (|S| - 1)/2$ se e solo se $x(Q(S)) \geq 1$ e in particolare $x(S) = (|S| - 1)/2$ se e solo se $x(Q(S)) = 1$.*

DIMOSTRAZIONE. Ai fini della dimostrazione è utile introdurre delle variabili y_{ij} , $i, j \in N$, tali che $y_{ij} := y_{ji} := x_e$ se $e = (i, j)$, $y_{ij} := 0$ altrimenti. Quindi

$$x(S) = \frac{1}{2} \sum_{i \in S} \sum_{j \in S} y_{ij} \qquad x(Q(S)) = \sum_{i \in S} \sum_{j \notin S} y_{ij}$$

Siccome

$$2x(S) + x(Q(S)) = \sum_{i \in S} \sum_{j \in S} y_{ij} + \sum_{i \in S} \sum_{j \notin S} y_{ij} = \sum_{i \in S} \sum_j y_{ij} = \sum_{i \in S} 1 = |S|$$

si ricava immediatamente che la disequaglianza $x(S) \leq (|S| - 1)/2$ è equivalente alla disequaglianza $x(Q(S)) \geq 1$ e che in particolare $x(S) = (|S| - 1)/2 \iff x(Q(S)) = 1$. ■

10.21 ESERCIZIO. Nella definizione di $Q_M^1(G)$ (ma non in quella di $Q_M^1(G)$) molte disequazioni sono ridondanti. Quali? ■

10.22 TEOREMA. $P_M(G) = Q_M^1(G)$, ovvero ogni vertice di $Q_M^1(G)$ corrisponde ad un accoppiamento perfetto e viceversa.

DIMOSTRAZIONE. L'inclusione $P_M(G) \subset Q_M^1(G)$ è ovvia. La difficoltà del teorema consiste nel dimostrare che $Q_M^1(G) \subset P_M(G)$.

Faremo vedere che un punto di $Q_M^1(G)$ con coordinate frazionarie può essere espresso come combinazione convessa stretta di altri punti di $Q_M^1(G)$ e quindi non può essere vertice. Dunque i vertici di $Q_M^1(G)$ devono avere coordinate intere e devono corrispondere ad accoppiamenti da cui l'inclusione.

Consideriamo dapprima il caso di un punto $x \in Q_M^1(G)$ tale che $Dx < d$. Indichiamo come frazionari gli archi e tali che $0 < x_e < 1$. Sia $\hat{G} = (N, \hat{E})$ il grafo formato dagli archi frazionari e supponiamo senza perdita di generalità che \hat{G} sia connesso. Si noti che ogni nodo è incidente ad almeno due archi di \hat{G} e quindi \hat{G} deve contenere almeno un circuito. Sono possibili i seguenti casi:

1) \hat{G} contiene un circuito C pari: si indichino alternativamente gli archi del circuito come pari (C^+) e dispari (C^-) e si considerino le soluzioni

$$x'_e := \begin{cases} x_e & e \notin C \\ x_e + \varepsilon & e \in C^+ \\ x_e - \varepsilon & e \in C^- \end{cases}, \quad x''_e := \begin{cases} x_e & e \notin C \\ x_e - \varepsilon & e \in C^+ \\ x_e + \varepsilon & e \in C^- \end{cases}$$

Ovviamente $x = 1/2(x' + x'')$. Inoltre siccome $Dx < d$ e $0 < x < 1$ esiste $\varepsilon > 0$ per cui $x' \in Q_M^1(G)$ e $x'' \in Q_M^1(G)$ (ovviamente x' e x'' rispettano il vincolo sul grado di ogni nodo).

2) \hat{G} non contiene circuiti pari. \hat{G} non può essere costituito da un solo circuito dispari, perché in questo caso il vincolo sul grado ammetterebbe per gli archi del circuito solo il valore $x_e = 1/2$, inammissibile però per il vincolo $Dx \leq d$. Quindi devono esistere almeno due circuiti dispari. Siccome due circuiti dispari con archi in comune generano un circuito pari, devono esistere due circuiti connessi con un cammino. Si possono aumentare e diminuire di ε le soluzioni lungo i circuiti e sfruttare il cammino che congiunge i circuiti per bilanciare i gradi. Anche in questo caso allora si ottiene che x è combinazione convessa di soluzioni ammissibili.

Si supponga ora che vi siano sottoinsiemi S per i quali $x(S) = (|S| - 1)/2$. Si consideri nuovamente senza perdita di generalità solamente il grafo dato dagli archi frazionari. Dal lemma precedente esiste quindi un taglio $Q(S)$ (contenente almeno due archi) tale che $\sum_{e \in Q(S)} x_e = 1$. I valori $x_e, e \in Q(S)$, verranno usati come coefficienti di una combinazione convessa stretta. Dato il taglio, si possono generare due grafi G_1 e G_2 , G_1 ottenuto da \hat{G} contraendo $N \setminus S$ in uno pseudonodo j^* e G_2 ottenuto da \hat{G} contraendo S in uno pseudonodo i^* . Inoltre possiamo definire su G_1 la soluzione

$$x_e^1 := \begin{cases} x_e & e \in E(S) \\ \sum_{j \notin S} x_{ij} & e = (i, j^*) \end{cases}$$

e su G_2 la soluzione

$$x_e^2 := \begin{cases} x_e & e \in E(N \setminus S) \\ \sum_{i \in S} x_{ij} & e = (i^*, j) \end{cases}$$

Si noti che il vincolo del grado è rispettato nei due pseudonodi dato che $\sum_{e \in Q(S)} x_e = 1$.

Ora lo scopo è di dimostrare che esistono soluzioni y_1, \dots, y_p in G_1 la cui combinazione convessa è x^1 , e soluzioni z_1, \dots, z_q in G_2 la cui combinazione convessa è x^2 . In particolare si vuole che y_i e z_j siano accoppiamenti. Note queste soluzioni si può trovare la combinazione convessa che genera x su \hat{G} nel modo che verrà illustrato nel prossimo paragrafo. La dimostrazione è ricorsiva in quanto la determinazione dell'esistenza delle soluzioni y_i e z_i può a sua volta richiedere la contrazione di altre parti del grafo (ogniqualevolta esiste S per cui $x(S) = (|S| - 1)/2$) fino ad ottenere sottografi i cui valori frazionari sono combinazione convessa di accoppiamenti in base ai risultati precedenti.

La procedura secondo cui si fondono assieme le soluzioni y_i e z_j per ottenere la combinazione convessa che genera x è alquanto complessa e viene qui spiegata facendo direttamente riferimento ad un esempio concreto. Si supponga allora che \hat{G} sia il grafo in figura 10.14 dove i numeri accanto agli archi sono i valori x_e moltiplicati per 100:

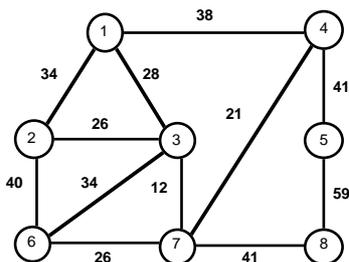


FIGURA 10.14

Si noti che $x(Q(\{1, 2, 3, 6, 7\})) = 1$ e $x(Q(\{2, 3, 6\})) = 1$. Si consideri inizialmente $S = \{1, 2, 3, 6, 7\}$ e si operino le due contrazioni di S e $N \setminus S$ che danno luogo ai grafi G_1 e G_2 indicati nella figura 10.15 insieme alle rispettive soluzioni x^1 e x^2 .

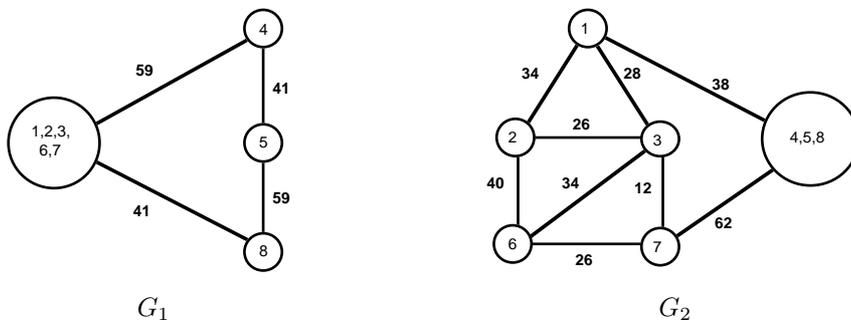


FIGURA 10.15

La scomposizione di x^1 come combinazione convessa dei due accoppiamenti in figura 10.16 è immediata

$$x^1 = \frac{59}{100} e(M_1) + \frac{41}{100} e(M_2)$$

dove $e(M_i)$ è il vettore d'incidenza dell'accoppiamento M_i .

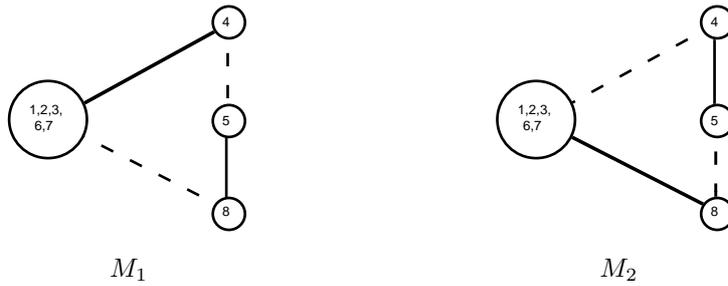


FIGURA 10.16

Il calcolo di x^2 richiede l'ulteriore contrazione in G_2 dell'insieme $\{2, 3, 6\}$ e del suo complementare, producendo i grafi G_3 e G_4 indicati nella figura 10.17 insieme alle soluzioni x^3 e x^4 , per le quali sono disponibili immediatamente le decomposizioni (si veda la figura 10.18)

$$\begin{aligned}
 x^3 &= \frac{62}{100} e(M_3) + \frac{38}{100} e(M_4) \\
 x^4 &= \frac{34}{100} e(M_5) + \frac{26}{100} e(M_6) + \frac{40}{100} e(M_7)
 \end{aligned}
 \tag{10.2}$$

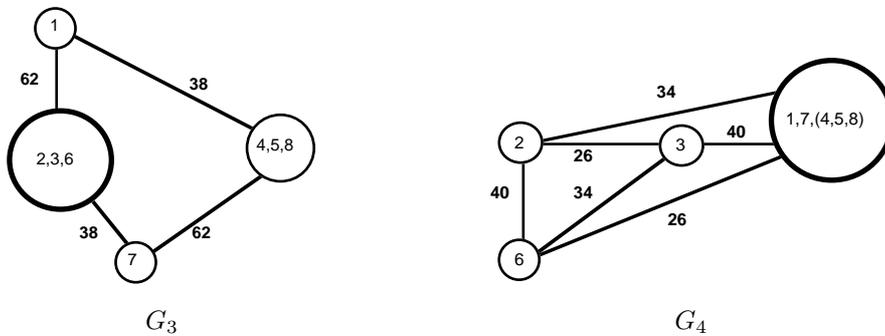


FIGURA 10.17

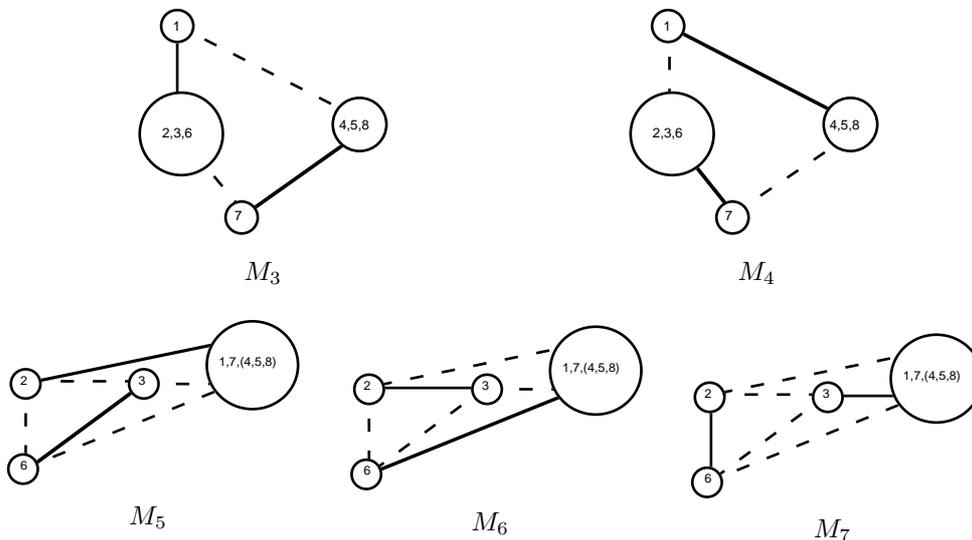


FIGURA 10.18

Si tratta ora di ‘fondere’ assieme gli accoppiamenti M_3 e M_4 con gli accoppiamenti M_5 , M_6 e M_7 . Si immagini di moltiplicare i valori di x^2 per un intero in modo da avere valori interi. Questa operazione è sempre possibile a causa della razionalità dei numeri coinvolti e in questo caso tale intero è 100. Possiamo allora pensare, data l’espressione (10.2), di avere a disposizione 62 copie di M_3 , 38 di M_4 , 34 di M_5 , 26 di M_6 e 40 di M_7 . Si noti che, per ognuno di questi accoppiamenti, solo un arco appartiene al taglio e che ogni arco del taglio in G_2 si trova ‘immerso’ in esattamente un arco di G_3 e un arco di G_4 . Ad esempio l’arco $(1, 2)$ di G_2 è immerso in $(1, (2, 3, 6))$ di G_3 e $(2, (1, 7, (4, 5, 8)))$ di G_4 . L’arco $(1, (2, 3, 6))$ è coperto da M_3 e l’arco $(2, (1, 7, (4, 5, 8)))$ è coperto da M_5 . Quindi per ottenere le 34 copie di $(1, 2)$ dobbiamo prelevare 34 copie da M_3 e 34 copie da M_5 e unire questi due accoppiamenti in G_2 . Si procede così per ogni arco del taglio in G_2 ottenendo complessivamente

- arco $(1, 2)$: 34 copie di M_3 e di M_5
- arco $(1, 3)$: 28 copie di M_3 e di M_7
- arco $(3, 7)$: 12 copie di M_4 e di M_7
- arco $(6, 7)$: 26 copie di M_4 e di M_6

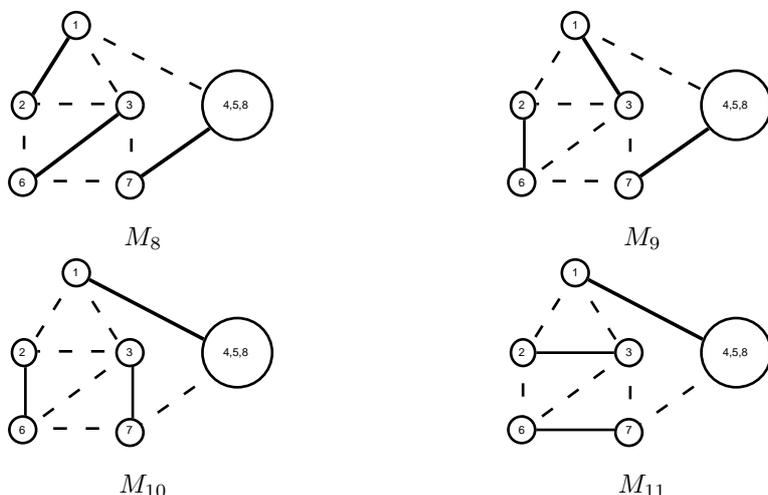


FIGURA 10.19

Si indichi $M_8 := M_3 \cup M_5$, $M_9 := M_3 \cup M_7$, $M_{10} := M_4 \cup M_7$, $M_{11} := M_4 \cup M_6$ (figura 10.19) e quindi si ha

$$x^2 = \frac{34}{100} e(M_8) + \frac{28}{100} e(M_9) + \frac{12}{100} e(M_{10}) + \frac{26}{100} e(M_{11})$$

A questo punto si tratta di unire con un procedimento analogo gli accoppiamenti M_1 e M_2 con gli accoppiamenti M_8 , M_9 , M_{10} e M_{11} . Si ottiene allora per gli archi del taglio in \hat{G}

- arco $(1, 4)$: 12 copie di M_1 e di M_{10}
- arco $(1, 4)$: 26 copie di M_1 e di M_{11}
- arco $(4, 7)$: 21 copie di M_1 e di M_8
- arco $(7, 8)$: 13 copie di M_2 e di M_8
- arco $(7, 8)$: 28 copie di M_2 e di M_9

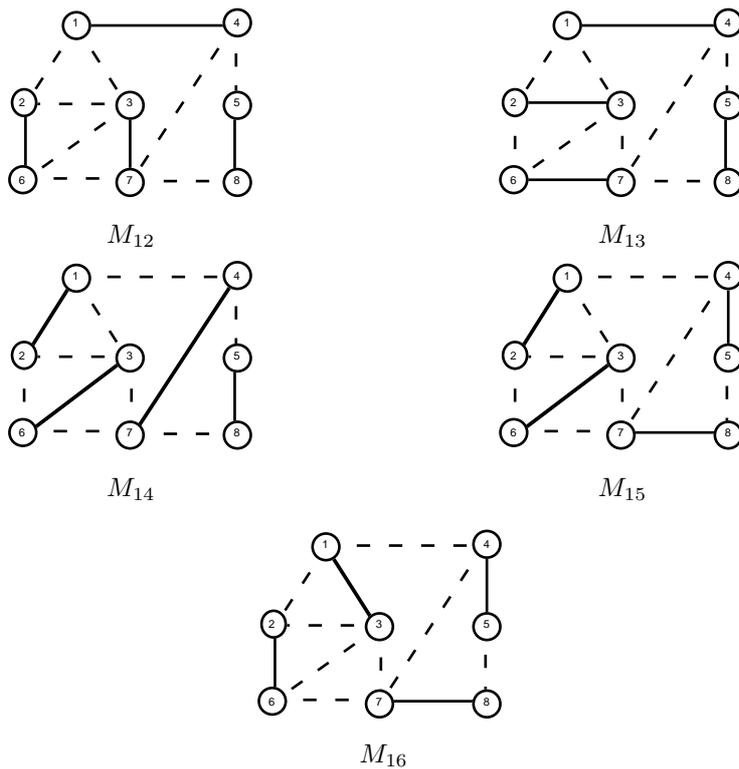


FIGURA 10.20

Si indichi allora $M_{12} := M_1 \cup M_{10}$, $M_{13} := M_1 \cup M_{11}$, $M_{14} := M_1 \cup M_8$, $M_{15} := M_2 \cup M_8$, $M_{16} := M_2 \cup M_9$ (figura 10.20) e si può concludere affermando che

$$x = \frac{12}{100} e(M_{12}) + \frac{26}{100} e(M_{13}) + \frac{21}{100} e(M_{14}) + \frac{13}{100} e(M_{15}) + \frac{28}{100} e(M_{16})$$

■

Non è difficile ora estendere i risultati a $Q_M^1(G)$:

10.23 TEOREMA. $P_M(G) = Q_M^1(G)$, ovvero ogni vertice di $Q_M^1(G)$ corrisponde ad un accoppiamento e viceversa.

DIMOSTRAZIONE. Si può ripercorrere la dimostrazione del precedente teorema. Si suppone dapprima $x \in Q_M^1(G)$ e $Dx < d$. In questo caso però vi possono essere dei nodi del grafo \hat{G} con un solo arco incidente, se il vincolo del grado non è attivo per la particolare soluzione x frazionaria. Non è tuttavia difficile vedere che esiste sempre la possibilità di esprimere x come combinazione convessa di altre soluzioni ammissibili. Poi, se esistono insiemi S tali $x(S) = (|S| - 1)/2$, si ragiona come nel teorema precedente. ■

Nell'esercizio 10.21 si è visto come, fra le disuguaglianze definenti $Q_M^1(G)$, molte siano ridondanti e (si veda anche ad esempio l'esercizio 10.19) non tutte definiscano faccette. La descrizione di $Q_M^1(G)$ risulta invece essenziale se il grafo è completo.

10.24 TEOREMA. Se $n > 3$, ogni disuguaglianza di $Q_M^1(G)$ definisce una faccetta di $P_M(G)$.

DIMOSTRAZIONE. Per dimostrare che una disuguaglianza definisce una faccetta basta far vedere che esiste una soluzione ammissibile in cui la disuguaglianza è attiva mentre tutte le

altre non lo sono. Vi sono tre tipi di disequaglianze da verificare, e cioè

$$\text{a) } x_e \geq 0 \quad \text{b) } x(Q(\{i\})) \leq 1 \quad \text{c) } x(S) \leq (|S| - 1)/2$$

a) Sia $\bar{x}_e := \varepsilon > 0$, $\forall e \neq e'$, $\bar{x}_{e'} := 0$. Per ε sufficientemente piccolo le disequaglianze di tipo b) e c) non sono attive e l'unica attiva è $x_{e'} \geq 0$.

b) Sia $\bar{x}_e := 1/(n-1)$ per ogni arco incidente nel nodo i e $\bar{x}_e := \varepsilon > 0$ per ogni altro arco. È immediato vedere che le disequaglianze di tipo a) e b) (tranne quella per il nodo i) sono non attive. Per le disequaglianze di tipo c) si noti che, dato un insieme S di nodi, vi sono al più $(|S|-1)$ archi di valore $1/(n-1)$ in $E(S)$ dato che tali archi formano un albero. Siccome in $E(S)$ vi sono $|S|(|S|-1)/2$ archi, si ha (ovviamente se $\varepsilon < 1/(n-1)$)

$$\bar{x}(S) \leq \frac{|S|-1}{n-1} + \varepsilon \frac{(|S|-1)(|S|-2)}{2} < \frac{|S|-1}{2}$$

per $n > 3$ e per ε sufficientemente piccolo.

c) Dato un insieme dispari S di nodi, sia $\bar{x}_e := 1/|S|$ per $e \in E(S)$ e $\bar{x}_e := \varepsilon > 0$ per ogni altro arco. Allora $\bar{x}(S) = (|S|-1)/2$. Per le disequaglianze di tipo b) si ha

$$x(Q(\{i\})) \leq \frac{|S|-1}{|S|} + \varepsilon(n-|S|) < 1$$

per ε sufficientemente piccolo. Per le disequaglianze di tipo c) sia $T \neq S$ un insieme dispari di nodi e sia $R := S \cap T$. Si noti che almeno una delle due disequaglianze $|R| \leq |S|$ e $|R| \leq |T|$ è stretta dato che $|R| = |S| = |T|$ implica $S = T$. Allora si vede che per ε sufficientemente piccolo si ha

$$x(T) \leq \frac{|R|(|R|-1)}{2|S|} + \varepsilon \left(\frac{|T|(|T|-1)}{2} - \frac{|R|(|R|-1)}{2} \right) < \frac{|T|-1}{2}$$

■

10.25 ESERCIZIO. Per il grafo $G = (N, E)$ con $|N| = 4$ e $E = \{(1, 2), (1, 3), (2, 3), (3, 4)\}$ si trovi quali disequaglianze di $Q_M^1(G)$ sono ridondanti. La dimensione di $Q_M^1(G)$ è 4. Per ogni faccetta si indichi quali accoppiamenti stanno sulla faccetta. Si elenchino le facce di dimensione 2 e gli spigoli. Individuati gli spigoli si disegni una proiezione bidimensionale dei vertici e degli spigoli.

■

Sia $\mathcal{S}' \subset \mathcal{S}$ un sottoinsieme di insiemi dispari di nodi e si definisca il poliedro

$$Q' := \{x \in \mathbb{R}^m : Ax = \mathbf{1}, x(S) \leq (|S|-1)/2, S \in \mathcal{S}', x \geq 0\}$$

I vertici di Q' hanno coordinate frazionarie in generale in quanto 'mancano' alcuni piani di taglio. Si può pensare tuttavia di risolvere con il metodo del simplesso il problema

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = \mathbf{1} \\ & x(S) \leq (|S|-1)/2 \quad \forall S \in \mathcal{S}' \\ & x \geq 0 \end{aligned} \tag{10.3}$$

dove $\mathcal{S}' \subset \mathcal{S}$ è un opportuno sottoinsieme generato esplicitamente. Se la soluzione di (10.3) è intera, si tratta dell'ottimo del problema di accoppiamento. Altrimenti esiste fra le disequaglianze $x(S) \leq (|S|-1)/2$, $S \notin \mathcal{S}'$, una che è 'violata' dalla soluzione frazionaria.

Il problema della separazione per il poliedro dell'accoppiamento perfetto è stato studiato da Padberg e Rao [1982] che hanno fornito un algoritmo polinomiale di separazione di complessità $O(n^4)$. Bisogna tuttavia dire che nella maggior parte dei casi i vertici di Q' hanno coordinate in $\{0, 1/2, 1\}$ e in questo caso è immediato individuare un circuito dispari di valori $1/2$ e da questo l'insieme S . Altrimenti, per individuare S , si usa la procedura di Padberg e Rao, computazionalmente più onerosa.

A questo punto si aggiorna S' aggiungendo S a S' e si risolve nuovamente (10.3) (con il semplice duale partendo dal precedente ottimo). Si procede nello stesso modo finché non si trova una soluzione intera. Inizialmente si pone $S' = \emptyset$.

Questo metodo è stato proposto da Grötschel e Holland [1983] insieme con tutta una serie di accorgimenti computazionali per poter risolvere istanze molto grandi. Si tratta di un metodo, ovviamente senza garanzie di complessità polinomiale, ma altamente raccomandabile per la maggior semplicità di implementazione rispetto all'algoritmo di Edmonds (vedi più avanti) e anche per i buoni risultati computazionali verificati sperimentalmente.

10.3. L'algoritmo ungherese

Si è appena visto che il problema dell'accoppiamento bipartito pesato può essere risolto tramite un semplice problema di programmazione lineare. Inoltre un algoritmo polinomiale per il problema di complessità $O(n^3)$ è stato presentato nella sezione 8.10. Questo algoritmo non è quello che storicamente fu proposto per primo da Kuhn [1955]. Il risultato di Kuhn all'epoca fu notevole, tenendo conto che, nonostante l'incredibile numero di soluzioni alternative ($n!$), si riusciva a trovare l'ottimo addirittura con metodi manuali e in tempi ragionevoli. La tecnica adottata da Kuhn può essere descritta come una tecnica primale-duale, cioè un metodo di massima ascesa per il problema duale. Siccome venivano usate proprietà dei grafi bipartiti che erano state introdotte da matematici ungheresi, tale algoritmo divenne noto come *algoritmo ungherese*.

L'esposizione dell'algoritmo ungherese viene fatta seguendo i concetti esposti nella sezione 7.13. Possiamo quindi evidenziare la seguente coppia di problemi primale-duale

$$\begin{array}{ll} \min & cx \\ & Ax = \mathbf{1} \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \max & \sum_{i=1}^n u_i^- + u_i^+ \\ & u_i^- + u_j^+ \leq c_{ij} \quad (i, j) \in E \end{array}$$

Sia definito il seguente insieme di *archi ammissibili*:

$$E(u) := \{(ij) : u_i^- + u_j^+ = c_{ij}\}$$

e sia $G(u) := (N^-, N^+; E(u))$. Sfruttando i risultati della sezione 7.13, la direzione di massima ascesa nel duale si ottiene dal problema primale ristretto

$$\begin{array}{ll} \max & \sum_{e \in E(u)} x_e \\ & \sum_{e \in E(u) \cap Q(\{i\})} x_e \leq \mathbf{1} \quad \forall i \in N^- \cup N^+ \\ & x_e \geq 0 \quad \forall e \in E(u) \end{array} \quad (10.4)$$

In particolare, se \bar{h}_i è l'ottimo duale di (10.4), la direzione di massima ascesa è $h_i := 1 - 2\bar{h}_i$. Il problema (10.4) è un problema di accoppiamento di massima cardinalità sul grafo $G(u)$.

Per determinare le variabili duali ottime di (10.4) conviene far ricorso al metodo che risolve il problema di massimo accoppiamento come un problema di massimo flusso.

In corrispondenza del massimo flusso che dà luogo al massimo accoppiamento in $G(u)$, sia $S(u) \subset N^- \cup N^+$ l'insieme di nodi di cardinalità minima che genera il taglio di minima capacità. Tale insieme è univocamente determinato e si trova con una ricerca di cammini aumentanti (nel senso del massimo flusso) sulla rete residua. Possiamo dire che i nodi in $S(u)$ sono 'raggiungibili'. Possiamo dimostrare allora che:

10.26 TEOREMA. *L'ottimo duale di (10.4) è dato dai seguenti valori*

$$\bar{h}_i^- := \begin{cases} 0 & \text{se } i \in S(u) \\ 1 & \text{se } i \notin S(u) \end{cases} \quad \bar{h}_i^+ := \begin{cases} 1 & \text{se } i \in S(u) \\ 0 & \text{se } i \notin S(u) \end{cases}$$

DIMOSTRAZIONE. Il teorema verrà dimostrato facendo dapprima vedere che i suddetti valori sono ammissibili e poi che valori primali e duali coincidono.

Per ciò che riguarda l'ammissibilità ($\bar{h}_i^- + \bar{h}_j^+ \geq 1, \forall (ij) \in E(u)$) si tratta di far vedere che il caso $(ij) \in E(u), i \in S(u)$ e $j \notin S(u)$ non può mai verificarsi. Infatti se (ij) risultasse libero, j sarebbe raggiungibile da $i \in S(u)$, mentre se risultasse accoppiato, i sarebbe raggiungibile solo da j .

Per dimostrare l'ottimalità si noti che in ogni arco accoppiato i due nodi devono essere entrambi in $S(u)$ oppure nessuno dei due, per il motivo visto nel precedente paragrafo. Inoltre un nodo esposto in N^- deve stare in $S(u)$, mentre un nodo esposto in N^+ non può stare in $S(u)$ altrimenti l'accoppiamento non sarebbe massimo. Quindi l'espressione $\sum_i h_i^- + h_j^+$ (funzione obiettivo duale) non è altro che il numero di archi accoppiati. ■

Abbiamo allora che la direzione $h = 1 - 2\bar{h}$ di massima ascesa è data da

$$h_i^- := \begin{cases} +1 & \text{se } i \in S(u) \\ -1 & \text{se } i \notin S(u) \end{cases} \quad h_i^+ := \begin{cases} -1 & \text{se } i \in S(u) \\ +1 & \text{se } i \notin S(u) \end{cases}$$

La determinazione della lunghezza del passo viene fatta direttamente calcolando il prossimo punto di rottura come il massimo valore di α per cui

$$u_i^- + \alpha h_i^- + u_j^+ + \alpha h_j^+ \leq c_{ij} \quad \forall (ij)$$

Siccome, come già verificato, non può essere $h_i^- = h_j^+ = 1$ se $(ij) \in E(u)$, basta verificare le disequazioni per $(ij) \notin E(u)$ e in particolare solo per quegli indici per cui $h_i^- = h_j^+ = 1$. Il passo è pertanto dato da

$$\alpha := \min \left\{ \frac{c_{ij} - u_i^- - u_j^+}{2} : (ij) \notin E(u), i \in S(u), j \notin S(u) \right\} \quad (10.5)$$

Sia (lk) un arco per cui viene realizzato il minimo. Quindi si ha $(lk) \in E(u + \alpha h)$ e viene aggiunto un arco a $G(u)$. Il calcolo dell'accoppiamento massimo in $G(u + \alpha h)$ si può fare direttamente a partire dall'accoppiamento massimo in $G(u)$. Infatti la ricerca sul grafo può ripartire da l e proseguire attraverso il nuovo arco (lk) fino al nodo k . A questo punto si possono verificare due casi: o k è esposto e quindi si è trovato un cammino aumentante, oppure k è accoppiato con un nodo di N^- , che indichiamo come m . Una semplice verifica fa vedere che $(m, k) \in E(u)$ e quindi $S^- := S^- \cup \{m\}$ e $S^+ := S^+ \cup \{k\}$ da cui si ottiene una nuova direzione di ascesa.

Abbiamo perciò individuato due tipi di soluzione del primale ristretto: una fase aumentante, quando si individua un accoppiamento aumentato dal precedente, e una fase di ricerca, quando si continua la ricerca. È chiaro che tale metodo è finito: infatti non vi possono essere più di n fasi aumentanti e fra due fasi aumentanti non vi possono essere più di n fasi di ricerca. Si hanno perciò $O(n^2)$ fasi di ricerca.

In ogni fase di ricerca viene inoltre richiesto il calcolo di (10.5). Se fatto esplicitamente ad ogni iterazione comporterebbe un onere computazionale eccessivo. Convienne invece procedere nel seguente modo: sia

$$\sigma_j(u) := \min\{c_{ij} - u_i^- - u_j^+ : (ij) \notin E(u), i \in S(u)\} \quad (10.6)$$

Pertanto $\alpha := 1/2 \sigma_k(u) := 1/2 \min_{j \notin S(u)} \sigma_j(u)$. Sia ora (lk) l'arco che entra in $G(u + \alpha h)$. Supponendo k non esposto, altrimenti si darebbe luogo ad un aumento di accoppiamento, mentre ora stiamo considerando fasi di ricerca, la ricerca si propaga da k all'indietro su un arco accoppiato, ad esempio (m, k) , raggiungendo $m \in N^-$. Quindi si può scrivere, indicando $\bar{u} := u + \alpha h$:

$$\begin{aligned} \sigma_j(\bar{u}) &:= \min\{c_{ij} - \bar{u}_i^- - \bar{u}_j^+ : (ij) \notin E(\bar{u}), i \in S(\bar{u})\} = \\ &\min\{\min\{c_{ij} - \bar{u}_i^- - \bar{u}_j^+ : (ij) \notin E(u), i \in S(u)\} ; c_{mj} - \bar{u}_m^- - \bar{u}_j^+\} = \\ &\min\{\min\{c_{ij} - u_i^- - u_j^+ - 2\alpha : (ij) \notin E(u), i \in S(u)\} ; c_{mj} - u_m^- - u_j^+\} = \\ &\min\{\sigma_j(u) - \sigma_k(u) ; c_{mj} - u_m^- - u_j^+\} \end{aligned} \quad (10.7)$$

Si tenga presente che si è posto $\bar{u}_j^+ := u_j^+ + \alpha$ perché poi si calcola il minimo di $\sigma_j(\bar{u})$ soltanto per $j \notin S$. Con questo accorgimento il calcolo di α ha un costo pari a $O(n)$ per ogni nuovo nodo raggiunto. Le fasi di ricerca hanno pertanto un costo complessivo di $O(n^3)$. Inoltre ogni fase aumentante richiede un calcolo ex novo delle quantità σ_j e quindi si ha un costo globale per le fasi aumentanti di $O(n^3)$. L'algoritmo ha pertanto complessivamente un costo $O(n^3)$.

Si sarà certamente notata la rassomiglianza delle espressioni scritte con quelle dell'algoritmo di Dijkstra per il cammino minimo. Non si tratta di un fatto casuale. Il metodo delineato calcola, per ogni fase aumentante, proprio il cammino minimo rispetto ai costi ridotti $\bar{c}_{ij} := c_{ij} - u_i^- - u_j^+$, dove u è il valore iniziale della fase aumentante. Ricordiamo che l'algoritmo di Dijkstra si basa sul calcolo $\rho_k := \min \rho_j$ e sull'aggiornamento $\rho'_j := \min\{\rho_j ; \rho_k + \bar{c}_{kj}\}$. Basta far vedere che $\rho_j = \sigma_j + \Delta$ per ogni fase di aggiornamento dei valori, per cui il calcolo di $\min \sigma_j$ produce lo stesso risultato di $\min \rho_j$. In base a (10.7) si ha la formula di aggiornamento per i valori σ_j

$$\sigma'_j := \min\{\sigma_j - \sigma_k ; \bar{c}_{mj}\} \quad (10.8)$$

Supponiamo che, prima dell'aggiornamento, sia vero $\rho_j = \sigma_j + \Delta$. Allora da (10.8)

$$\sigma'_j = \min\{\sigma_j + \Delta ; \bar{c}_{mj} + \sigma_k + \Delta\} - \sigma_k - \Delta = \min\{\rho_j ; \bar{c}_{mj} + \rho_k\} - \sigma_k - \Delta$$

Tenendo presente che $\bar{c}_{mk} = 0$ si ha $\rho_m = \rho_k$ e ponendo $\Delta' = \Delta + \sigma_k$ (σ_k è un valore costante all'interno di una fase di ricerca) si ha

$$\sigma'_j + \Delta' = \min\{\rho_j ; \bar{c}_{mj} + \rho_m\} = \rho'_j$$

e quindi la relazione $\rho_j = \sigma_j + \Delta$ è vera anche dopo l'aggiornamento. Siccome è vera anche inizialmente (facile verifica), è vera sempre.

Algoritmo ungherese

```

input( $c$ );
 $u^- := 0$ ;  $u^+ := 0$ ;  $M := \emptyset$ ;
for  $j := 1$  to  $n$   $\sigma_j := \min_i c_{ij}$ ;
 $T^- := \{1, \dots, n\}$ ;  $T^+ := \{1, \dots, n\}$ ;
(*  $T^-$  e  $T^+$  sono gli insiemi dei nodi esposti*)
while  $|T^-| > 0$  do
  begin
     $S^- := T^-$ ;  $S^+ := \emptyset$ ;
    for  $j \notin S^+$  do  $\sigma_j := \min \{c_{ij} - u_i^- - u_j^+ : i \in S^-\}$ ;
    while  $S^+ \cap T^+ = \emptyset$  do
      begin
         $k := \operatorname{argmin}_{j \notin S^+} \sigma_j$ ;
         $S^+ := S^+ \cup \{k\}$ ;
        if  $k \notin T^+$  then  $S^- := S^- \cup \{m\}$ ; (* $m = \operatorname{mate}(k)$ *)
        calcola  $h^-$ ,  $h^+$ ,  $\alpha := \sigma_k/2$ ;
        for  $j \notin S^+$  do  $\sigma_j := \min \{\sigma_j - \sigma_k ; c_{mj} - u_m^- - u_j^+\}$ ;
         $u := u + \alpha h$ 
      end;
      ricostruisci cammino aumentante  $h \rightsquigarrow k$ ;
      aggiorna  $M$ ;
       $T^- := T^- \setminus h$ ;  $T^+ := T^+ \setminus k$ ;
    end
  end
output( $M$ ).

```

È opportuno far rilevare che nel caso il minimo di α venga realizzato da più di un arco, non è strettamente necessario inserire in $G(u)$ tutti gli archi che realizzano il minimo. È sufficiente inserirne uno qualsiasi. Se l'algoritmo, per determinare successive direzioni di ascesa, avesse bisogno anche degli altri archi, si verificherebbe una fase di ricerca con lunghezza di passo nulla, assimilabile pertanto ad una iterazione con soluzione di base degenerare. Una tale scelta non pregiudica il funzionamento dell'algoritmo e ne rende invece più semplice la scrittura.

L'inizializzazione delle variabili duali viene fatta ponendo $u_i^- := 0, \forall i$, e $u_j^+ := \min_i c_{ij}, \forall j$. In questo modo il grafo iniziale $G(u)$ è dato da quegli archi per cui si realizza l'indicato minimo. Anche in questo caso è opportuno inserire in $G(u)$ solo un arco per ogni nodo $j \in N^+$. Questa scelta dà anche un metodo immediato di calcolo dell'accoppiamento massimo nel primo $G(u)$: basta scandire i nodi $j \in N^+$ e in corrispondenza dell'arco (i, j) che realizza il minimo di c_{ij} accoppiare i e j se i non risulta già accoppiato. Si può provare facilmente che l'accoppiamento così ottenuto è massimo in $G(u)$.

10.27 ESEMPIO. Sia $|N^-| = |N^+| = 3$, con i seguenti costi

$$c := \begin{pmatrix} 64 & 20 & 36 \\ 48 & 48 & 28 \\ 96 & 56 & 98 \end{pmatrix}$$

L'algoritmo è inizializzato con $u^- := (0, 0, 0)$, $u^+ := (0, 0, 0)$, $\sigma = (48, 20, 28)$. Con questi valori di u si ha $E(u) = \emptyset$, $S^-(u) = \{1, 2, 3\}$, $S^+(u) = \emptyset$, da cui $h^- = (1, 1, 1)$, $h^+ = (1, 1, 1)$, $k = 2$, $\alpha = 10$.

Quindi il nuovo valore di $u := u + \alpha h$ è $u^- = (10, 10, 10)$, $u^+ = (10, 10, 10)$ da cui $E(u) = \{(1, 2)\}$. L'arco $(1, 2)$ può essere accoppiato e il massimo flusso corrispondente fornisce $S^-(u) = \{2, 3\}$, $S^+(u) = \emptyset$ da cui $h^- = (-1, 1, 1)$, $h^+ = (1, 1, 1)$. I costi ridotti e i valori σ (da ricalcolare in base a (10.6)) sono

$$\bar{c} = \begin{pmatrix} 44 & 0 & 16 \\ 28 & 28 & 8 \\ 76 & 36 & 78 \end{pmatrix} \quad \sigma = (28, 28, 8)$$

e quindi $k = 3$, $\alpha = 4$.

Il nuovo valore di u è $u^- = (6, 14, 14)$, $u^+ = (14, 14, 14)$ da cui $E(u) = \{(1, 2), (2, 3)\}$. Anche l'arco $(2, 3)$ può essere accoppiato e il massimo flusso corrispondente fornisce $S^-(u) = \{3\}$, $S^+(u) = \emptyset$. Con questi valori si ottiene $h^- = (-1, -1, 1)$, $h^+ = (1, 1, 1)$. I costi ridotti e i valori σ sono

$$\bar{c} = \begin{pmatrix} 44 & 0 & 16 \\ 20 & 20 & 0 \\ 68 & 28 & 70 \end{pmatrix} \quad \sigma = (68, 28, 70)$$

e quindi $k = 2$, $\alpha = 14$.

Si calcola $u^- = (-8, 0, 28)$, $u^+ = (28, 28, 28)$, da cui $E(u) = \{(1, 2), (2, 3), (3, 2)\}$ (si veda $E(u)$ in figura 10.21-(a)). L'algoritmo di ricerca può così raggiungere il nodo 2 di destra e, all'indietro attraverso l'arco accoppiato $(1, 2)$, il nodo 1 di sinistra. Allora $S^-(u) = \{1, 3\}$, $S^+(u) = \{2\}$ e $h^- = (1, -1, 1)$, $h^+ = (1, -1, 1)$. Per il calcolo del passo bisogna aggiornare σ secondo la formula (10.7). Allora $(m, k) = (1, 2)$ e la formula (10.7) diventa, calcolata solo per $j \notin S^+(u)$ e cioè per $j \in \{1, 3\}$

$$\sigma_1 := \min \{68 - 28 ; 64 + 8 - 28\} = 40$$

$$\sigma_3 := \min \{70 - 28 ; 36 + 8 - 28\} = 16$$

da cui $k = 3$ e $\alpha = 8$. Si ricalcola u ottenendo $u^- = (0, -8, 36)$, $u^+ = (36, 20, 36)$, da cui $E(u) = \{(1, 2), (2, 3), (3, 2), (1, 3)\}$ (si veda $E(u)$ in figura 10.21-(b)). Questa volta l'algoritmo di ricerca raggiunge il nodo 3 di destra e il nodo 2 di sinistra tramite l'arco accoppiato $(2, 3)$ (quindi $(m, k) = (2, 3)$). Allora $S^-(u) = \{1, 2, 3\}$, $S^+(u) = \{2, 3\}$, $h^- = (1, 1, 1)$, $h^+ = (1, -1, -1)$. L'aggiornamento di σ_1 dà

$$\sigma_1 := \min \{40 - 16 ; 48 + 8 - 36\} = 20$$

da cui $\alpha = 10$, $u^- = (10, 2, 46)$, $u^+ = (46, 10, 26)$ e $E(u) = \{(1, 2), (2, 3), (3, 2), (1, 3), (2, 1)\}$ (si veda $E(u)$ in figura 10.21-(c)). Si è finalmente trovato un cammino aumentante che produce un accoppiamento perfetto e quindi l'accoppiamento ottimo è quello in figura 10.21-(d). Si può verificare la dualità forte: $\sum_i u_i^- + u_i^+ = 140$ che è anche il costo dell'accoppiamento ottimo $c_{13} + c_{21} + c_{32} = 36 + 48 + 56 = 140$. Verifichiamo inoltre che i costi ridotti sono non negativi:

$$\bar{c} = \begin{pmatrix} 8 & 0 & 0 \\ 0 & 36 & 0 \\ 4 & 0 & 26 \end{pmatrix}$$

■

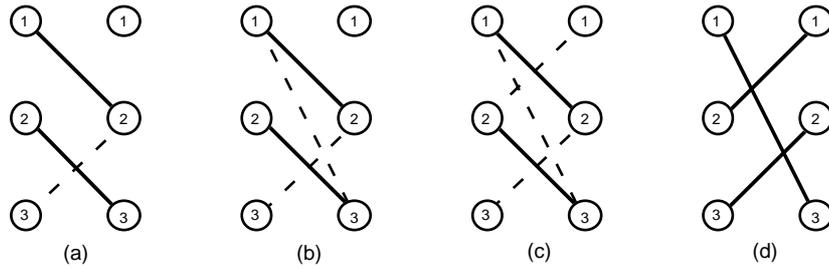


FIGURA 10.21

10.4. Accoppiamento non bipartito pesato

Anche il problema dell'accoppiamento pesato generico viene risolto con un metodo primale-duale. Tuttavia il metodo risulta in questo caso molto più complicato a causa della presenza dei vincoli sugli insiemi dispari di nodi. Come già visto precedentemente il problema primale da risolvere è, in forma standard:

$$\begin{aligned} \min \quad & cx \\ & Ax = \mathbf{1} \\ & Dx + y = d \\ & x \geq 0, y \geq 0 \end{aligned}$$

il cui duale è

$$\begin{aligned} \max \quad & \sum_{i \in N} u_i + \sum_{S \in \mathcal{S}} d_S w_S \\ & u_i + u_j + \sum_{S \ni i, j} w_S \leq c_{ij} \quad \forall i, j \in N \\ & w_S \leq 0 \quad \forall S \in \mathcal{S} \end{aligned}$$

In corrispondenza di una generica variabile duale (u, w) si definisca il seguente insieme di *archi ammissibili*:

$$E(u, w) := \{(ij) : u_i + u_j + \sum_{S \ni i, j} w_S = c_{ij}\}$$

e il seguente insieme di *insiemi ammissibili*:

$$\mathcal{S}(w) := \{S : w_S = 0\}.$$

Applicando i risultati della sezione 7.13, il problema primale ristretto (7.35) diventa

$$\begin{aligned} \min \quad & \sum_{i \in N} \gamma_i + 2 \sum_{S \notin \mathcal{S}(w)} \eta_S \\ & \sum_{e \in E(v, w)} \sum_{e \in \delta(i)} x_e + \gamma_i = 1 \quad i \in N \\ & \sum_{e \in E(v, w)} \sum_{e \in E(S)} x_e + \eta_S = d_S \quad S \notin \mathcal{S}(w) \\ & \sum_{e \in E(v, w)} \sum_{e \in E(S)} x_e + y_S = d_S \quad S \in \mathcal{S}(w) \\ & x_e \geq 0, y_S \geq 0, \gamma_i \geq 0, \eta_S \geq 0, \end{aligned} \tag{10.9}$$

Si noti che, rispetto a (7.35), compare il coefficiente 2 per le variabili η_S . Questo si può fare senza perdita di generalità in quanto corrisponde semplicemente ad una normalizzazione diversa delle variabili duali. Inoltre le variabili η_S non compaiono nei vincoli relativi agli insiemi in $\mathcal{S}(w)$. È immediato notare che tali variabili sono sempre nulle in ottimalità (il loro valore può essere assunto da y_S) e quindi è inutile introdurle. Il duale di (10.9) è

$$\begin{aligned} \max \quad & \sum_{i \in N} h_i + \sum_{S \in \mathcal{S}} d_S g_S \\ & h_i + h_j + \sum_{S \ni i, j} g_S \leq 0 \quad \forall (ij) \in E(u, w) \\ & h_i \leq 1 \quad \forall i \in N \\ & g_S \leq 2 \quad \forall S \notin \mathcal{S}(w) \\ & g_S \leq 0 \quad \forall S \in \mathcal{S}(w) \end{aligned} \quad (10.10)$$

Per capire la natura del primale ristretto si impieghi la formulazione (7.37) che diventa in questo caso (oppure anche operando direttamente su (10.9) per sostituzione delle variabili γ_i e η_S)

$$\begin{aligned} \max \quad & \sum_{e \in E(v, w)} (2 + 2K_e(w)) x_e \\ & \sum_{e \in E(v, w)} \sum_{e \in Q(i)} x_e \leq 1 \quad i \in N \\ & \sum_{e \in E(v, w)} \sum_{e \in E(S)} x_e \leq d_S \quad S \in \mathcal{S} \end{aligned} \quad (10.11)$$

dove $K_e(w)$ è il numero di sottoinsiemi dispari non ammissibili che contengono l'arco e . Se non fossero presenti i termini $K_e(w)$ dipendenti dall'arco, il problema (10.11) sarebbe un problema di accoppiamento di cardinalità sul grafo $(N, E(v, w))$. Però, se si riesce a fare in modo che gli insiemi non ammissibili siano i germogli che vengono generati nella ricerca dei cammini aumentanti, e questi vengono contratti, allora non ci sono insiemi non ammissibili nel grafo contratto e per questo grafo (10.11) rappresenta proprio un problema di cardinalità. Questa è l'idea di fondo che sta alla base dell'algoritmo proposto da Edmonds.

Il problema che si pone allora nella risoluzione del primale ristretto (10.9) è quello di tenere sotto controllo l'evoluzione dell'insieme $\mathcal{S}(w)$ in modo che gli insiemi non ammissibili siano pochi e di struttura particolare. Per ottenere ciò il metodo procede mantenendo costantemente valide per il problema primale le seguenti ipotesi:

- 1) $S \notin \mathcal{S}(w)$ implica che S ha il massimo numero di archi accoppiati, cioè è *pieno*;
- 2) $S, T \notin \mathcal{S}(w)$ implica che $S \subset T$ oppure $S \supset T$ oppure $S \cap T = \emptyset$.

L'ipotesi 1) serve a far sì che in ogni soluzione di (10.9) si abbia $\eta_S = 0$ e quindi il valore ottimo di (10.9) rappresenti il numero di nodi esposti di un accoppiamento massimo di $G(u, w)$.

L'ipotesi 2) permette, come vedremo, di contrarre ogni S , $S \notin \mathcal{S}(w)$, in uno pseudonodo e, nel caso più pseudonodi si contraggano ulteriormente in un nuovo pseudonodo, di tener conto solo di quest'ultimo.

Il seguente teorema fornisce la soluzione di (10.10) sfruttando l'algoritmo che trova la soluzione ottima dell'accoppiamento massimo di $G(u, w)$. Gli insiemi non ammissibili sono identificati con i germogli determinati dalla ricerca di cammini aumentanti. Si indichi pertanto con $\tilde{G}(u, w)$ il grafo finale ottenuto attraverso successive contrazioni di germogli. Grazie all'ipotesi che ogni possibile germoglio generato da cammini alternanti è stato contratto e

che non esistono cammini aumentanti perché l'accoppiamento è massimo, ogni nodo si trova in modo non ambiguo in uno dei seguenti stati: esposto, interno in un cammino alternante, esterno in un cammino alternante, nessuno di questi (si veda un esempio in figura 10.22). Si lascia come facile esercizio questa verifica. Inoltre tutti i nodi vengono ripartiti in tanti sottoinsiemi quanti sono i nodi esposti più un sottoinsieme di nodi irraggiungibili. Ognuno dei sottoinsiemi collegato con i nodi esposti è costituito dai nodi raggiungibili dal nodo esposto con cammini alternanti. L'enunciato del teorema si basa sullo stato di ogni nodo in un accoppiamento massimo.

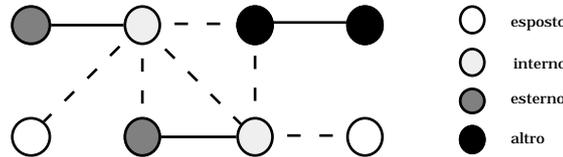


FIGURA 10.22

10.28 TEOREMA. Si determini l'accoppiamento massimo di $G(u, w)$ con l'algoritmo dei cammini aumentanti e sia $\tilde{G}(u, w)$ il grafo contratto finale. Allora l'ottimo di (10.10) è dato dai seguenti valori

$$h_i := \begin{cases} +1 & \text{se } i \text{ è un nodo (o fa parte di uno pseudonodo) esposto o esterno} \\ & \text{in un cammino alternante di } \tilde{G}(u, w); \\ -1 & \text{se } i \text{ è un nodo (o fa parte di uno pseudonodo) interno} \\ & \text{in un cammino alternante di } \tilde{G}(u, w); \\ 0 & \text{altrimenti} \end{cases}$$

$$g_S := \begin{cases} -2 & \text{se } S \notin \mathcal{S}(w) \text{ e } S \text{ è un insieme massimale corrispondente} \\ & \text{ad uno pseudonodo di } \tilde{G}(u, w) \text{ esposto o esterno in un cammino alternante} \\ +2 & \text{se } S \notin \mathcal{S}(w) \text{ e } S \text{ è un insieme massimale corrispondente} \\ & \text{ad uno pseudonodo di } \tilde{G}(u, w) \text{ interno in un cammino alternante} \\ 0 & \text{altrimenti} \end{cases}$$

DIMOSTRAZIONE. La dimostrazione verrà fatta analogamente al caso dell'accoppiamento bipartito facendo vedere che i valori indicati sono ammissibili in (10.10) e che la funzione obiettivo di (10.10) calcolata con questi valori è pari al valore ottimo di (10.9), cioè al numero di nodi esposti nell'accoppiamento massimo di $G(u, w)$.

Per ciò che riguarda l'ammissibilità si noti che l'unico vincolo da verificare è

$$h_i + h_j + \sum_{S \ni i, j} g_S \leq 0 \quad \forall (i, j) \in E(u, w)$$

in quanto gli altri sono immediati. Si noti che, in base all'ipotesi 2), nella somma $\sum_{S \ni i, j} g_S$ vi è al più un termine non nullo, quello relativo all'insieme massimale che contiene l'arco (i, j) . Consideriamo i vari casi:

- h_i e h_j hanno valori diversi: allora i e j non possono appartenere al medesimo insieme massimale S e quindi il vincolo si riduce a $h_i + h_j \leq 0$. L'unico caso critico sarebbe $h_i = 0$ e $h_j = 1$ (o viceversa). Ma questo non si può verificare, perché $h_j = 1$ implicherebbe che j sia nodo esposto o esterno e quindi i sarebbe raggiungibile con un cammino alternante contraddicendo $h_i = 0$;

- $h_i = h_j = -1$: il vincolo è banamente soddisfatto;

- $h_i = h_j = 0$: entrambi i nodi non sono raggiungibili. Se appartengono al medesimo pseudonodo neppure questo può essere raggiungibile e quindi $g_S = 0$;

– $h_i = h_j = 1$: non vi può essere un arco fra due nodi esposti oppure fra due nodi esterni di cammini alternanti diversi, perché l'accoppiamento non sarebbe massimo. Neppure vi può essere un arco fra due nodi esterni (o uno esposto ed uno esterno) dello stesso cammino alternante, perché questo implicherebbe l'esistenza di un germoglio. Quindi i e j appartengono ad un insieme massimale S contratto in uno pseudonodo esterno per il quale si ha necessariamente $g_S = -2$ e quindi il vincolo è soddisfatto.

Per ciò che riguarda la funzione obiettivo, la sommatoria $\sum_{S \in \mathcal{S}} d_S g_S$ può essere limitata agli insiemi non ammissibili massimali interni, esterni o esposti, ovvero agli pseudonodi finali interni, esterni o esposti. Sia \mathcal{S}^- l'insieme degli insiemi non ammissibili massimali interni, \mathcal{S}^+ l'insieme degli insiemi non ammissibili massimali esterni e sia \mathcal{S}^0 l'insieme degli insiemi non ammissibili massimali esposti. Sia N^- l'insieme dei nodi (non pseudonodi) interni in $\tilde{G}(u, w)$, sia N^+ l'insieme dei nodi esterni e sia N^0 l'insieme dei nodi esposti. La sommatoria $\sum_{i \in N} h_i$ può essere riscritta, eliminando i termini nulli,

$$\sum_{i \in N^0} h_i + \sum_{i \in N^-} h_i + \sum_{i \in N^+} h_i + \sum_{S \in \mathcal{S}^0} \sum_{i \in S} h_i + \sum_{S \in \mathcal{S}^-} \sum_{i \in S} h_i + \sum_{S \in \mathcal{S}^+} \sum_{i \in S} h_i$$

Quindi la funzione obiettivo diventa

$$\begin{aligned} & \sum_{i \in N^0} h_i + \sum_{i \in N^-} h_i + \sum_{i \in N^+} h_i + \\ & + \sum_{S \in \mathcal{S}^0} (d_S g_S + \sum_{i \in S} h_i) + \sum_{S \in \mathcal{S}^-} (d_S g_S + \sum_{i \in S} h_i) + \sum_{S \in \mathcal{S}^+} (d_S g_S + \sum_{i \in S} h_i) \end{aligned} \quad (10.12)$$

È semplice verificare che

$$d_S g_S + \sum_{i \in S} h_i = \begin{cases} -1 & \text{se } S \in \mathcal{S}^- \\ +1 & \text{se } S \in \mathcal{S}^+ \cup \mathcal{S}^0 \end{cases}$$

Siccome il numero di nodi e pseudonodi interni in $\tilde{G}(u, w)$ è uguale a quello degli esterni l'espressione (10.12) si riduce a $\sum_{i \in N^0} 1 + \sum_{S \in \mathcal{S}^0} 1$, cioè al numero di nodi esposti di $\tilde{G}(u, w)$ che sono gli stessi di $G(u, w)$.

L'accoppiamento massimo in $G(u, w)$ trovato in questo modo è una soluzione ammissibile di (10.9) in cui, come già osservato, $\eta_S = 0$ e quindi il valore corrispondente della funzione obiettivo è uguale a $\sum_{i \in N} \gamma_i$ cioè al numero di nodi esposti in $G(u, w)$. L'uguaglianza di obiettivo fra problema primale e duale implica l'ottimalità. ■

Il teorema pertanto fornisce una direzione di massima ascesa. La determinazione del passo viene fatta calcolando il prossimo punto di rottura. Si vede dai vincoli di (10.10) che il passo α deve soddisfare:

$$\begin{aligned} u_i + u_j + \alpha &\leq c_{ij} && \text{se } h_i = 0, h_j = 1 \text{ e } (ij) \notin E(u, w) \\ u_i + \alpha + u_j + \alpha &\leq c_{ij} && \text{se } h_i = 1, h_j = 1 \text{ e } (ij) \notin E(u, w) \\ w_S + 2\alpha &\leq 0 && \text{se } S \notin \mathcal{S}(w) \text{ e } S \text{ corrisponde ad uno pseudonodo interno} \end{aligned}$$

Quindi, indicando

$$\begin{aligned} \alpha_1 &:= \min \{c_{ij} - u_i - u_j : h_i = 0, h_j = 1\} \\ \alpha_2 &:= \min \left\{ \frac{c_{ij} - u_i - u_j}{2} : h_i = 1, h_j = 1 \right\} \\ \alpha_3 &:= \min \left\{ -\frac{w_S}{2} : S \text{ pseudonodo interno} \right\} \end{aligned}$$

si ha

$$\alpha := \min\{\alpha_1, \alpha_2, \alpha_3\}.$$

Anziché iniziare dal valore $u = 0$ conviene calcolare i valori iniziali per le variabili duali nel seguente modo:

$$u := 0; \text{ for } i := 1 \text{ to } n \text{ do } u_i := \min_j c_{ij} - u_j \quad (10.13)$$

In base a questi valori viene costruito il grafo $G(u, w)$ che inizialmente non contiene pseudonodi. Si calcola l'accoppiamento massimo di $G(u, w)$ e, se non ci sono nodi esposti, tale accoppiamento è l'ottimo del problema. Altrimenti si determina h come delineato dal teorema e dai valori di h si calcola la lunghezza del passo. Se il punto di rottura viene determinato da α_1 o α_2 almeno un nuovo arco viene aggiunto a $G(u, w)$ e contemporaneamente alcuni archi possono diventare non ammissibili ovvero essere tolti da $G(u, w)$. Esaminiamo prima il caso degli archi nuovi.

Se avviene $\alpha = \alpha_1$ il nuovo arco congiunge un cammino alternante (oppure un nodo esposto) con una componente connessa senza nodi esposti ed è quindi su questa componente che prosegue la ricerca di un cammino aumentante. Questa fase viene detta *fase di ricerca*. In questa fase l'algoritmo di ricerca può scoprire dei germogli che vengono contratti in pseudonodi; si noti che uno pseudonodo, quando viene creato, è un nodo esterno in un cammino alternante. Se si genera un germoglio la fase di ricerca viene interrotta per dar luogo ad una *fase di contrazione* dopo la quale riprende la fase di ricerca.

Se avviene $\alpha = \alpha_2$ possono succedere due casi: se il nuovo arco congiunge due cammini alternanti tramite due nodi esterni o esposti si ottiene un cammino aumentante e l'accoppiamento corrente viene pertanto aumentato. Questa fase viene detta *fase aumentante*. Se invece il nuovo arco congiunge due nodi esterni o esposti del medesimo cammino alternante si genera un germoglio che viene contratto. Questa fase è quindi una fase di contrazione.

Il punto di rottura viene determinato da α_3 solo quando uno pseudonodo sia interno in un cammino alternante; quindi è necessario che un aumento abbia avuto luogo su un cammino alternante contenente lo pseudonodo e che successivamente questa componente accoppiata sia stata congiunta con un cammino alternante tramite un arco incidente in uno dei precedenti nodi esterni del cammino alternante. Quando il punto di rottura viene determinato da α_3 deve avvenire $S \in \mathcal{S}(w)$ e quindi lo pseudonodo deve venire riespanso nei suoi nodi costituenti. Si noti che quest'espansione non dà luogo ad un germoglio e che nel passo successivo uno degli archi non accoppiati del germoglio deve diventare non ammissibile in quanto $h_i = h_j = -1$ per tale arco. Questa fase viene detta *fase di espansione*.

Si vede inoltre che le ipotesi 1) e 2), valide al passo iniziale, vengono mantenute valide durante tutta l'iterazione.

Per quel che riguarda gli archi che diventano inammissibili, si tratta di archi i cui estremi hanno valori $h_i = 0$ e $h_j = -1$, oppure $h_i = h_j = -1$. Si noti che archi ammissibili in un germoglio rimangono ammissibili fintantoché il germoglio non viene riespanso. Facendo riferimento alla figura 10.22 la nuova configurazione di $G(u, w)$ potrebbe essere come in figura 10.23 se $\alpha = \alpha_1$, oppure come in figura 10.24 se $\alpha = \alpha_2$. Lo stato indicato in figura 10.23 è quello successivo all'introduzione dell'arco, mentre in figura 10.24 si nota l'esistenza di un cammino aumentante per cui, una volta operato l'aumento, la soluzione è l'ottimo.

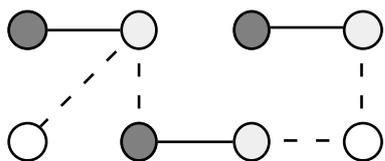


FIGURA 10.23

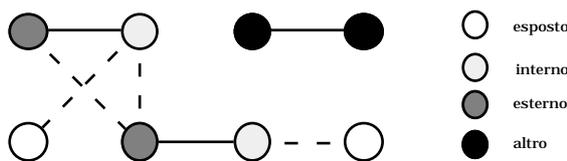


FIGURA 10.24

Valutiamo ora la complessità computazionale dell'algoritmo di Edmonds. Ovviamente questa dipende da come si realizzano e si usano le varie strutture dati. In ogni caso (anche con implementazioni "ingenue") la complessità è polinomiale e questo è il dato rilevante. Vedremo ora come sia possibile, con strutture abbastanza semplici ottenere una complessità inferiore a $O(n^4)$.

10.29 TEOREMA. *L'algoritmo di Edmonds trova un accoppiamento di minimo costo con complessità $O(nm \log n)$.*

DIMOSTRAZIONE. Conviene valutare la complessità computazionale delle operazioni comprese fra due fasi aumentanti, dato che non vi possono essere più di $n/2 = O(n)$ fasi aumentanti. Denominiamo stadio l'insieme di operazioni comprese fra due fasi aumentanti.

Indichiamo con E lo stato esposto od esterno di un nodo, con I lo stato interno e con O lo stato di nodo non raggiungibile. Notiamo che, all'interno di uno stadio, un nodo non cambia più stato se si trova nello stato E . Se si trova nello stato O può cambiare lo stato sia in E che in I . Infine se si trova in I , può mutare lo stato in E a seguito di una contrazione di un germoglio (nodo interno di un cammino alternante che viene contratto in un germoglio necessariamente esterno), oppure a seguito di una espansione di un germoglio (germoglio interno che viene espanso e quindi parte dei suoi nodi diventano nodi esterni di un cammino alternante). Dato che ad ogni passo dell'algoritmo almeno un nodo cambia stato si ottiene un numero di passi $O(n)$. Tuttavia ad ogni passo vi sono operazioni che coinvolgono un certo numero di archi e sono alla fine queste operazioni che intervengono nella valutazione della complessità. Di queste operazioni vogliamo avere una valutazione globale (all'interno di uno stadio).

Possiamo allora definire degli stati anche per gli archi a seconda degli stati dei due estremi, e precisamente abbiamo gli stati EE, EI, EO, II, IO e OO . Distinguiamo inoltre lo stato EE in uno stato EE_0 e in uno stato EE_1 a seconda se l'arco non faccia parte di uno pseudonodo oppure sì. Date le possibilità di cambiamento di stato dei nodi, un arco può cambiare stato al più quattro volte durante uno stadio e quindi si hanno al più $O(m)$ cambiamenti di stato durante uno stadio. Per il calcolo di α solo archi in EE_0 oppure in EO entrano in gioco. Inoltre tutti gli archi che non cambiano stato e che non sono contenuti in uno pseudonodo variano il costo ridotto della stessa quantità. Quindi nel calcolo di α_1 e α_2 si può sfruttare questo fatto. A questo fine si costruiscano due heap, uno, indicato con H_2 , per gli archi in EE_0 e un altro, indicato con H_1 , per gli archi in EO . Si definisca inoltre Δ la somma cumulata nei precedenti passi dei valori α . Allora non è necessario calcolare per ogni arco i valori $c_{ij} - u_i - u_j$ dato che per ogni arco in EE_0 che non abbia cambiato stato si ha $c_{ij} - u_i - u_j = c_{ij} - u_i^0 - u_j^0 - 2\Delta$ (con u_i^0 valori al momento dell'inizializzazione di H_2) e per ogni arco in EO che non abbia cambiato stato si ha $c_{ij} - u_i - u_j = c_{ij} - u_i^0 - u_j^0 - \Delta$. H_1 e H_2 possono pertanto rimanere inalterati per questi archi con i valori iniziali $c_{ij} - u_i^0 - u_j^0$. Bisogna però adottare degli accorgimenti quando un arco cambia stato. Ad esempio se assume lo stato EO il suo valore vero di costo ridotto $c_{ij} - u_i - u_j$ va aumentato di Δ prima di essere inserito in H_1 affinché il confronto con gli altri archi in E_0 sia corretto e similmente, per gli archi che assumono lo stato EE_0 , il costo ridotto va inserito in H_2 aumentato di 2Δ .

Queste operazioni di aggiustamento degli heap costano $O(\log n)$ e, siccome i cambiamenti di stato degli archi sono $O(m)$ si ha un costo (per stadio) di $O(m \log n)$.

Per ciò che riguarda la fase di ricerca, questa è assimilabile alla ricerca di un cammino aumentante per il problema di massima cardinalità con la possibilità però di riespansione dei germogli. Consideriamo prima questo problema. Si è già notato che un germoglio può venire riespanso solo se è un nodo interno di un cammino alternante e questo può avvenire solo se vi è stato un aumento sul cammino. Quindi i germogli che vengono espansi durante un certo stadio sono stati generati in stadi precedenti. Siccome vi sono al più $n/2$ germogli contemporaneamente presenti (ogni germoglio toglie almeno due nodi al grafo) non vi possono essere più di $O(n)$ espansioni. Quindi la riespansione dei germogli non aggiunge complessità alla ricerca di un cammino aumentante che richiede complessità $O(m \log n)$.

Tenendo conto della limitazione sul numero di stadi si ottiene la complessità $O(n m \log n)$. ■

10.30 ESEMPIO. La procedura verrà illustrata sul grafo K_{12} con costi euclidei (cioè il costo di ogni arco è la distanza fra i rispettivi nodi nella rappresentazione planare del grafo). I costi sono i seguenti:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	24	28	16	60	48	44	68	58	36	52	34
2	24	0	36	38	60	32	44	56	42	56	60	28
3	28	36	0	24	32	38	16	46	44	24	24	18
4	16	38	24	0	56	56	40	70	64	24	46	38
5	60	60	32	56	0	40	16	28	40	44	16	32
6	48	32	38	56	40	0	32	26	8	60	48	20
7	44	44	16	40	16	32	0	32	36	32	16	16
8	68	56	46	70	28	26	32	0	18	64	42	32
9	58	42	44	64	40	8	36	18	0	66	50	26
10	36	56	24	24	44	60	32	64	66	0	28	40
11	52	60	24	46	16	48	16	42	50	28	0	32
12	34	28	18	38	32	20	16	32	26	40	32	0

Usando la procedura (10.13) le variabili duali vengono calcolate come

$$u = (16, 8, 12, 0, 16, 8, 0, 12, 0, 12, 0, 6)$$

che determinano gli archi ammissibili indicati in figura 10.25. La procedura di accoppiamento massimo porta all'accoppiamento in figura. Da questo si determinano gli stati dei nodi e , in base al teorema 10.28, la seguente direzione di ascesa

$$h = (-1, 1, -1, 1, -1, 0, 1, 1, 0, 1, 1, 1)$$

Eseguendo i calcoli si trova $\alpha_1 = 6$ e $\alpha_2 = 5$, per cui $\alpha = 5$. Allora verrà aggiunto uno (o più) archi ammissibili congiungenti nodi esterni. Si trova

$$u := u + 5h = (11, 13, 7, 5, 11, 8, 5, 17, 0, 17, 5, 11)$$

con il relativo grafo $G(u, 0)$ (figura 10.26). Si è creato un cammino aumentante e quindi si opera una fase di aumento. L'accoppiamento massimo con i relativi stati dei nodi è in figura 10.27.

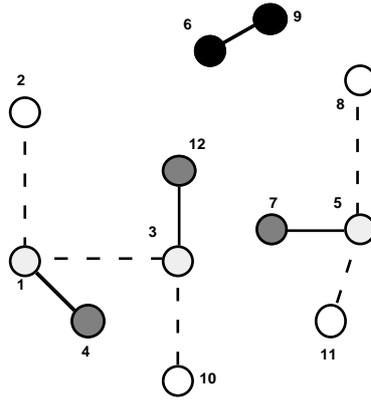


FIGURA 10.25

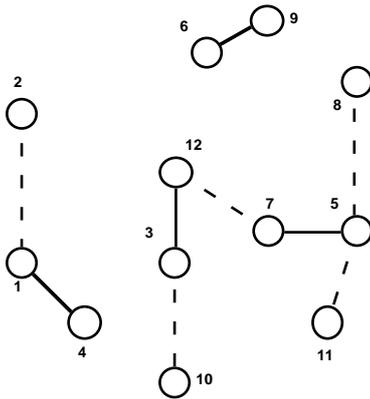


FIGURA 10.26

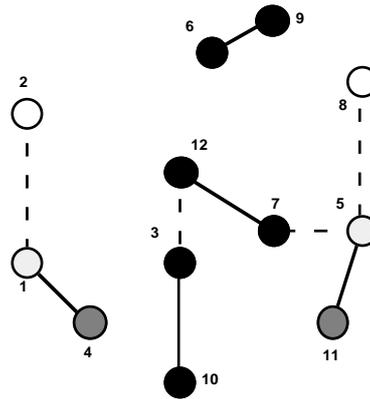


FIGURA 10.27

Allora $h = (-1, 1, 0, 1, -1, 0, 0, 1, 0, 0, 1, 0)$ e $\alpha_1 = 1, \alpha_2 = 10$. Quindi $\alpha = 1$ e si ha una fase di ricerca

$$u := u + 1h = (10, 14, 7, 6, 10, 8, 5, 18, 0, 17, 6, 11)$$

Il nuovo grafo $G(u, 0)$ è rappresentato in figura 10.28. Si è generato il germoglio $S_1 = \{6, 8, 9\}$. Quindi lo stato dei nodi 6 e 8 cambia e si introduce la variabile duale per lo pseudonodo S_1 (figura 10.29).

Le direzioni sono allora

$$h = (-1, 1, 0, 1, -1, 1, 0, 1, 1, 0, 1, 0) \quad g(S_1) = -2$$

da cui $\alpha_1 = 1$ e $\alpha_2 = 5$, cioè $\alpha = 1$. Allora

$$u := u + 1h = (9, 15, 7, 7, 9, 9, 5, 19, 1, 17, 7, 11) \quad w(S_1) := 0 + 1g(S_1) = -2$$

L'aggiunta degli archi $(6, 12)$ e $(4, 10)$ cambia lo stato di alcuni nodi e si ha la situazione in figura 10.30 che implica

$$h = (-1, 1, 1, 1, -1, 1, 1, 1, 1, -1, 1, -1) \quad g(S_1) = -2$$

e $\alpha = \infty, \alpha_2 = 2$, quindi $\alpha = 2$ da cui

$$u = (7, 17, 9, 9, 7, 11, 7, 21, 3, 15, 9, 9) \quad w(S_1) = -6,$$

con gli archi ammissibili indicati in figura 10.31. Si è trovato un cammino aumentante per cui, operato l'aumento e non essendoci più archi esposti, la soluzione è ottima (figura 10.32). Come verifica di ottimalità si ha che

$$\sum_{e \in E} x_e = 118 = \sum_{i \in N} u_i + w(S_1)$$

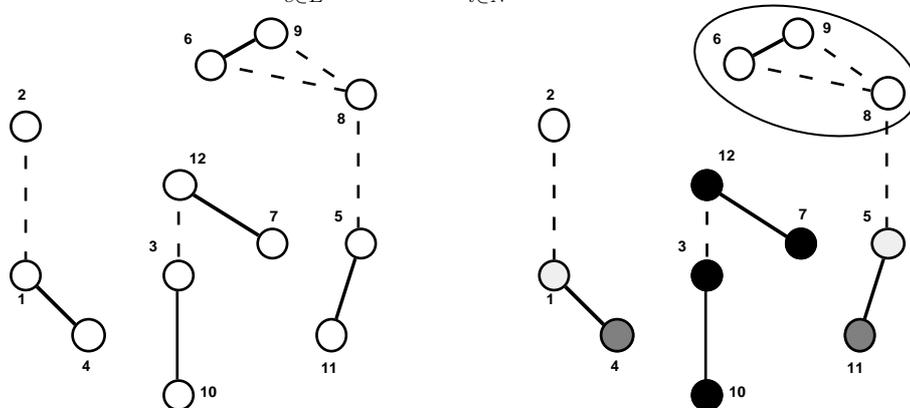


FIGURA 10.28

FIGURA 10.29

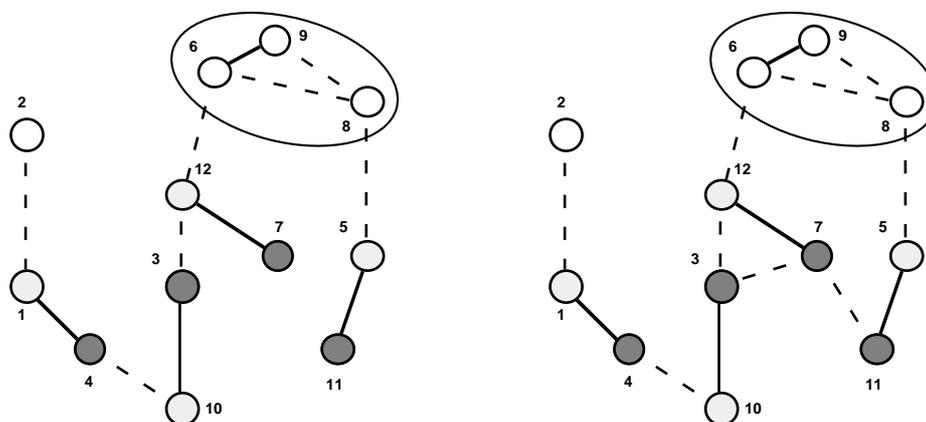


FIGURA 10.30

FIGURA 10.31

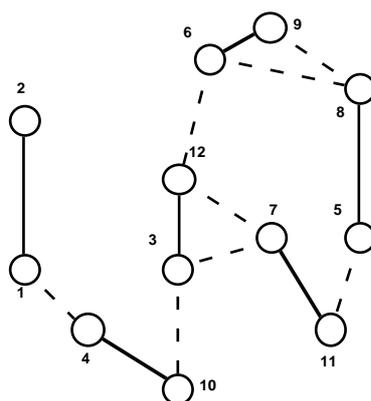


FIGURA 10.32