

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI MATEMATICA
"TULLIO LEVI-CIVITA"

Corso di Laurea Triennale in Matematica

Il problema del matching massimo

Relatore:
Prof. Marco Di Summa

Laureanda:
Eleonora Belleso
Matricola:
1217915

23 Settembre 2022
Anno Accademico: 2021/2022

Indice

Introduzione	3
1 Cammini alternanti e aumentanti	5
1.1 Nozioni preliminari	5
1.2 Matching e cammini alternanti	7
1.3 Condizione di ottimalità di un matching massimo	9
2 Assegnamento	13
2.1 Matching massimo su grafo bipartito	14
2.2 L'algoritmo di Hopcroft-Karp	17
2.3 Minima copertura di vertici in grafi bipartiti	19
2.4 Assegnamento pesato	21
2.5 Metodo ungherese	25
3 Algoritmo del matching massimo di Edmonds	31
3.1 Alberi e foreste alternanti	32
3.2 Fiori e blossoms	36
3.3 L'algoritmo di Edmonds	39
Bibliografia	47

Introduzione

Alla base del presente studio vi è l'intento di analizzare in maniera precisa e dettagliata un problema di ottimizzazione combinatoria, per il quale esistono algoritmi risolutivi polinomiali e che presenta anche interessanti applicazioni alla realtà di tutti i giorni. Si vuole, per esempio, dare una risposta logico-matematica alle seguenti domande:

1. si vuole dividere un gruppo di studenti in coppie, ma alcuni studenti non vogliono fare coppia con certi altri: come scoprire qual è la miglior soluzione che vede accoppiati il maggior numero di studenti?
2. si vogliono assegnare delle attività (*jobs*) j_1, \dots, j_n a degli impiegati (*individuals*) i_1, \dots, i_n rispettando le competenze di ciascun impiegato (ad esempio l'individuo i_k sa svolgere i lavori j_{k_1}, j_{k_2}, \dots). È possibile? Qual è il maggior numero di attività che possono essere effettivamente svolte dagli impiegati?

Questi sono in realtà casi particolari di uno stesso problema che prende il nome di "*problema del matching massimo*". Lo scopo della presente trattazione è quello di fornire e descrivere dettagliatamente algoritmi efficienti per un'effettiva risoluzione di tale problema.

L'elaborato consta di tre capitoli: nel primo viene fornita un'analisi accurata delle strutture discrete (quali grafi, matchings, cammini alternanti, etc...) necessarie ad ideare un coerente modello matematico.

Nel secondo capitolo verrà affrontato il problema dell'*assegnamento*, un caso particolare del matching massimo e di più facile risoluzione (corrispondente all'istanza della *domanda-2*), per il quale verranno presentati e discussi diversi algoritmi risolutivi. Sempre in questo secondo capitolo il problema verrà inoltre posto in termini della programmazione lineare e rivisitato in combinazione con il suo programma duale.

La trattazione del matching massimo nel caso generale è invece riservata al terzo capitolo.

In conclusione, questo studio approfondito è giustificato dal fatto che, all'interno dei problemi di ottimizzazione combinatoria, quello del matching

massimo è certamente uno dei più affascinanti, inoltre, data l'efficienza dei suoi algoritmi risolutivi, è naturale che questo problema venga poi spesso invocato nella costruzione di successivi algoritmi per problematiche di più difficile risoluzione.

Capitolo 1

Cammini alternanti e aumentanti

In questo capitolo vengono inizialmente richiamate alcune nozioni di base della teoria dei grafi, come presentate in [3], necessarie per una completa comprensione degli argomenti trattati nel presente studio. Segue un'analisi dettagliata del concetto di cammini alternanti ed aumentanti, e in particolare delle loro prime proprietà impiegate per determinare le condizioni di ottimalità per un matching massimo. Gran parte degli enunciati e delle dimostrazioni proviene da [2] e [3].

1.1 Nozioni preliminari

Definizione 1.1. Un *grafo non orientato (semplice)* G è una coppia ordinata $G = (V, E)$ dove V è un insieme finito di *vertici* o *nodi*, mentre E è un insieme di coppie non ordinate di nodi distinti dette *archi*.

Nella presente trattazione, dato un arco $e \in E$ si è scelto di indicare lo stesso come $e = (u, v)$ oppure $e = (v, u)$ indifferentemente, dove i nodi u, v sono detti *estremi* dell'arco e , inoltre si dice che e *incide* su u e v . D'altra parte due vertici $u, v \in V$ si dicono *adiacenti* o *vicini* se sono estremità di un arco in G . Quindi scelto un vertice $v \in V$, l'insieme dei vertici adiacenti a v viene detto *intorno* (*neighborhood*) di v ed è rappresentato con il simbolo $N(v)$. Viene invece indicato con $d(v)$ il grado di un vertice $v \in V$ che corrisponde al numero di archi incidenti su di esso.

Definizione 1.2. Un grafo $H = (V(H), E(H))$ è un *sottografo* di $G = (V(G), E(G))$ se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. (Si noti che $E(H)$ non contiene nessun arco con un'estremità in $V(G) \setminus V(H)$ per definizione di grafo).

Dato un sottoinsieme $S \subseteq V$, viene indicato con $\delta(S)$ il *taglio* (*cut*) di S , cioè l'insieme degli archi del grafo $G = (V, E)$ che hanno esattamente un'estremità in S . Si definisce quindi con $d(S) = |\delta(S)|$ il *grado* dell'insieme S . Nel caso in cui $S = \{v\}$ per qualche $v \in V$, allora il taglio viene detto "*stella*" di v ed è indicato con il simbolo $\delta(v)$ invece di $\delta(\{v\})$, per semplicità di scrittura.

A partire da queste basilari nozioni è poi possibile definire l'operazione di "*differenza*" tra il grafo $G = (V(G), E(G))$ ed un suo sottografo $H = (V(H), E(H))$ come il nuovo grafo $G \setminus H$ di vertici $V(G) \setminus V(H)$ ed archi $E(G) \setminus E(H) \setminus \delta(H)$.

Un *cammino* (*path*) P_{v_1, v_k} fra due nodi v_1 e v_k di un grafo $G = (V, E)$ è una sequenza alternata $v_1, e_1, v_2, e_2, \dots, v_{k-1}, e_{k-1}, v_k$ di nodi e archi distinti di G , dove $e_i = (v_i, v_{i+1})$, $\forall i = 1, \dots, k-1$. I nodi v_1, v_k sono detti *estremi* del cammino, mentre v_2, \dots, v_{k-1} vengono chiamati nodi *interni* al cammino. La *lunghezza* di un cammino è pari al numero dei suoi archi.

Infine un cammino i cui estremi coincidono ($v_1 = v_k$) è chiamato *ciclo*.

Definizione 1.3. Un grafo $G = (V, E)$ si dice *connesso* se per ogni coppia di nodi $u, v \in V$ contiene un cammino di estremi u e v . Si definiscono quindi le *componenti connesse* di un grafo G come i sottografi connessi massimali di G .

Definizione 1.4. Un *albero* è un grafo $T = (V, E)$ connesso e aciclico.

Una *foglia* di un albero corrisponde ad un nodo di grado 1 del grafo, mentre un grafo le cui componenti connesse sono alberi viene generalmente chiamato *foresta*.

Lemma 1.1. *Ogni albero con almeno due vertici ha almeno due foglie.*

Dimostrazione. Si consideri P un cammino massimale per l'albero T e siano u, v i suoi estremi. Si vuole mostrare che entrambi i nodi u, v sono foglie in T . Se per assurdo $d(u) > 1$ allora esisterebbe un altro arco (u, w) dove sicuramente $w \notin V(P)$ se no si avrebbe un ciclo. *Assurdo.* Contro la definizione di albero. Però così facendo si potrebbe comunque continuare il cammino P aggiungendo l'arco (u, w) . *Assurdo.* Contro la massimalità di P .

Pertanto $d(u) = 1$, e analogamente si dimostra lo stesso risultato per il vertice v . Si conclude quindi che entrambi i nodi u, v sono foglie per T . \square

Proposizione 1.2. *Sia $T = (V, E)$ un albero, allora $|V| = |E| + 1$.*

Dimostrazione. Si dimostra per induzione sul numero dei vertici n . Per $n = 1$ la tesi è banalmente verificata.

Sia quindi $T = (V, E)$ un albero con $|V| = n > 1$. Per il *lemma 1.1* si ha che T ha almeno una foglia $v \in V$. Rimuovendo quindi il nodo v da T , e di conseguenza anche l'unico arco che lo connette all'albero, si ottiene un nuovo albero $T' = (V', E')$ dove $|V'| = |V| - 1 = n - 1$ e $|E'| = |E| - 1$ in cui è quindi possibile applicare l'ipotesi induttiva $|V'| = |E'| + 1 \iff n - 1 = |E| - 1 + 1$ da cui segue la tesi. \square

Nella trattazione che segue saranno di rilevante importanza anche altre classi di grafi, quali:

- *grafo vuoto* $G = (V, E)$ dove $V = \{v_1, \dots, v_n\}$ ed $E = \emptyset$
- *grafo completo* (o *clique*) dove ogni coppia di vertici è adiacente, cioè $V = \{v_1, \dots, v_n\}$ ed $E = \{(v_i, v_j) \mid i, j \in \{1, \dots, n\}, \text{ con } i \neq j\}$. Si noti che per ogni numero naturale n esiste uno ed un solo grafo completo con n vertici che sarà indicato con K_n
- *grafo bipartito* se è possibile partizionare i suoi vertici in due insiemi A e B tali che ogni arco del grafo abbia un'estremità in A e l'altra in B . Un grafo bipartito sarà quindi rappresentato come $G = (A \cup B, E)$. In particolare fissato un numero naturale n il grafo bipartito $G = (A \cup B, E)$ tale che $|A| = |B| = n$ e per cui ogni nodo di A è adiacente a tutti quelli di B , è detto *grafo bipartito completo* e sarà indicato con il simbolo $K_{n,n}$.

1.2 Matching e cammini alternanti

Si è ora interessati a definire in maniera formale il concetto di matching già in parte anticipato nell'introduzione al testo.

Definizione 1.5. Dato un grafo $G = (V, E)$ un *matching* di G è un sottoinsieme $M \subseteq E$ di archi senza estremi in comune.

Un vertice $v \in V$ si dice *coperto dal matching* M (o *M-coperto*) se esattamente un arco di M incide su v , al contrario viene detto *M-esposto* se nessun arco di M incide su v .

Un matching M di G può anche essere definito come:

- *massimale* se non esiste un altro matching di G che contiene propriamente M
- *massimo* se non esiste un altro matching di G che ha cardinalità strettamente maggiore di M (si osservi quindi che *massimo* \implies *massimale*, mentre in generale non vale il viceversa)

- *perfetto* se copre tutti i nodi del grafo. Ovviamente un matching perfetto ha cardinalità $\frac{n}{2}$ (dove $n = |V(G)|$) e pertanto nessun grafo con un numero dispari di nodi ammette un matching perfetto (si noti che *perfetto* \implies *massimo* ma anche in questo caso non vale il viceversa, come controesempio basti pensare ad un grafo con n dispari)

Viene quindi indicata con $\nu(G)$ la cardinalità di un matching massimo per un generico grafo G .

Sottolineando e ribadendo fin da subito che l'obiettivo del presente studio è quello di determinare un matching massimo M per un dato grafo G , si decide quindi di proseguire introducendo alcuni strumenti essenziali a tal fine.

Definizione 1.6. Dato un grafo $G = (V, E)$ e un suo matching $M \subseteq E$, un cammino (o ciclo) di G viene detto *M-alternante* se i suoi archi appartengono alternativamente ad M ed a $E \setminus M$.

Definizione 1.7. Un cammino $v_1, e_1, v_2, e_2, \dots, v_{k-1}, e_{k-1}, v_k$ *M-alternante* viene anche detto *M-aumentante* se le sue estremità v_1, v_k risultano essere vertici *M-esposti*.

Osservazione 1. Un cammino *M-aumentante* ha sempre lunghezza l dispari, con gli archi estremanti $e_1, e_{k-1} \notin M$. Pertanto tale cammino contiene esattamente $\frac{l+1}{2}$ archi non del matching e $\frac{l-1}{2}$ archi del matching.

Per semplificare la notazione, da ora in poi si sceglie di indicare con \bar{M} l'insieme degli archi di $G = (V, E)$ che non stanno nel matching M .

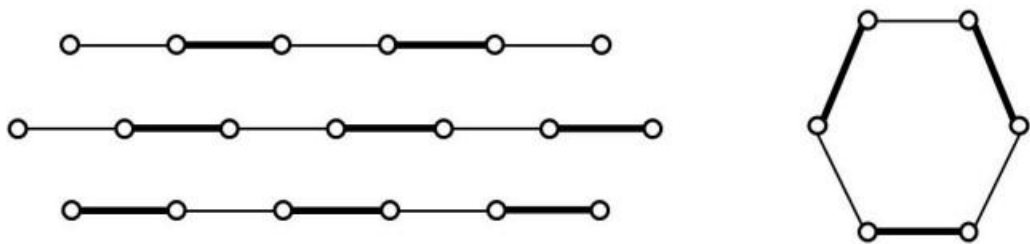


Figura 1.1: cammini e cicli alternanti, gli archi ingrossati sono quelli che fanno parte del matching M

Siccome tornerà utile nella prossima sezione, si definisce già a questo livello l'operazione di *differenza simmetrica* tra due matchings: dati due matchings M_1 ed M_2 per un generico grafo G si indica con $M_1 \triangle M_2$ l'insieme degli archi $(M_1 \setminus M_2) \cup (M_2 \setminus M_1)$. Vale in particolare il seguente risultato:

Lemma 1.3. *Sia $G = (V, E)$ un grafo e M_1, M_2 due matchings di G , allora le componenti connesse del sottografo $G' = (V, M_1 \Delta M_2)$ di G sono cammini e cicli alternanti per M_1 e M_2 . Inoltre ognuno degli estremi di uno di questi cammini è esposto per M_1 o M_2 .*

Dimostrazione. Ogni vertice $v \in V$ di G è estremo di al massimo un arco per M_1 e un arco per M_2 , per definizione di matching. Quindi esistono al massimo due archi di $M_1 \Delta M_2$ incidenti in v , uno di $M_1 \cap \bar{M}_2$ e l'altro di $M_2 \cap \bar{M}_1$. Questo basta per dimostrare che le componenti connesse del grafo $G' = (V, M_1 \Delta M_2)$ sono cammini o cicli alternanti (si ricordi che un singolo vertice è di per sè un cammino alternante di lunghezza nulla).

Sia ora $v' \in V$ un nodo estremo di un cammino in G' per cui esiste un arco di $M_1 \cap \bar{M}_2$ incidente v' stesso. Si suppone per assurdo che esista anche un arco di M_2 incidente v' e che quindi non appartiene ad M_1 , ma allora anche quest'arco farà parte di $M_1 \Delta M_2$. *Contraddizione.* Infatti così facendo v' non sarebbe più un estremo del cammino alternante in G' , ma un nodo interno ad esso. Perciò v' deve essere vertice esposto per M_2 . Analogamente si può ripercorrere lo stesso ragionamento invertendo i ruoli di M_1 e M_2 . \square

1.3 Condizione di ottimalità di un matching massimo

A partire dalle nozioni illustrate nella sezione precedente è possibile delineare, in maniera abbastanza naturale ed intuitiva, una prima formulazione algoritmica per il calcolo di un matching massimo in un grafo dato.

Vale innanzitutto il seguente risultato:

Lemma 1.4. *Siano M un matching di $G = (V, E)$ e P un cammino M -aumentante. Allora $M' = M \Delta P$ è un matching di G di cardinalità $|M| + 1$.*

Dimostrazione. Sia $M' = M \Delta P$ dove M è un matching di G e P è un cammino M -aumentante. Si vuole mostrare che M' così definito è ancora un matching di G , cioè che su ogni vertice di G incide al più un arco di M' . Sia quindi $v \in V$, si possono avere i seguenti tre casi:

1. $v \notin P$ e un arco $e \in M'$ incide su v , ma ciò è possibile se e solo se $e \in M$. Essendo quindi M un matching di G , allora su $v \notin P$ incide al più un arco di M'
2. $v \in P$ è un vertice interno al cammino per cui vale che $|\delta(v) \cap M'| = |\delta(v) \cap M| = 1$ per definizione di cammino alternante

3. $v \in P$ è uno dei due estremi del cammino per cui vale che $|\delta(v) \cap M'| = |\delta(v) \cap M| + 1 = 1$, dove $|\delta(v) \cap M| = 0$ segue dal fatto che le estremità di un cammino aumentante sono vertici M -esposti.

Resta quindi da mostrare che $|M'| = |M| + 1$. Poiché P è M -aumentante dall'osservazione 1 fatta nella sezione precedente, segue che:

$$|P \setminus M| = |P \cap M| + 1$$

In conclusione si ha che:

$$\begin{aligned} |M'| &= |M \triangle P| = |(M \setminus P) \cup (P \setminus M)| = |M \setminus P| + |P \setminus M| = \\ &|M \setminus P| + |P \cap M| + 1 = |M| + 1 \end{aligned}$$

□

Dal precedente lemma segue subito che se un matching M di un grafo G ammette un cammino M -aumentante in G , allora non è massimo. In particolare vale anche il viceversa, come affermato nel seguente teorema:

Teorema di Berge 1.5. *Un matching M di un grafo $G = (V, E)$ è massimo se e solo se non esiste un cammino M -aumentante in G .*

Dimostrazione. Poiché la prima implicazione (\implies) discende direttamente dal lemma 1.4 (appena discusso) è sufficiente dimostrare che se un matching M non è massimo, allora esiste sempre un cammino M -aumentante (\impliedby). A tal fine sia M' un matching più grande di M (non necessariamente massimo) e si consideri il sottografo $G' = (V, M' \triangle M)$ di G . Dato che $|M'| > |M|$ sicuramente qualche componente connessa di G' dovrà contenere più archi di M' che di M . Si è già mostrato, per il lemma 1.3, che tali componenti connesse sono cammini e cicli alternanti per M' e M . Dato però che i cicli alternanti hanno cardinalità pari e quindi ugual numero di archi sia di M che di M' , allora necessariamente deve esistere almeno un cammino alternante che contiene un arco in più di M' rispetto a quelli di M . Si è allora mostrato che tale cammino alternante è anche M -aumentante, come si voleva. □

Fu proprio sulla base di tale teorema 1.5 che il matematico francese *Cau-de Berge* propose per primo l'idea di ricercare questi cammini aumentanti all'interno di un dato grafo G come un effettivo algoritmo per risolvere il problema del matching massimo. Si delineava quindi un primitivo SCHEMA ALGORITMICO PER IL CALCOLO DEL MATCHING MASSIMO:

- *Input*: un grafo $G = (V, E)$

1.3. CONDIZIONE DI OTTIMALITÀ DI UN MATCHING MASSIMO 11

- *Inizializzazione*: $M = \emptyset$
- *Ricorsione*: finché esiste un cammino M -aumentante P in G poni $M = M \triangle P$
- *Output*: un matching M di cardinalità massima in G

Non era però chiaro a priori come, dato un grafo G e un matching M su G , calcolare in maniera efficiente un cammino M -aumentante.

Tale problematica venne successivamente affrontata e risolta dal matematico statunitense *Harold Kuhn*, ma solamente ridotta al caso su grafi bipartiti (di più facile risoluzione), come viene presentato nella *sezione 2.1* della presente trattazione.

D'altra parte una tale procedura polinomiale per il calcolo di cammini aumentanti su grafo generale venne estesa solo in un secondo momento grazie al fondamentale lavoro di *Jack Edmonds*, che verrà esposto e discusso nel capitolo terzo del presente studio.

Capitolo 2

Assegnamento

«Suppose n *individuals* ($i = 1, \dots, n$) are available for n *jobs* ($j = 1, \dots, n$) [...] If an assignment (not necessarily optimal) of certain qualified individuals to jobs is given, then the easiest way to improve it is to assign any unassigned individual to an unassigned job for which he qualifies. If this is possible, the given assignment is said to be *incomplete*; otherwise, it is *complete*. If the assignment is complete, then it is reasonable to attempt an improvement by means of a *transfer*. A transfer changes the assignment of r distinct individuals i_1, \dots, i_r employed in jobs j_1, \dots, j_r . It moves i_1 into an unassigned job j_0 and i_k into job j_{k-1} for $k = 2, \dots, r$. All of the new assignments (i_k to j_{k-1}) are assumed to be qualified for $k = 1, \dots, r$. [...]

Any assignment is optimal if and only if it is complete after every possible transfer.»¹

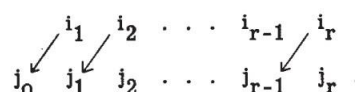


Figura 2.1: esempio di "transfer"

Il tema principale di questo capitolo è l'"assegnamento", cioè il caso particolare del problema del matching massimo su grafo bipartito. Vengono esposti principalmente risultati dovuti al matematico *Harold William Kuhn* [6] che per primo ideò un algoritmo efficiente in grado di risolvere tale problema. In questo capitolo viene inoltre presentato l'algoritmo di *Hopcroft-Karp*, trattato da [8], come un miglioramento del precedente.

¹ H. W. Kuhn, "The Hungarian method for the assignment problem", ONR Logistics Project, Department of Mathematics, Princeton University, "Bryn Mawr College"

Successivamente sarà introdotto il concetto di *trasversale* (in inglese *vertex cover*) necessario per riformulare tale problema nel suo programma duale. In conclusione, riprendendo il pensiero di Kuhn [6], verrà proposta una variazione del presente problema nota come "*assegnamento pesato*", per il quale l'algoritmo più veloce conosciuto fino ad oggi è anche quello più vecchio ed è equivalente al così detto "*metodo ungherese*". In generale il problema del matching pesato su grafo non bipartito è uno dei problemi "più difficili" di ottimizzazione combinatoria che può essere risolto in tempo polinomiale. Un tale algoritmo fu ideato ed implementato ad opera del matematico Jack Edmonds, ma non verrà approfondito nel presente studio.

2.1 Matching massimo su grafo bipartito

In questa sezione si decide di focalizzare lo studio del matching su grafi bipartiti, come dato in [1], [2] e [5]. Vengono inizialmente introdotte condizioni necessarie e sufficienti affinché un grafo bipartito ammetta un matching perfetto.

Teorema di Hall 2.1. *Sia $G = (A \cup B, E)$ un grafo bipartito. Allora G ammette un matching che copre tutti i vertici di A se e solo se per ogni sottoinsieme $S \subseteq A$ si ha che $|S| \leq |N(S)|$.*

Dimostrazione. Si ricorda innanzitutto che $\forall S \subseteq A$ l'insieme $N(S)$ definito come $N(S) = \{v \in V \mid (u, v) \in E \text{ per qualche } u \in S\}$ è detto l'*intorno* o *vicinato* di S .

(\implies) Sia M un matching che copre tutti i vertici di A . Allora $\forall S \subseteq A$ il matching M copre anche tutti i vertici di S , ovvero $\forall u \in S$ esiste un $v \in B$ t.c. $(u, v) \in M$. Poiché $v \in N(S)$ e gli estremi degli archi di M sono tutti distinti per definizione di matching, si conclude che $|S| \leq |N(S)|$.

(\impliedby) Viceversa, sia M un matching massimo. Supponendo per assurdo che $a \in A$ sia un vertice M -esposto, e che quindi non esiste un matching che copre tutti i nodi di A , si troverà un sottoinsieme $S \subseteq A$ per cui $|S| > |N(S)|$ giungendo dunque ad una contraddizione. Siano infatti $Z := \{v \in A \cup B \mid v \text{ è raggiungibile da } a \text{ tramite un cammino } M\text{-alternante}\}$ e $S := Z \cap A$. Poiché M è massimo, allora G non contiene cammini M -aumentanti (*teorema 1.5*), quindi i cammini alternanti che partono da a non possono finire in un altro vertice esposto per M . Ne segue che tutti i vertici di $Z \setminus \{a\}$ sono coperti dal matching M . In particolare sia $P = a, v_1, v_2, \dots, v_n$ un cammino M -alternante dove si ha che $v_i \in A$ per ogni i pari, mentre $v_i \in B$ per ogni i dispari, e che l'arco $(v_i, v_{i+1}) \in M$ se e solo se i è dispari.

Si può dunque concludere che:

1. M induce un matching perfetto tra $S \setminus \{a\}$ e $Z \cap B$
2. $|Z \cap B| = |S| - 1$
3. $N(S) = Z \cap B$. Infatti l'inclusione $Z \cap B \subseteq N(S)$ è ovvia poiché G è bipartito, mentre per provare che $N(S) \subseteq Z \cap B$ si consideri un nodo $v \in N(S)$ e sia $u \in S$ adiacente a v . Sia P un cammino M -alternante da a ad u . Se $(u, v) \in M$ allora $v \in P$, quindi $v \in Z$. Se invece $(u, v) \notin M$, o v è un nodo interno di P oppure P può essere esteso ad un cammino M -alternante più lungo aggiungendo v , quindi in ogni caso $v \in Z$. Poiché $v \in N(S) \subseteq B$ segue l'uguaglianza.

In conclusione si è mostrato che $\exists S \subseteq A$ t.c. $|N(S)| = |Z \cap B| = |S| - 1$, cioè $|N(S)| < |S|$. *Contraddizione*. L'insieme S così definito viola la condizione di Hall. \square

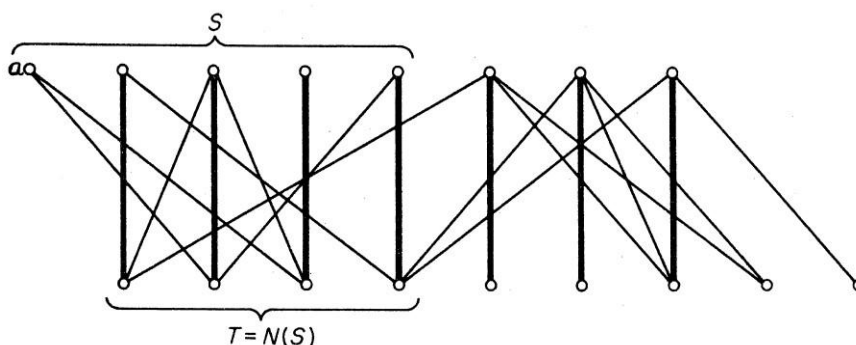


Figura 2.2: grafo esemplificativo per il teorema di Hall

Corollario 2.2. *Sia $G = (A \cup B, E)$ un grafo bipartito. Allora G ha un matching perfetto se e solo se $|A| = |B|$ e $\forall S \subseteq A$ vale che $|S| \leq |N(S)|$.*

In generale dato un grafo bipartito $G = (A \cup B, E)$ è possibile calcolare un matching massimo di G , e quindi il parametro $\nu(G) = \max_{M \subseteq E \text{ matching}} |M|$, utilizzando la teoria dei flussi. Ovviamente $\nu(G) \leq \min\{|A|, |B|\}$ e si è appena mostrato che $\nu(G) = |A|$ o $\nu(G) = |B|$ qualora sia soddisfatta la condizione di Hall (teorema 2.1).

Si consideri infatti il grafo orientato $D = (V', A)$ ottenuto a partire da G aggiungendo due vertici s e t , e connettendoli rispettivamente a tutti i nodi di A e a tutti quelli di B . Si orientino poi gli archi da s ad A , da A a B e da B a t . Inoltre siano le capacità di tutti gli archi poste uguali ad 1. Allora un flusso da s a t massimo intero in D (che esiste sempre poiché le capacità di partenza sono intere) corrisponde ad un matching di cardinalità massima

per G e viceversa. In particolare per ogni arco $(u, v) \in E$ si ha anche che $(u, v) \in M$ se e solo se il flusso attraverso l'arco (u, v) vale effettivamente 1, in simboli $f_{(u,v)} = 1$. Inoltre $|M| = \sum_{a \in A} f_{(s,a)} = \nu(G)$.

Applicando quindi l'ALGORITMO DI FORD-FULKERSON è possibile determinare un flusso massimo intero da s a t e quindi un matching massimo dopo al più $n = |V(G)|$ aumenti, dove ogni aumento richiede un tempo $O(m)$, con $m = |E(G)|$. In conclusione si è appena mostrato come il problema del matching di cardinalità massima per grafi bipartiti possa essere risolto in un tempo $O(nm)$.

Come ribadito più volte nella presente trattazione, questo risultato è essenzialmente dovuto a Kuhn.

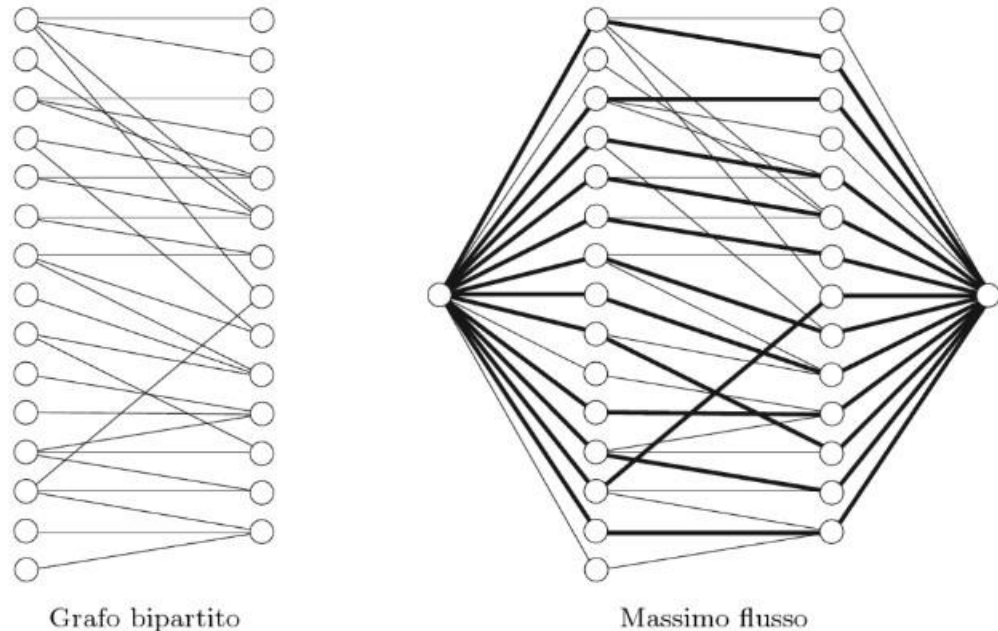


Figura 2.3: algoritmo per il calcolo del matching massimo su grafi bipartiti

Si osservi in particolare che ad ogni iterazione l'algoritmo di Ford-Fulkerson determina un cammino f -aumentante (minimo) in D che corrisponde ad un cammino M -aumentante in G . Tale algoritmo termina quindi quando non esistono più cammini aumentanti e in effetti il matching ritornato è massimo. Quest'ultima osservazione sarà di fondamentale importanza nella prossima sezione.

2.2 L'algoritmo di Hopcroft-Karp

Tutti i risultati della presente sezione sono tratti da [8].

Si vuole ora presentare ed analizzare un algoritmo "alternativo" a quello precedente, o meglio un suo miglioramento, che infatti permette di risolvere il problema del matching massimo su grafi bipartiti in $O(m\sqrt{n})$.

Sia $G = (A \cup B, E)$ un grafo bipartito e M un matching di G . Invece di ricercare i cammini M -aumentanti uno alla volta, l'algoritmo di Hopcroft-Karp lavora in "fasi". In ogni singola fase (mediante metodo *BST*) si costruisce una raccolta massimale di cammini M -aumentanti minimi e disgiunti sui vertici, che verranno poi utilizzati contemporaneamente per aumentare il matching stesso.

La correttezza di tale algoritmo discende dai seguenti risultati, il primo dei quali è valido in generale anche per grafi non bipartiti.

Lemma 2.3. *Siano $G = (V, E)$ un grafo non orientato ed M, M' due matchings di G , tali che $|M'| = |M| + k$, dove $k \geq 1$ è intero. Allora esistono almeno k cammini M -aumentanti disgiunti sui vertici in G , e almeno uno di questi cammini ha lunghezza al massimo pari a $\frac{n}{k} - 1$, dove $n = |V(G)|$.*

Dimostrazione. Si consideri il sottografo $G' = (V, M \triangle M')$ le cui componenti connesse, per il *lemma 1.3*, risultano essere cammini e cicli alternanti per M ed M' . Poiché $|M'| = |M| + k$ allora l'insieme $M \triangle M'$ contiene sicuramente k archi in più di M' rispetto a quelli di M . Dato che cicli alternanti sia per M che per M' sono cicli pari e contengono quindi in ugual numero archi di M ed M' , devono esistere allora almeno k cammini alternanti disgiunti sui vertici che iniziano e terminano con un arco di M' . Questi cammini risultano quindi essere anche M -aumentanti, come si voleva. Inoltre almeno uno di questi cammini contiene al massimo $\frac{n}{k}$ vertici e quindi la sua lunghezza può essere al massimo pari a $\frac{n}{k} - 1$. \square

Lemma 2.4. *Si considerino $G = (A \cup B, E)$ un grafo bipartito e M un matching di G . Sia P un cammino M -aumentante minimo e P' un cammino aumentante rispetto al nuovo matching $M' = M \triangle P$. Allora si ha che $|P'| \geq |P| + 2|P \cap P'|$.*

Dimostrazione. Si definisce $N = (M \triangle P) \triangle P'$. Allora N è un matching di G di cardinalità $|N| = |M \triangle P| + 1 = |M| + 1 + 1 = |M| + 2$ (*lemma 1.4*). Per il *lemma 2.3* appena dimostrato si ha quindi che il sottografo $G' = (A \cup B, M \triangle N)$ contiene due cammini M -aumentanti disgiunti sui vertici che saranno indicati con P_1 e P_2 . Data la proprietà di minimalità di P , si ha allora che $|P_1|, |P_2| \geq |P|$. Poiché $M \triangle N = P \triangle P'$ segue che

$|P \Delta P'| \geq |P_1| + |P_2| \geq 2|P|$. Ricordando che $|P \Delta P'| = |P| + |P'| - 2|P \cap P'|$ si conclude allora che $|P'| \geq |P| + 2|P \cap P'|$. \square

Lemma 2.5. *Siano $G = (A \cup B, E)$ un grafo bipartito e M un matching di G . Data $\mathcal{P} = \{P_1, P_2, \dots, P_\ell\}$ una raccolta massimale di cammini M -aumentanti minimi disgiunti sui vertici si definisce $M' = M \Delta P_1 \Delta P_2 \Delta \dots \Delta P_\ell$. Se P' è un cammino M' -aumentante, allora $|P'| > |P_1| = |P_2| = \dots = |P_\ell|$.*

Dimostrazione. Si riordinano i cammini M -aumentanti in modo tale che i primi k abbiano intersezione non vuota con il cammino P' , pertanto $k \leq \ell$. Dato che i cammini selezionati sono disgiunti sui vertici, allora P_k è ancora un cammino aumentante minimo rispetto al matching $M_{k-1} = M \Delta P_1 \Delta P_2 \Delta \dots \Delta P_{k-1}$. Inoltre P' risulta essere un cammino M_k -aumentante, dove $M_k = M_{k-1} \Delta P_k$. Per il lemma precedente (*lemma 2.4*) si conclude allora che $|P'| \geq |P_k| + 2|P' \cap P_k| > |P_k|$, poiché per ipotesi $|P' \cap P_k| \neq 0$. \square

Tali lemmi mostrano in sostanza come i cammini aumentanti minimi, trovati ad ogni successiva fase dell'algoritmo, diventino effettivamente sempre più lunghi. Vale in conclusione il seguente fondamentale risultato:

Teorema 2.6. *L'algoritmo di Hopcroft-Karp trova un matching massimo in un grafo bipartito dopo al massimo $2\sqrt{n}$ fasi.*

Dimostrazione. Dopo \sqrt{n} fasi, se l'algoritmo non è ancora terminato, allora la lunghezza del cammino aumentante minimo nel grafo è almeno \sqrt{n} . Applicando il *lemma 2.3* si verifica che la cardinalità del matching corrente è più piccola di quella del matching massimo di al massimo \sqrt{n} . Infatti sia M' un effettivo matching massimo per il grafo. Supponendo per assurdo che $|M'| = |M| + k$ con $k > \sqrt{n}$ si giunge subito ad una contraddizione, in quanto qualsiasi cammino aumentante minimo avrebbe al massimo lunghezza pari a $\frac{n}{k} - 1 < \frac{n}{\sqrt{n}} - 1 = \sqrt{n} - 1$, mentre si è appena mostrato che ogni cammino aumentante minimo ha lunghezza almeno \sqrt{n} . *Assurdo.* Segue allora che è possibile determinare un matching massimo per tale grafo bipartito dopo al più \sqrt{n} iterazioni aggiuntive. \square

Si è provato in sostanza che l'algoritmo di Hopcroft-Karp termina in al massimo $2\sqrt{n}$ iterazioni, restituendo una soluzione corretta per il problema del matching massimo su grafi bipartiti. Inoltre, come accennato nel precedente capitolo, poiché ogni aumento richiede un tempo $O(m)$, allora il costo computazionale complessivo dell'algoritmo è proprio $O(m\sqrt{n})$.

2.3 Minima copertura di vertici in grafi bipartiti

In questa sezione si cerca di legare il problema dell'assegnamento al suo programma duale, come presentato in [1], [2] e [6].

Riprendendo il caso esemplificativo degli n individui $\{i_1, \dots, i_n\}$ selezionati per svolgere gli n lavori $\{j_1, \dots, j_n\}$, si decide ora di assegnare uno tra i due possibili valori 0 o 1 ad ogni elemento dei due insiemi, creando così un "portafoglio" la cui somma può assumere valori in $\{0, 1, \dots, 2n\}$. Un portafoglio si dice "adeguato" se per ogni individuo effettivamente qualificato per un lavoro o all'individuo o al lavoro è associato il valore 1 ed eventualmente ad entrambi.

La seguente trattazione ha il fine di mostrare che la più piccola somma possibile per un portafoglio adeguato coincide con il più grande numero di lavori che possono effettivamente essere svolti dagli n individui, ovvero la massima cardinalità di un matching nel grafo bipartizionato in individui e lavori, dove esiste un arco (i_k, j_ℓ) se e solo se l'individuo i_k è qualificato per il lavoro j_ℓ , per $k, \ell \in \{1, \dots, n\}$.

Si parte col dare la definizione di "trasversale", ovvero l'oggetto matematico corrispondente al "portafoglio adeguato" dell'esempio.

Definizione 2.1. Dato il grafo $G = (V, E)$, un sottoinsieme $T \subseteq V$ è un *trasversale* (o *vertex cover*) di G se ogni arco del grafo ha almeno un'estremità in T .

La minima cardinalità di un trasversale è denotata con $\tau(G)$.

Come si può vedere tale definizione vale in generale per qualsiasi grafo, non necessariamente bipartito. Nel caso in cui invece $G = (A \cup B, E)$ sia un grafo bipartito, allora sia l'insieme A che l'insieme B sono due possibili trasversali e pertanto si può concludere subito che $\tau(G) \leq \min\{|A|, |B|\}$.

Si vogliono ora cercare le relazioni nascoste tra matchings e trasversali di un generico grafo non orientato G .

Lemma 2.7. Per ogni matching M e trasversale T di un grafo $G = (V, E)$ si ha che $|M| \leq |T|$.

Dimostrazione. Per ogni arco del matching M almeno una delle due estremità deve appartenere al trasversale T . Poiché su ogni vertice di G incide al massimo un arco del matching, ne segue che $|M| \leq |T|$. \square

Conseguentemente si ha che $\nu(G) \leq \tau(G)$ è verificata per qualsiasi grafo G ed è immediato mostrare che esistono grafi per cui $\nu(G) < \tau(G)$, basti

pensare al triangolo K_3 . Se ci si restringe però al caso di grafi bipartiti vale l'uguaglianza. Questo risultato è essenzialmente dovuto a König:

Teorema di König 2.8. *Sia $G = (A \cup B, E)$ un grafo bipartito, allora $\nu(G) = \tau(G)$.*

Dimostrazione. Si è già provato, grazie al *lemma 2.7*, che $\nu(G) \leq \tau(G)$.

Sia quindi M un matching massimo di G e U l'insieme dei vertici M -esposti di A , si hanno allora due possibilità:

1. se $U = \emptyset$ ciò significa che $|M| = |A|$ e pertanto $|M| \geq \tau(G)$ in quanto è già stato osservato che per grafi bipartiti vale $\tau(G) \leq \min\{|A|, |B|\}$. Poiché M è massimo si ha allora che $|M| = \nu(G) \leq \tau(G) \leq |M|$ e in conclusione deve valere l'uguaglianza.
2. se invece $U \neq \emptyset$, si definisce $Z = \{v \in A \cup B \mid v \text{ è raggiungibile da un nodo di } U \text{ tramite un cammino } M\text{-alternante}\}$. Siano ora $S = Z \cap A$ e $K = Z \cap B$. Allora come nella dimostrazione del *teorema di Hall 2.1* si ha che ogni vertice di K è coperto dal matching M ed in particolare $K = N(S)$. Successivamente si definisce $T = (A \setminus S) \cup K$ (si veda *figura 2.4*). Ogni arco di G deve avere almeno un'estremità in T . Se così non fosse, allora esisterebbe un arco con un'estremità in S e una in $B \setminus K$ contraddicendo il fatto che $K = N(S)$. Quindi T risulta essere una vertex cover di G ed è chiaro che $|M| = |T|$, poiché per definizione di U e S segue che $A \setminus S$ sono nodi M -coperti, mentre $|S \setminus U| = |N(S)|$ è il numero dei rimanenti nodi coperti dal matching (si veda sempre *figura 2.4*). Come prima dato che $|M| = \nu(G) \leq \tau(G)$ si è allora dimostrato che T è una vertex cover minima di G e pertanto vale proprio l'uguaglianza $\nu(G) = \tau(G)$.

□

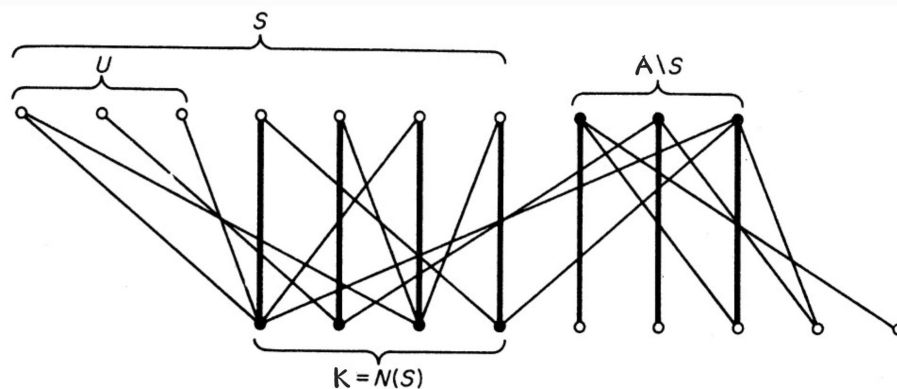


Figura 2.4: grafo esemplificativo per il teorema di König

La dimostrazione del *teorema di König* fornisce anche un algoritmo efficiente per trovare una vertex cover minima in un grafo bipartito dato $G = (A \cup B, E)$: applicando infatti l'algoritmo di Kuhn o di Hopcroft-Karp (discussi nelle sezioni precedenti) per determinare un matching massimo M , basta poi costruire T come nel secondo caso della dimostrazione precedente oppure (primo caso) porre direttamente $T = A$ per determinare così un trasversale di cardinalità minima in G .

2.4 Assegnamento pesato

Come già accennato nel preambolo del corrente capitolo, si è ora interessati ad una piccola variazione del problema dell'assegnamento che prende il nome di "*assegnamento pesato*" o che talvolta viene anche indicato con il titolo "*the General Assignment Problem*". Come suggerisce il nome stesso la differenza fondamentale è data dal fatto che vengono ora introdotti dei "*pesi*" associati agli archi del grafo bipartito.

L'esposizione iniziale del problema è tratta da [5], mentre i successivi risultati teorici sono dovuti al matematico Kuhn [6].

Per questa stessa istanza esistono in realtà due formulazioni diverse del medesimo problema:

1. PROBLEMA DEL MATCHING DI PESO MASSIMO

Istanza: Un grafo bipartito non orientato $G = (A \cup B, E)$ e pesi
 $c : E(G) \longrightarrow \mathbb{R}$

Obiettivo: Trovare un matching di peso massimo in G

2. PROBLEMA DEL MATCHING PERFETTO DI PESO MINIMO

Istanza: Un grafo bipartito non orientato $G = (A \cup B, E)$ e pesi
 $c : E(G) \longrightarrow \mathbb{R}$

Obiettivo: Trovare un matching perfetto di peso minimo in G o decidere che G non ha un matching perfetto.

Senza particolare sforzo è possibile far vedere che i due problemi sono tra loro equivalenti:

(2 \implies 1) data (G, c) un'istanza del PROBLEMA DEL MATCHING PERFETTO DI PESO MINIMO si definisce il parametro $K = \sum_{e \in E(G)} |c(e)|$ e si pone $c'(e) = K - c(e)$, $\forall e \in E(G)$. Allora qualsiasi matching M di peso massimo in (G, c') è anche un matching di cardinalità massima. Infatti se

per assurdo esistesse un matching M' tale che $|M'| > |M|$ allora

$$\sum_{e \in M'} |c'(e)| = |M'| \cdot K - \sum_{e \in M'} |c(e)| > |M| \cdot K - \sum_{e \in M} |c(e)| = \sum_{e \in M} |c'(e)|.$$

Contraddizione. Contro l'ipotesi che M sia un matching di peso massimo in (G, c') . Tale matching M fornisce in realtà una soluzione anche per il secondo problema, più precisamente si ha che:

- se $\nu(G) = |M| < \frac{|A \cup B|}{2}$ allora G non ammette un matching perfetto
- se invece $|M| = \frac{|A \cup B|}{2}$ allora M è un matching perfetto di (G, c) di peso minimo poiché la somma dei pesi c' degli archi di M è massima in (G, c') se e solo se è minima la somma dei pesi c degli stessi archi in (G, c) .

(1 \implies 2) Viceversa sia (G, c) un'istanza del PROBLEMA DEL MATCHING DI PESO MASSIMO e si aggiungano tutti gli archi mancanti in modo tale che il grafo G risulti essere completo. Si noti che, senza perdita di generalità, è sempre possibile assumere che tutti i pesi $c(e) \forall e \in E(G)$ dati siano positivi, in quanto eventuali archi di peso strettamente negativo possono essere rimossi fin da subito poiché non possono far parte di un matching di peso massimo, mentre archi di peso nullo possono far parte di un matching di peso massimo anche se certamente ne esiste uno che non li contiene. Si pone quindi $c'(e) = -c(e)$, $\forall e \in E(G)$ e $c'(e) = 0$ per tutti i nuovi archi e . Allora si può vedere facilmente come un matching perfetto di peso minimo in (G, c') , che sicuramente esiste poiché per costruzione il grafo è ora completo, corrisponde ad un matching di peso massimo in (G, c) , come si voleva.

Si osservi in particolare come tale equivalenza valga anche in generale nel caso di un grafo non orientato generico, non necessariamente bipartito, in quanto nella sua dimostrazione non vengono effettivamente mai utilizzate nozioni e proprietà proprie dei grafi bipartiti.

Quindi senza perdita di generalità nel seguito verrà considerato solo il PROBLEMA DEL MATCHING DI PESO MASSIMO su grafi bipartiti. Inoltre si assumerà sempre di avere un grafo bipartito $G = (A \cup B, E)$ completo con $|A| = |B| = n$ e dove gli archi e del grafo stesso saranno dotati di pesi interi positivi. Infatti, come è già stato osservato sopra, è sempre possibile assumere che i pesi c dati siano positivi ed eventualmente aggiungere archi di peso nullo al fine di completare il grafo.

Ovviamente sotto tali ipotesi un matching di cardinalità massima corrisponde ad un matching perfetto, nonché semplicemente una permutazione degli

elementi $\{1, \dots, n\}$, che verrà in seguito quasi sempre indicata con il titolo di "assegnamento".

Pertanto "the General Assignment Problem" richiede di determinare un opportuno assegnamento di peso massimo in G , cioè la permutazione

$$\begin{pmatrix} 1 & 2 & \dots & n \\ j_1 & j_2 & \dots & j_n \end{pmatrix}$$

tale per cui la somma $c_{1,j_1} + \dots + c_{n,j_n}$ sia massima.

È bene introdurre fin dal principio il suo problema duale, indispensabile per la formulazione di un efficiente algoritmo risolutore. Analogamente a quanto fatto nella sezione precedente, si sceglie di associare ad ogni nodo del grafo un "valore di portafoglio" intero e non-negativo che sarà indicato con i simboli a_i e b_j , $\forall i, j = 1, \dots, n$ per gli insiemi A e B rispettivamente. Si definisce un "portafoglio adeguato", per semplicità detto anche "cover", l'insieme dei valori $\{a_1, \dots, a_n, b_1, \dots, b_n\} = \{a_i, b_j\}$ per cui è sempre soddisfatta la relazione $a_i + b_j \geq c_{i,j}$, $\forall i, j \in \{1, \dots, n\}$. Il problema duale per il "General Assignment Problem" richiede allora di trovare una cover per cui sia minima la somma $a_1 + \dots + a_n + b_1 + \dots + b_n$, talvolta chiamata anche "prezzo del portafoglio". In formule si ha che:

PRIMALE

$$\begin{aligned} & \max \sum_{i=1}^n \sum_{j=1}^n c_{i,j} x_{i,j} \\ \text{s.a. } & \sum_{j=1}^n x_{i,j} = 1, \forall i \in A \\ & \sum_{i=1}^n x_{i,j} = 1, \forall j \in B \\ & x_{i,j} \in \{0, 1\}, \forall (i, j) \in E \end{aligned}$$

DUALE

$$\begin{aligned} & \min \sum_{i=1}^n a_i + \sum_{j=1}^n b_j \\ \text{s.a. } & a_i + b_j \geq c_{i,j}, \forall (i, j) \in E \end{aligned}$$

Per giustificare e completare tale procedura si mostrano ora la seguenti osservazioni:

Lemma 2.9. *Il prezzo totale di un qualsiasi portafoglio adeguato non supera la somma dei pesi sugli archi di un qualunque assegnamento.*

Dimostrazione. Si consideri un generico assegnamento $\begin{pmatrix} 1 & 2 & \dots & n \\ j_1 & j_2 & \dots & j_n \end{pmatrix}$ e un qualsiasi portafoglio adeguato $\{a_i, b_j\}$. Dalle proprietà di quest'ultimo segue allora che: $a_1 + b_{j_1} \geq c_{1,j_1}, \dots, a_n + b_{j_n} \geq c_{n,j_n}$. In conclusione sommando

queste disuguaglianze si ottiene: $a_1 + b_{j_1} + \dots + a_n + b_{j_n} \geq c_{1,j_1} + \dots + c_{n,j_n}$, che verifica la tesi. \square

Da quest'ultimo risultato segue immediatamente che se è possibile esibire un portafoglio adeguato tale per cui il suo prezzo eguaglia la somma dei pesi sugli archi di un matching perfetto, allora entrambi risulteranno essere contemporaneamente soluzione per il problema dell'assegnamento pesato ed il suo duale.

Si vuole ora mostrare che è sempre possibile determinare un siffatto portafoglio ed assegnamento.

Prima di continuare, però, è opportuno introdurre la seguente nozione: un qualsiasi "accoppiamento" (i, j) , per $i, j \in \{1, \dots, n\}$ è detto *ammissibile* se vale che $a_i + b_j = c_{i,j}$, altrimenti verrà detto *inammissibile*.

Risulta immediato verificare che:

Teorema 2.10. *Data una cover per il grafo bipartito $G = K_{n,n}$, se è possibile determinare un matching perfetto M in cui ogni arco (i, j) del matching risulta essere ammissibile, allora matching e cover sono soluzioni ottime del problema primale e duale rispettivamente, inoltre la somma dei pesi sugli archi di M coincide con il prezzo della cover.*

Dimostrazione. Segue direttamente dalle ipotesi che: $a_1 + b_{j_1} = c_{1,j_1}, \dots, a_n + b_{j_n} = c_{n,j_n}$, per il dato matching perfetto $M = \begin{pmatrix} 1 & 2 & \dots & n \\ j_1 & j_2 & \dots & j_n \end{pmatrix}$.

Sommando tutte queste uguaglianze si ottiene infine la tesi:

$$a_1 + b_1 + \dots + a_n + b_n = c_{1,j_1} + \dots + c_{n,j_n} \quad \square$$

Riassumendo sia $G = K_{n,n}$ un grafo bipartito completo e $c : E(G) \rightarrow \mathbb{R}^+$ una funzione peso. Dato un portafoglio adeguato $\{a_1, \dots, a_n, b_1, \dots, b_n\}$ se esiste un matching perfetto M tale che $a_i + b_j = c_{i,j}, \forall (i, j) \in M$, allora si ha già una soluzione ottima. Altrimenti è possibile determinare una procedura che riduca ad ogni iterazione il prezzo del portafoglio iniziale.

Osservazione 2. Si osserva innanzitutto che, senza perdita di generalità, è sempre possibile assumere di avere una cover $\{a_i, b_j\}$ in cui $a_i > 0, \forall i = 1, \dots, n$.

Teorema 2.11. *Data una cover per il grafo bipartito $G = K_{n,n}$, se per qualsiasi matching perfetto M di G il numero degli archi ammissibili di M è al massimo $m < n$, allora è possibile costruire una nuova cover di prezzo inferiore alla precedente.*

Dimostrazione. Sia M un matching perfetto di G tale che $|\{(i, j) \in M \mid a_i + b_j = c_{i,j}\}| = m < n$.

Si consideri ora $N = \{(i, j) \in E \mid a_i + b_j = c_{i,j}\}$, ovvero l'insieme degli archi ammissibili di G , e il grafo ausiliario $G' = (A \cup B, N)$. Ovviamente l'insieme $M' = \{(i, j) \in M \mid a_i + b_j = c_{i,j}\}$ è un matching massimo di G' . Riprendendo la dimostrazione del *teorema di König 2.8* si definisce U come l'insieme dei nodi M' -esposti di A , per cui sicuramente $U \neq \emptyset$ poiché M' non è perfetto. Siano allora $Z = \{v \in A \cup B \mid v \text{ è raggiungibile da un nodo di } U \text{ tramite un cammino } M'\text{-alternante}\}$, $S = Z \cap A$ e $K = Z \cap B = N(S)$. Come si è già mostrato l'insieme $T = (A \setminus S) \cup K$ risulta essere un trasversale di cardinalità minima in G' . Se $|A \setminus S| = s$ e $|K| = k$ allora $s + k = |T| = |M'| = m$. Si decide quindi di costruire la nuova cover come segue:

$$\begin{cases} a'_i = a_i, \forall i \in A \setminus S \\ a'_i = a_i - 1, \forall i \in S \\ b'_j = b_j + 1, \forall j \in K \\ b'_j = b_j, \forall j \in B \setminus K \end{cases} .$$

Dato che si è assunto $a_i > 0, \forall i = 1, \dots, n$ (*osservazione 2*) allora ogni a_i risulta essere ancora un intero positivo. Si deve quindi verificare che:

1. il nuovo portafoglio è ancora adeguato
2. il prezzo totale è diminuito

L'unica possibilità in cui la disuguaglianza $a'_i + b'_j \geq c_{i,j}$ potrebbe venir meno si ha quando l'elemento a_i è stato diminuito di un'unità e b_j è rimasto inalterato. Supponendo quindi per assurdo che $a'_i + b'_j = a_i - 1 + b_j < c_{i,j}$ e pertanto $a_i + b_j = c_{i,j}$ per definizione di cover iniziale, allora l'arco (i, j) sarebbe stato ammissibile nel grafo G' . *Contraddizione*. Poiché o $i \in T$ oppure $j \in T$, essendo T un trasversale per G' , e quindi il caso in esame non può mai verificarsi, in quanto o $a'_i = a_i$ oppure $b'_j = b_j + 1$.

Infine la somma totale della cover è stata diminuita di $n - s$ unità ed aumentata di altre k , dunque in sostanza il valore finale risulterà essere inferiore al precedente di $n - (s + k) = n - m > 0$ unità, come volevasi dimostrare. \square

Nella sezione successiva tutte queste fondamentali nozioni verranno riprese ed assemblate, al fine di comporre un algoritmo in grado di risolvere il problema dell'assegnamento pesato.

2.5 Metodo ungherese

Si è appena discusso il problema dell'assegnamento pesato, per il quale si cercherà ora di descrivere un efficiente algoritmo risolutore, storicamente

proposto per primo dal matematico Kuhn nel 1955 [6]. All'epoca il risultato di Kuhn fu notevole, poiché nonostante l'elevato numero di soluzioni alternative (ovvero $n!$ permutazioni) era possibile trovare l'ottimo in tempi ragionevoli pari ad un $O(n^4)$ nella sua analisi originale, e poi nel tempo ridotto ad un $O(n^3)$. Il procedimento adottato può essere descritto come una tecnica primale-duale, cioè un metodo di massima ascesa per il problema duale. Siccome venivano utilizzate proprietà dei grafi bipartiti introdotte da matematici ungheresi (illustrate nelle *sezioni 2.1* e *2.3* della presente trattazione) tale algoritmo fu soprannominato con l'epiteto di "*metodo ungherese*".

L'esposizione dell'algoritmo ungherese viene fatta riprendendo i concetti presentati nella sezione precedente. Sia quindi $G = (A \cup B, E) = K_{n,n}$ un grafo bipartito completo e sia $c : E(G) \rightarrow \mathbb{R}^+$ una funzione peso. Innanzitutto si decide di indicare con C la matrice quadrata $n \times n$ contenente tutti i pesi associati agli archi, dove ovviamente le n righe e colonne corrispondono ai nodi di A e B rispettivamente. A partire dal grafo G è sempre possibile costruire in maniera abbastanza elementare una cover iniziale: fissata la riga $i \in \{1, \dots, n\}$ sia $u_i = \max_{j=1, \dots, n} c_{i,j}$ e, analogamente, fissata la colonna $j \in \{1, \dots, n\}$ sia $v_j = \max_{i=1, \dots, n} c_{i,j}$. Si definiscono ulteriormente le quantità $u = \sum_{i=1}^n u_i$ e $v = \sum_{j=1}^n v_j$ per cui si ha che:

1. se $u \leq v$ si pone
$$\begin{cases} a_i = u_i, & \text{per } i = 1, \dots, n \\ b_j = 0, & \text{per } j = 1, \dots, n \end{cases}$$
2. se $u > v$ si pone
$$\begin{cases} a_i = 0, & \text{per } i = 1, \dots, n \\ b_j = v_j, & \text{per } j = 1, \dots, n \end{cases}$$

Allora l'insieme $\{a_1, \dots, a_n, b_1, \dots, b_n\}$ così definito risulta essere effettivamente una cover per il grafo G . La scelta di una siffatta cover è in parte giustificata dal fatto che, a partire da tali valori, è possibile mostrare come l'algoritmo ungherese termini rapidamente, restituendo così il risultato corretto per entrambi i problemi dell'assegnamento pesato ed il suo duale.

Successivamente data una cover $\{a_i, b_j\}$ per $G = (A \cup B, E)$, si denota con $N = \{(i, j) \in E \mid a_i + b_j = c_{i,j}\}$, ovvero l'insieme degli archi "*ammissibili*" di G . Si consideri ora il *grafo ausiliario* $G' = (A \cup B, N)$. Per quanto mostrato nelle *sezioni 2.1* e *2.2* è sempre possibile determinare in maniera efficiente un matching massimo M in G' . Se $|M| = n$ allora tale matching è anche perfetto in G ed in particolare corrisponde all'assegnamento cercato (*teorema 2.10*). Altrimenti è sempre possibile diminuire la somma totale della cover aumentando così il numero di archi ammissibili, per poi arrivare a soddisfare una tale condizione (*teorema 2.11*). Dato che il numero degli archi di G è

finito, tale procedura termina sempre restituendo la soluzione ottima.

Sia quindi M un matching massimo di G' tale per cui $|M| < n$. Per ogni arco $(u, v) \in M$ si sceglie di "etichettare" i suoi estremi come segue: se esiste un altro arco in G' avente sempre un'estremità in $u \in A$ e l'altra in $w \in B \setminus \{v\}$ dove w è un nodo M -esposto, allora u è detto "essenziale". In tutti gli altri casi sarà invece il nodo v ad essere indicato come tale.

Osservazione 3. Ne segue subito che: per ogni arco (u, v) del matching M , allora o v oppure u è un nodo essenziale, e non entrambi. Inoltre per ogni arco di G' almeno uno dei due estremi è essenziale. Questi risultati sono in realtà una diretta applicazione del *teorema di König 2.8* (sulla falsa riga di quanto fatto nella dimostrazione del *teorema 2.11*) dove l'insieme dei vertici così detti "essenziali" forma un trasversale di cardinalità minima (poiché coincide con quella del matching massimo M) per G' .

In conclusione un qualsiasi nodo di G' che non risulta essere essenziale sarà ovviamente indicato come "inessenziale".

Ritornando ora a focalizzare l'attenzione sul grafo G e la matrice dei pesi C , si vuole mostrare com'è possibile determinare un'altra cover per G di peso inferiore. Siano $A' = \{i \in A \mid i \text{ è inessenziale}\}$ e $B' = \{j \in B \mid j \text{ è inessenziale}\}$. Si calcoli il parametro $d = \min\{a_i + b_j - c_{i,j} \mid i \in A' \text{ e } j \in B'\}$, dove sicuramente $d \geq 0$ per definizione di cover. Se per assurdo $d = 0$ allora esisterebbe almeno una coppia (i, j) con $i \in A'$ e $j \in B'$ tale che $a_i + b_j - c_{i,j} = 0$, ovvero l'arco (i, j) sarebbe ammissibile in G' e pertanto almeno uno tra i e j dovrebbe essere un nodo essenziale per l'*osservazione 3* fatta sopra. *Contraddizione.* Si ha quindi $d > 0$.

A seconda delle caratteristiche della cover in esame, è possibile considerare uno tra i due seguenti casi che si escludono a vicenda:

- PRIMO CASO: se $\forall i \in A'$ si ha che $a_i > 0$, allora sia $m = \min\{d, \min\{a_i \mid i \in A'\}\}$ e si modifichi la cover $\{a_i, b_j\}$ ponendo:

$$\begin{cases} a_i \leftarrow a_i - m, \forall i \in A' \\ b_j \leftarrow b_j + m, \forall j \in B \setminus B' \end{cases}$$

- SECONDO CASO: se $\exists i \in A'$ tale che $a_i = 0$, allora sia $m = \min\{d, \min\{b_j \mid j \in B'\}\}$ e si modifichi la cover $\{a_i, b_j\}$ ponendo:

$$\begin{cases} a_i \leftarrow a_i + m, \forall i \in A \setminus A' \\ b_j \leftarrow b_j - m, \forall j \in B' \end{cases}$$

È possibile verificare che dopo tali modifiche si ha ancora una cover per G la cui somma $a_1 + \dots + a_n + b_1 + \dots + b_n$ risulta essere inferiore alla precedente. Inoltre anche l'insieme degli archi ammissibili, ovvero per i quali vale l'uguaglianza $a_i + b_j = c_{i,j}$ con $i, j \in \{1, \dots, n\}$, risulta essere aumentato di almeno un'unità.

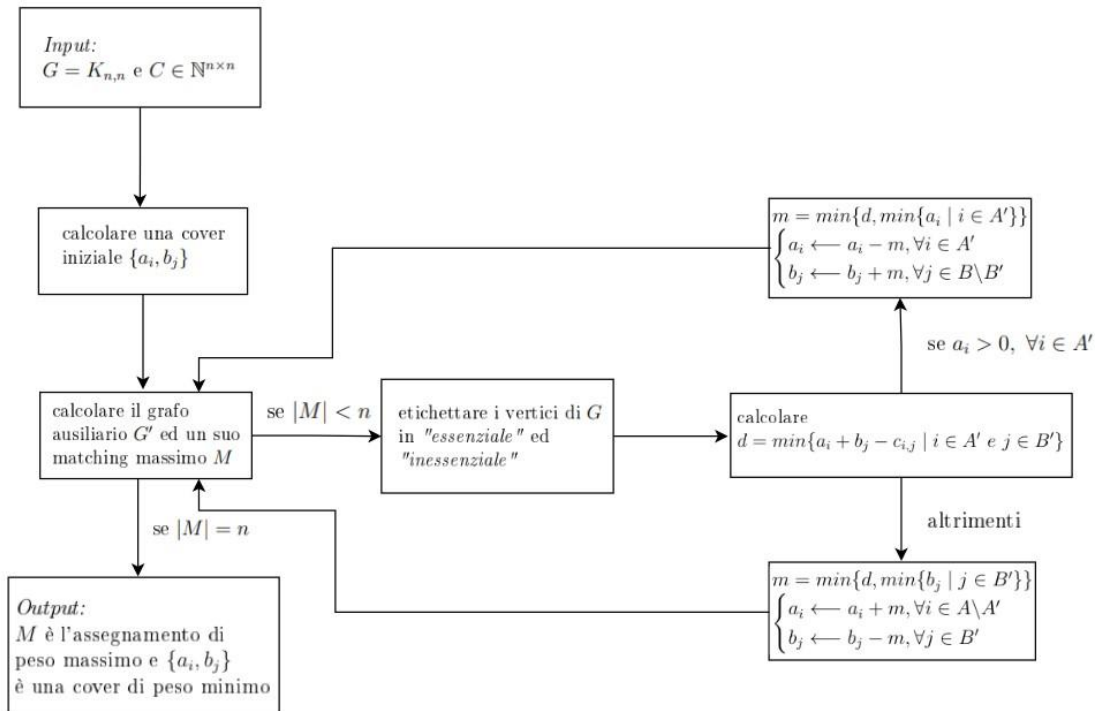


Figura 2.5: schema algoritmo ungherese

UN ESEMPIO

Con il seguente esempio, sebbene di piccole dimensioni, si cerca di mostrare tutte le possibilità che possono occorrere nell'algoritmo.

Si consideri il grafo bipartito $G = K_{4,4}$ a cui è associata la matrice dei pesi

$$C = \begin{pmatrix} 8 & 7 & 9 & 9 \\ 5 & 2 & 7 & 8 \\ 6 & 1 & 4 & 9 \\ 2 & 3 & 2 & 6 \end{pmatrix}.$$

Il primo passo consiste nel costruire una cover iniziale per G :

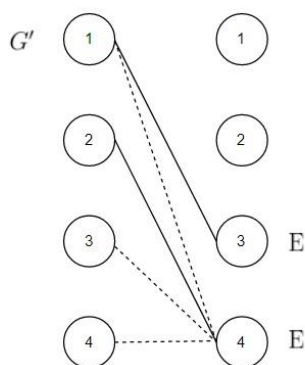
- $u_i = \max_{j=1, \dots, 4} c_{i,j}, \forall i = 1, \dots, 4 \implies u_1 = 9, u_2 = 8, u_3 = 9, u_4 = 6$
 $\implies u = \sum_{i=1}^4 u_i = 9 + 8 + 9 + 6 = 32$

- $v_j = \max_{i=1, \dots, 4} c_{i,j}, \forall j = 1, \dots, 4 \implies v_1 = 8, v_2 = 7, v_3 = 9, v_4 = 9 \implies v = \sum_{j=1}^4 v_j = 8 + 7 + 9 + 9 = 33$

Dato che $u < v$, allora si pone: $\begin{cases} a_i = u_i, \forall i = 1, \dots, 4 \\ b_j = 0, \forall j = 1, \dots, 4 \end{cases}$ determinando così

una cover iniziale per G : $\{9, 8, 9, 6, 0, 0, 0, 0\}$.

Ha quindi inizio il processo ricorsivo:



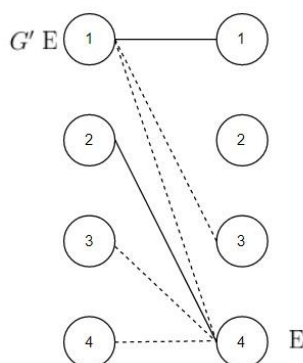
$$C = \begin{pmatrix} \boxed{0} & \boxed{0} & 0 & 0 \\ 8 & 7 & 9 & 9 \\ 5 & 2 & 7 & 8 \\ 6 & 1 & 4 & 9 \\ 2 & 3 & 2 & 6 \end{pmatrix} \begin{matrix} 9 \\ 8 \\ 9 \\ 6 \end{matrix}$$

$$d = \min\{1, 2, 3, 6, 3, 8, 4, 3\} = 1$$

$$m = \min\{1, 9, 8, 9, 6\} = 1$$

$$\begin{cases} a_1 \rightarrow 9 - 1 = 8 \\ a_2 \rightarrow 8 - 1 = 7 \\ a_3 \rightarrow 9 - 1 = 8 \\ a_4 \rightarrow 6 - 1 = 5 \\ b_3 \rightarrow 0 + 1 = 1 \\ b_4 \rightarrow 0 + 1 = 1 \end{cases}$$

$\{8, 7, 8, 5, 0, 0, 1, 1\}$ è la nuova cover



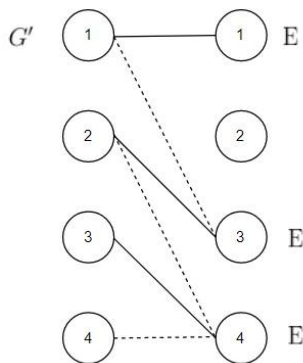
$$C = \begin{pmatrix} \boxed{0} & \boxed{0} & \boxed{1} & 1 \\ 8 & 7 & 9 & 9 \\ 5 & 2 & 7 & 8 \\ 6 & 1 & 4 & 9 \\ 2 & 3 & 2 & 6 \end{pmatrix} \begin{matrix} 8 \\ 7 \\ 8 \\ 5 \end{matrix}$$

$$d = \min\{2, 5, 1, 2, 7, 5, 3, 2, 4\} = 1$$

$$m = \min\{1, 7, 8, 5\} = 1$$

$$\begin{cases} a_2 \rightarrow 7 - 1 = 6 \\ a_3 \rightarrow 8 - 1 = 7 \\ a_4 \rightarrow 5 - 1 = 4 \\ b_4 \rightarrow 1 + 1 = 2 \end{cases}$$

$\{8, 6, 7, 4, 0, 0, 1, 2\}$ è la nuova cover



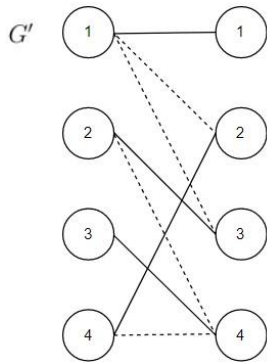
$$C = \begin{pmatrix} 0 & \boxed{0} & 1 & 2 \\ 8 & 7 & 9 & 9 \\ 5 & 2 & 7 & 8 \\ 6 & 1 & 4 & 9 \\ 2 & 3 & 2 & 6 \end{pmatrix} \begin{matrix} 8 \\ 6 \\ 7 \\ 4 \end{matrix}$$

$$d = \min\{1, 4, 6, 1\} = 1$$

$$m = \min\{1, 8, 6, 7, 4\} = 1$$

$$\begin{cases} a_1 \rightarrow 8 - 1 = 7 \\ a_2 \rightarrow 6 - 1 = 5 \\ a_3 \rightarrow 7 - 1 = 6 \\ a_4 \rightarrow 4 - 1 = 3 \\ b_1 \rightarrow 0 + 1 = 1 \\ b_3 \rightarrow 1 + 1 = 2 \\ b_4 \rightarrow 2 + 1 = 3 \end{cases}$$

$\{7, 5, 6, 3, 1, 0, 2, 3\}$ è la nuova cover



Output:

- $M = \{(1, 1), (2, 3), (3, 4), (4, 2)\}$ è un matching perfetto di peso massimo: $\sum_{e \in M} c(e) = 8 + 7 + 9 + 3 = 27$
- $\{a_i, b_j\} = \{7, 5, 6, 3, 1, 0, 2, 3\}$ è una cover per G di peso minimo:
 $\sum_{i=1}^4 a_i + \sum_{j=1}^4 b_j = 7 + 5 + 6 + 3 + 1 + 0 + 2 + 3 = 27$

Ovviamente matching M e cover $\{a_i, b_j\}$ hanno lo stesso peso totale essendo di fatto soluzione ottima del problema primale e duale rispettivamente.

Capitolo 3

Algoritmo del matching massimo di Edmonds

Nel capitolo precedente è stato possibile discutere in maniera precisa e rigorosa il problema del matching massimo ristretto al caso di grafi bipartiti. Si è ora invece interessati ad estendere tali ragionamenti anche al caso di grafi generali, non necessariamente bipartiti. Verranno quindi inizialmente introdotte alcune nozioni preliminari, necessarie per una completa comprensione e formulazione di un efficiente algoritmo risolutore, che verrà analizzato e descritto in seguito.

Si ricordi il *teorema di Berge 1.5* dove essenzialmente un matching in un dato grafo è massimo se e solo se non esiste nessun cammino aumentante. L'algoritmo di matching massimo realizzato ad opera del matematico Jack Edmonds [4], sarà ancora ovviamente basato su questo fondamentale risultato. Riprendendo le considerazioni esposte alla fine della *sezione 1.3*, dato un grafo non orientato G e un matching M , non era ancora chiaro come trovare dei cammini aumentanti, o decidere in caso che non ce ne fossero. Come si è visto successivamente nella *sezione 2.1*, nel caso bipartito a partire da un vertice M -esposto era sufficiente segnare i vertici raggiungibili da una progressione alternante di archi. Poiché infatti in grafi bipartiti non sono presenti cicli dispari, allora i vertici raggiungibili da una tale progressione risultano essere anche raggiungibili proprio da un cammino alternante. Ciò però non è più valido quando si considera un grafo qualsiasi. Il problema è infatti cosa fare nel momento in cui ci si imbatte in un ciclo dispari. Sorprendentemente, basta "*rimuoverlo*" contraendo i suoi vertici ed archi in un singolo vertice. Questa è l'idea fondamentale dell'algoritmo del matching di cardinalità massima di Jack Edmond che verrà esposta ed analizzata in maniera precisa e dettagliata, analogamente a quanto fatto dallo stesso Edmonds in [4], nel presente capitolo.

3.1 Alberi e foreste alternanti

Cercando di legare il concetto di albero, quale grafo connesso ed aciclico, a quello di cammino alternante ed aumentante, si presentano ora le seguenti definizioni e risultati chiave.

Definizione 3.1. Un *albero alternato* $J = (V, E)$ è un albero i cui vertici sono etichettati in "esterni" ed "interni" e dove ogni arco $e \in E$ collega per l'appunto un vertice esterno ad uno interno in modo tale che ogni vertice interno di J sia estremo di esattamente due archi.

Osservazione 4. L'insieme dei vertici di un albero alternato $J = (V, E)$ risulta essere quindi partizionato in nodi interni ed esterni che, per semplicità di notazione, saranno indicati con i simboli VI e VE rispettivamente. In particolare vale che $VI \cup VE = V$ mentre $VI \cap VE = \emptyset$.

Lemma 3.1. *Ogni albero alternato contiene un vertice esterno in più rispetto ai vertici interni.*

Dimostrazione. Sia $J = (VI \cup VE, E)$ un albero alternato. Si vuole mostrare che $|VE| = |VI| + 1$. Dalla definizione di albero alternato segue direttamente che $\forall v \in VI$, allora $d(v) = 2$. Per ogni nodo interno del grafo si sceglie quindi di rimuoverlo unendo i suoi due archi incidenti a formarne uno unico e ottenendo così un nuovo albero $T = (VE, E')$. Tale grafo continua ancora ad essere un albero, poiché l'operazione di rimozione appena descritta non genera cicli e non ne modifica la proprietà di connettività, in quanto i cammini precedenti che collegavano tra loro due vertici esterni continuano ad esistere ma solo più corti. Si osserva in particolare che ad ogni arco di T è associato un vertice interno di J e pertanto deve valere $|E'| = |VI|$. Grazie alla *proposizione 1.2* si conclude quindi che: $|VE| = |E'| + 1 = |VI| + 1$, come volevasi dimostrare. \square

Con il precedente lemma si è anche dimostrato che il numero di vertici di un qualsiasi albero alternato J è sempre dispari e pertanto J non ammette mai un matching perfetto. Allora un qualsiasi matching di J può avere cardinalità al massimo pari a $\frac{|V(J)|-1}{2}$ che corrisponde in particolare al numero dei suoi vertici interni. Si conclude quindi che se esiste un matching in J di cardinalità pari a $|VI|$, allora tale matching è anche massimo in J .

Osservazione 5. Sia $T = (V, E)$ un albero, allora $\forall e \in E$ si ha che il nuovo grafo $T \setminus \{e\}$ è una foresta formata da due alberi.

Questa osservazione tornerà utile nella dimostrazione del lemma successivo.

Lemma 3.2. *Per ogni fissato vertice esterno v di un albero alternato J , esiste un unico matching massimo M che copre tutti i nodi di J tranne il vertice v . Inoltre ogni matching massimo di J lascia un nodo esterno esposto.*

Dimostrazione. Si dimostra per induzione sul numero di vertici n dell'albero. Se $n = 1$, allora non ci sono archi nell'albero e quindi l'unico nodo è esterno ed è anche M -esposto.

Sia quindi $n > 1$. Si procede col rimuovere un qualsiasi nodo interno u del grafo J ottenendo così due alberi alternati J_1 e J_2 (tale proprietà discende dall'osservazione 5), dove si suppone che sia l'albero J_1 a contenere il vertice v come vertice esterno. Per ipotesi induttiva esiste un unico matching massimo M_1 di J_1 che lascia v come unico nodo esposto. Inoltre indicato con w il nodo esterno di J_2 che risultava essere collegato con il nodo interno u nel grafo di partenza, applicando di nuovo l'ipotesi induttiva è possibile determinare anche un matching massimo M_2 per J_2 che lascia esposto solamente il nodo w . Allora l'unione $M_1 \cup (u, w) \cup M_2$ forma un matching massimo in J che lascia esposto solamente il nodo v . In conclusione, dato che ogni arco di J ha come estremi un vertice interno ed uno esterno e $|VE| = |VI| + 1$ per il lemma 3.1, allora ogni matching massimo di J lascia un solo vertice esterno come esposto. \square

A partire da quest'ultimo risultato segue in maniera "naturale" la prossima definizione.

Definizione 3.2. Dato un grafo G e un matching M di G , un albero alternato J contenuto in G è detto anche "alternante" se $M \cap J$ è un matching massimo per J e inoltre l'unico vertice r esposto per $M \cap J$ è anche esposto per M . Pertanto tutti gli archi del matching M che toccano J sono direttamente contenuti in J , ed in particolare il nodo r è detto "radice" dell'albero alternante $J(M)$.

Osservazione 6. Anche un singolo vertice M -esposto forma di per sè un albero alternante, poiché ne soddisfa la definizione.

Lemma 3.3. *Dato un grafo non orientato G e un matching M di G , sia $J(M) \subseteq G$ un albero alternante. Allora ogni cammino alternante $P(M) \subseteq J(M)$ che ha un nodo esterno v ed un arco del matching incidente a v in una delle due estremità, è un sotto-cammino del cammino alternante $P_v(M) \subseteq J(M)$ che collega il vertice v alla radice r .*

Dimostrazione. Per $k \geq 1$, sia P_{2k-1} l'unico cammino $P(M)$ formato da $2k-1$ archi e che quindi all'estremità opposta a v presenta un vertice interno u_k ed un arco del matching. Infatti i nodi di qualsiasi cammino alternante in J

formano una sequenza alternata di vertici esterni ed interni. Dato che $d(u_k) = 2$, sia allora P_{2k} l'unico cammino che è possibile ottenere aggiungendo un arco non-del-matching a P_{2k-1} . Pertanto come nuovo estremo del cammino P_{2k} si avrà un vertice esterno v_k . Se $v_k = r$ si è concluso, altrimenti è ancora possibile aumentare il cammino P_{2k} aggiungendo l'unico arco del matching che copre v_k (ovviamente v_k non è M -esposto, poiché solo la radice lo è nell'albero J). Dato che tale procedura è verificata per $k = 1$ ed inoltre il numero degli archi di J è finito, allora la tesi segue per induzione. \square

Definizione 3.3. Dato un grafo G ed un matching M in G , si definisce uno "*stem*" come o un singolo vertice M -esposto, oppure un cammino M -alternante con un vertice esposto in un'estremità ed un arco del matching in corrispondenza dell'altra. Il vertice esposto ed il nodo all'estremità opposta del cammino sono chiamati rispettivamente *radice* e *punta* dello stem.

Alla luce di quest'ultima definizione, il lemma precedente (*lemma 3.3*) afferma che è sempre possibile determinare il cammino alternante $P_v \subseteq J(M)$ che collega un qualsiasi nodo esterno v alla radice r di J , e che tale cammino risulta essere effettivamente uno stem in (G, M) .

Definizione 3.4. Siano $G = (V, E)$ un grafo non orientato e M un matching di G . Si definisce un *albero aumentante* $J_A(M)$ come l'unione di un albero alternante $J(M) \subseteq G$ ed un arco $e \in E$ incidente un vertice esterno $v \in V(J)$ ed un nodo M -esposto $u \in V(G) \setminus V(J)$.

Dalla *definizione 3.4* e dal *lemma 3.3* segue immediatamente che l'unione del cammino P_v in $J_A(M)$ che collega il vertice v alla radice r dell'albero, e dell'arco $e = (v, u)$, forma un cammino M -aumentante in G , in quanto entrambi gli estremi r e u del cammino risultano essere vertici M -esposti.

Definizione 3.5. Dato un grafo non orientato G e un albero alternato $J \subseteq G$, allora J è detto "*albero ungherese*" se ogni suo vertice esterno è adiacente a soli vertici interni.

Analogamente a quanto è già stato fatto, è possibile estendere le nozioni precedenti con il concetto di foresta, ovvero un grafo le cui componenti connesse sono alberi. Di rilevante importanza sarà in particolare la seguente definizione.

Definizione 3.6. Siano dati un grafo non orientato G e un matching M in G , una *foresta alternante* rispetto ad M in G è una foresta le cui componenti connesse sono alberi alternanti.

Una foresta alternante verifica quindi le seguenti proprietà:

1. ogni componente connessa contiene esattamente un vertice M -esposto, ovvero la radice dell'albero alternante
2. una tale foresta contiene tutti i vertici M -esposti di G , infatti anche un singolo vertice non coperto dal matching forma di per sè un albero alternante
3. i suoi vertici possono essere suddivisi in "*esterni*" ed "*interni*" in modo tale che:
 - ogni vertice esterno abbia distanza pari dalla radice della componente connessa che lo contiene (in particolare le radici sono vertici esterni)
 - ogni vertice interno abbia distanza dispari dalla radice della componente connessa che lo contiene
 - tutti i vertici interni abbiano grado 2
 - il numero di vertici esterni che non sono una radice è uguale al numero di vertici interni
4. per qualsiasi nodo v della foresta alternante, allora l'unico cammino da v alla radice della stessa componente connessa che contiene v risulta essere un cammino M -alternante.

Queste proprietà discendono direttamente da tutte le osservazioni precedentemente dimostrate nella presente sezione.

In particolare, dato un grafo G e un matching M di G , allora l'insieme di tutti i nodi M -esposti di G è una foresta M -alternante.

Definizione 3.7. Una *foresta ungherese* in G è definita similmente ad un albero ungherese scambiando il termine "*albero*" con quello di "*foresta*".

Si noti però che gli alberi che formano una foresta ungherese non sono necessariamente alberi ungheresi. Infatti un vertice esterno di un albero alternante potrebbe essere adiacente anche ad un vertice interno di un diverso albero della foresta, e quindi non solo con i nodi interni della stessa componente connessa.

In conclusione i risultati sugli alberi presentati in questa sezione sono essenzialmente gli stessi per le foreste.

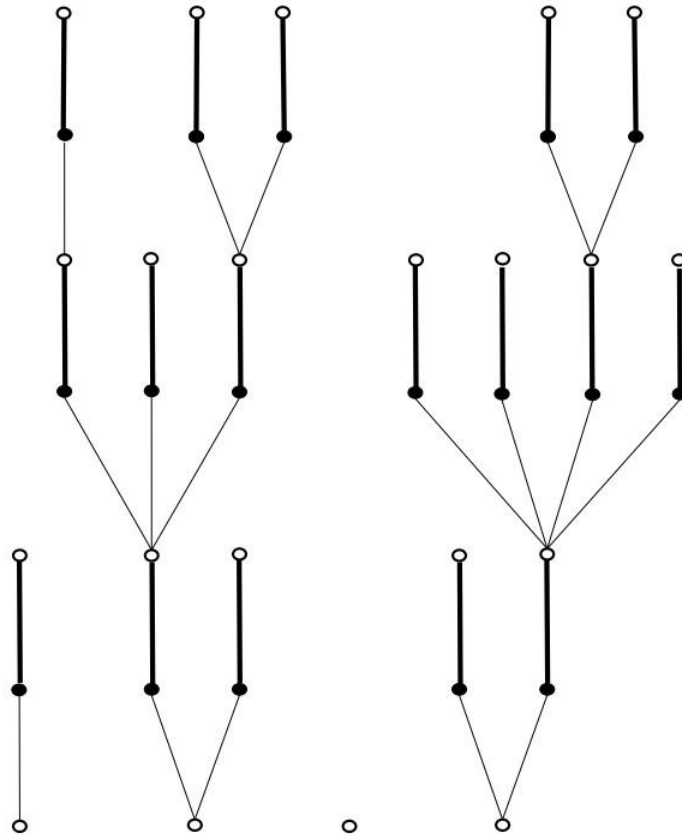


Figura 3.1: esempio di una foresta alternante, i nodi *esterni* vengono indicati con un pallino pieno, quelli *interni* con un pallino vuoto. Gli archi del matching sono evidenziati con le linee in grassetto.

3.2 Fiori e blossoms

In questa sezione si presentano le definizioni di "*blossom*" e "*fiore*", le loro proprietà e primi risultati. L'introduzione di questi nuovi oggetti, come sarà subito chiaro in seguito, ha lo scopo di gestire e risolvere la problematica data dalla presenza di eventuali cicli dispari all'interno del grafo. Si osserva innanzitutto che dato un ciclo dispari $B = (V(B), E(B))$ e un nodo $b \in V(B)$, allora esiste un unico matching massimo M in B per cui il nodo b risulti essere M -esposto.

Definizione 3.8. Dato un grafo non orientato $G = (V, E)$ e un matching M di G , si definisce un *blossom* $B(M)$ di G come un ciclo dispari contenuto in G e tale che $M \cap B$ sia un matching massimo di B .

Definizione 3.9. Un *fiore* $F(M)$ è formato dall'unione di un blossom B e uno stem P , intersecantisi solo in un nodo b che coincide sia con il vertice M -esposto del blossom, sia con la punta dello stem.

Definizione 3.10. Un *albero fiorito* J_F in (G, M) consiste nell'unione di un albero alternante $J(M)$ ed un arco e di G incidente due vertici esterni dell'albero J .

Da queste prime definizioni segue immediatamente il prossimo risultato:

Lemma 3.4. *In un albero fiorito $J_F(M)$ di (G, M) , l'unione dell'arco e di estremi i vertici esterni v_1, v_2 di J_F e dei due (unici) cammini P_1, P_2 che collegano rispettivamente i nodi esterni v_1 e v_2 alla radice r di J , forma un fiore $F(M)$ in G .*

Dimostrazione. Dato che i due cammini P_1 e P_2 si intersecano almeno nella radice r e che in J qualsiasi cammino che collega r ad un altro vertice dell'albero è unico, allora è ben definito il cammino alternante $P = P_1 \cap P_2$. Sicuramente tale cammino ammette la radice r in una delle due estremità. Sia quindi b il vertice corrispondente all'altra estremità di P . Se per assurdo b fosse un vertice interno di J allora sarebbe innanzitutto distinto da r, v_1 e v_2 . Di conseguenza anche r risulta essere diverso dai nodi v_1 e v_2 . Pertanto sarebbero incidenti in b un arco di P , uno di P_1 ma non di P , ed uno di P_2 ma non di P né di P_1 . *Contraddizione.* Poiché per ogni vertice interno v di J vale invece che $d(v) = 2$. Quindi b è un vertice esterno di J e per quanto è già stato osservato nella sezione precedente, il cammino P che collega b alla radice r è uno stem in G (*lemma 3.3*).

Si considerino ora i cammini $P'_1 = P_1 \setminus (P \setminus \{b\})$ e $P'_2 = P_2 \setminus (P \setminus \{b\})$. A meno che $v_1 = b$ oppure $v_2 = b$, tali cammini hanno un arco non del matching in corrispondenza dell'estremità b , mentre entrambi i nodi v_1 e v_2 sono M -coperti. Segue allora che in qualsiasi caso il sottografo $B = P'_1 \cup P'_2 \cup e$ è un ciclo dispari in G , dove b è il suo unico vertice M -esposto, ovvero B è un blossom in G .

In conclusione si è quindi mostrato che l'unione del blossom B e dello stem P forma effettivamente un fiore in G . \square

Arrivati a questo punto, si vuole ora definire l'operazione di "*contrazione*", o "*shrinking*" come indicata dallo stesso Edmonds, che verrà poi utilizzata all'interno dell'omonimo algoritmo per risolvere il così detto "*problema dei cicli dispari*", e riuscire quindi a trovare lo stesso dei cammini aumentanti, se ovviamente presenti, all'interno del grafo.

In generale dato $H = (V(H), E(H))$ un sottografo di $G = (V(G), E(G))$, si verifica che l'insieme degli archi del grafo G è pari all'unione disgiunta

$E(G) = E(G \setminus H) \cup \delta(H) \cup E(H)$. Se H è un grafo connesso, allora "contrarre" H significa costruire il nuovo grafo G/H ottenuto a partire da G rimuovendo H e aggiungendo un nuovo vertice h ed un arco (u, h) per ogni $(u, v) \in E$ tale che $u \in V(G) \setminus V(H)$ e $v \in V(H)$. Si noti in particolare che $|\delta_G(H)| = |\delta_{G/H}(h)|$. Così facendo potrebbero originarsi, però, degli archi paralleli che, nel caso della ricerca del matching massimo all'interno di un dato grafo, possono essere tranquillamente identificati a formarne uno unico. Il nuovo vertice h risultante della contrazione G/H sarà chiamato "pseudoverteice".

Si consideri ora M un matching di G , dopo la contrazione si definisce $M/H = M \cap E(G/H)$. Si osservi che tale insieme non è necessariamente un matching di G . Chiaramente, però, se B è un blossom in G rispetto al matching M , allora anche M/B è un matching in G/B .

Viceversa, a partire dal grafo "contratto" G/H , si definisce l'operazione inversa di "espandere" lo pseudoverteice h quale ripristinare il grafo G di partenza.

Vale innanzitutto il seguente elementare risultato:

Teorema 3.5. *Dato B un qualsiasi ciclo dispari contenuto in G , per qualsiasi matching M_1 di G/B esiste un matching massimo M_B di B tale che $M = M_1 \cup M_B$ è un matching di G .*

Dimostrazione. Dato che qualsiasi matching M_1 di G/B contiene al massimo un arco che ha come estremità lo pseudoverteice b , risultante dalla contrazione di B , allora gli archi di M_1 in G toccano al massimo un solo vertice di B , chiamiamolo b' . Dunque si conclude subito che il matching M_B cercato è il matching massimo di B che lascia b' come unico vertice esposto. \square

Osservazione 7. Poiché la cardinalità del matching massimo M_B è costante, allora qualsiasi aumento del matching M_1 in G/B corrisponde anche ad un aumento complessivo del matching $M = M_1 \cup M_B$. Tale osservazione sarà di rilevante importanza nella sezione successiva, dove verrà esposto il processo algoritmico di Edmonds che consta di "contrazioni" ed "espansioni" di blossoms con eventuali "aumenti" del matching di partenza.

Riprendendo ora i concetti di alberi e fiori, è possibile enunciare il seguente teorema, la cui dimostrazione è analoga e segue direttamente da quella del lemma 3.4.

Teorema 3.6. *Dati il grafo G e il matching M , sia J_F un albero fiorito in (G, M) e B un blossom di J_F . Allora J_F/B è un albero alternante in $(G/B, M/B)$ e lo pseudoverteice b è un nodo esterno in J_F/B . Inoltre vertici interni ed esterni sono rispettivamente gli stessi di quelli di J_F che non stanno in B .*

Si conclude la presente sezione con il seguente fondamentale risultato, la cui dimostrazione è tratta da [5]:

Teorema 3.7. *Sia B il blossom di un fiore F contenuto nel grafo G rispetto al matching M . Allora M è un matching massimo in G se e solo se M/B è un matching massimo in G/B .*

Dimostrazione. (\implies) Sia M un matching massimo di G e si supponga per assurdo che M/B non sia un matching massimo in G/B . Si consideri allora N/B come un matching più grande di M/B in G/B . Ovviamente gli archi di N/B formano anche un matching per il grafo G , che sarà indicato con N_0 . Tale matching può coprire al massimo un nodo di B in G . Dato che B è un ciclo dispari allora il matching N_0 può essere esteso da $k = \frac{|V(B)|-1}{2}$ archi ad un matching N in G , per cui si ha che:

$$|N| = |N_0| + k = |N/B| + k > |M/B| + k = |M|$$

Contraddizione. Contro l'ipotesi che M sia un matching massimo in G .

(\impliedby) Viceversa, si suppone per assurdo che M non sia un matching massimo in G . Indicato con P il cammino M -alternante che forma lo stem del fiore $F = B \cup P$, si definisce $N = M \triangle P$ un nuovo matching di G . È immediato verificare che N e M hanno la stessa cardinalità. Infatti qualsiasi stem è in realtà un cammino alternante di lunghezza pari poiché o è formato da un singolo vertice, oppure alle sue estremità si trovano rispettivamente un nodo esposto ed uno coperto dal matching di riferimento. Pertanto si conclude che neanche N è un matching massimo in G . Inoltre N non copre il nodo di raccordo b . Per il *teorema di Berge 1.5* esiste allora un cammino N -aumentante, chiamiamolo Q , in G . Ovviamente almeno uno dei due estremi di Q , diciamolo v , non appartiene a B , poiché b è l'unico nodo N -esposto in B . Si contrae ora il blossom B ottenendo così il matching $N' = N/B$ in G/B e il cammino $Q' = Q/B$ i cui estremi sono ancora vertici N' -esposti. Quindi Q' è un cammino N' -aumentante in G/B . Dunque N' non è un matching massimo in G/B . *Contraddizione.* Poiché $|N'| = |M/B|$ allora neanche M/B risulterebbe essere un matching massimo per G/B , contro le ipotesi fatte. \square

3.3 L'algoritmo di Edmonds

Lo scopo di quest'ultima sezione è quello di analizzare in maniera precisa e dettagliata l'algoritmo di Jack Edmonds, impiegato per il calcolo del matching massimo su un grafo non orientato dato. Si è scelto di descrivere

dapprima il processo algoritmico ristretto al caso di un singolo albero alternante, come dato in [4], poiché una tale procedura può poi essere facilmente estesa anche al caso più generale di foreste alternanti, come presentato in [5]. Nella trattazione che segue si assumerà sempre di avere a disposizione un grafo non orientato $G = (V, E)$ ed un matching M in G .

Teorema 3.8. *Qualsiasi albero alternante $J(M)$ in G può essere esteso o ad un albero aumentante, o ad un albero fiorito, o ad un albero ungherese.*

Dimostrazione. Dato $J(M)$ un albero alternante in G , sia $D = \{(u, v) \in G \setminus J \mid u \text{ è un vertice interno e } v \text{ è un vertice esterno di } J\}$ (eventualmente $D = \emptyset$). Se ogni nodo esterno in J è estremo di archi in $E(J) \cup D$, allora J è un albero ungherese (1) per definizione, poiché ogni nodo esterno risulta essere adiacente solamente a nodi interni. Se invece esistono un nodo esterno v_1 di J ed un arco $e \notin E(J) \cup D$ incidente v_1 e un altro vertice v_2 , si possono quindi avere i seguenti casi:

- se anche v_2 è un vertice esterno di J , allora $e \cup J$ è un albero fiorito (2)
- se v_2 è un vertice M -esposto non appartenente a J , allora $e \cup J$ è un albero aumentante (3)
- se v_2 è un vertice M -coperto non appartenente a J , allora l'arco del matching (v_2, v_3) non sta in J e in particolare anche l'estremo $v_3 \notin J$ per definizione di albero alternante. In quest'ultimo caso è quindi possibile estendere J ad un albero alternante più grande con un nuovo vertice interno v_2 ed un nuovo vertice esterno v_3 e l'arco (v_2, v_3) del matching che li collega. Dato che gli archi di G sono in numero finito allora prima o poi si verificherà comunque uno dei precedenti tre casi, poiché tale procedura aumentante "esaurisce" gli archi di G .

□

Come diretta conseguenza del precedente teorema si ha allora che: a partire da un albero alternante $J(M)$ in G (eventualmente formato da un singolo vertice M -esposto) è sempre possibile estendere J ad uno dei tre casi *aumentante*, *fiorito* o *ungherese*, visitando al massimo una volta ogni arco di G . Questo primo fondamentale risultato mostra anche come l'algoritmo che si sta iniziando a delineare sia nel contempo "efficiente", poiché non richiede di tracciare molte combinazioni diverse degli stessi archi per trovare un cammino aumentante, o verificare che non ce ne siano. Infatti si arriva all'una o all'altra conclusione senza mai rivisitare gli archi già incontrati nel processo, tranne che per selezionare il fiore o il cammino aumentante eventualmente

trovato all'interno dell'albero.

Nel caso particolare in cui ci si dovesse imbattere in un fiore $F(M)$, si decide allora di proseguire "contraendo" il blossom $B(M)$ ed ottenendo così un nuovo albero alternante da cui far ripartire il processo di estensione (si veda *teorema 3.6*). Quest'operazione di "shrinking" e la sua inversa "espansione" sono di fatto già state formalmente definite nella sezione precedente. Mentre, da un punto di vista computazionale è possibile realizzare comunque tale contrazione in maniera abbastanza semplice: dato $B(M)$ un blossom in (G, M) basta essenzialmente etichettare tutti i nodi e gli archi di B con "b" e poi, finché lo pseudovertice b non sarà espanso cancellando le suddette etichette, tutti gli elementi di B verranno considerati come un singolo vertice b ignorando qualsiasi distinzione tra di essi. Così facendo si riesce a tener traccia del blossom B all'interno dell'algoritmo che continuerà poi ad operare sul grafo ridotto G/B . Ovviamente è possibile generalizzare tale struttura anche al caso di più contrazioni successive.

In sostanza si è mostrato come a partire da un qualsiasi albero alternante J in (G, M) ed eventualmente contraendo i blossom B_i incontrati nel processo, è sempre possibile ottenere in un grafo ridotto $G_k = G/B_1/\dots/B_k$ o un albero aumentante oppure un albero ungherese.

Nel primo caso si è quindi in presenza di un cammino aumentante e pertanto è effettivamente possibile aumentare il matching $M_k = E(G_k) \cap M$ mediante differenza simmetrica (si veda *teorema di Berge 1.5*). Grazie poi ai *teoremi 3.5* e *3.7* ed alla definizione stessa di blossom (*definizione 3.8*) è possibile indurre un tale aumento anche per il matching $M_{k-1} = E(G_{k-1}) \cap M$ e così via espandendo di volta in volta gli pseudovertici b_i fino al matching iniziale M in G .

Nel secondo caso invece, se è possibile ottenere un albero ungherese J si verifica allora che il "sottomatching" $(J \cup B_1 \cup \dots \cup B_k) \cap M$ di (G, M) non può più essere migliorato e pertanto questa parte di G non verrà più considerata all'interno dell'algoritmo. Tale fatto si deve in realtà al seguente teorema, in cui per una più ampia generalizzazione il grafo G_k viene denotato semplicemente con G .

Teorema 3.9. *Sia J un albero ungherese in un dato grafo $G = (V, E)$. Un matching M_1 di $G \setminus J$ è massimo in $G \setminus J$ se e solo se M_1 unito ad un qualsiasi matching massimo M_J di J è un matching massimo di G .*

Dimostrazione. (\Leftarrow) Una prima implicazione è banale. Dato infatti che J e $G \setminus J$ sono disgiunti, se per assurdo esistesse un matching M'_1 di $G \setminus J$ tale che $|M'_1| > |M_1|$, allora $M'_1 \cup M_J$ sarebbe un matching di G di cardinalità

maggiore rispetto a quella di $M_1 \cup M_J$, infatti $|M'_1 \cup M_J| = |M'_1| + |M_J| > |M_1| + |M_J| = |M_1 \cup M_J|$. *Contraddizione*. Contro l'ipotesi che $M_1 \cup M_J$ sia massimo in G .

(\implies) Viceversa, siano M_1 un matching massimo di $G \setminus J$ e M_J un matching massimo di J . Un qualsiasi matching M' di G può essere suddiviso in tre "sottomatchings" disgiunti $M' = M'_{G \setminus J} \cup M'_J \cup M'_C$ rispettivamente contenuti in $G \setminus J$, J e $C = \{(u, v) \in E \mid u, v \in V(J), \text{ ma } (u, v) \notin E(J)\}$. Sicuramente $|M'_{G \setminus J}| \leq |M_1|$, poiché M_1 è massimo in $G \setminus J$. Si osserva invece che per ogni arco di M'_C almeno uno dei due estremi è un vertice interno di J , poiché J è ungherese. Sia quindi $I' \subseteq VI$ l'insieme dei nodi interni che sono estremità di un arco di M'_C , senz'altro vale quindi che $|M'_C| \leq |I'|$. Si ricordi ora che se a partire da un albero alternante J si decide di sottrarre un nodo interno u , allora il nuovo grafo ottenuto $J \setminus \{u\}$ risulta essere formato da due alberi alternanti disgiunti. Si sceglie quindi di procedere rimuovendo tutti i vertici interni I' dell'albero ungherese J , ottenendo così $|I'| + 1$ alberi alternanti disgiunti con un totale di $|VI - I'|$ vertici interni. Dato che (come è già stato osservato nella *sezione 3.1*) un matching massimo in un albero alternante ha cardinalità pari al numero dei suoi vertici interni, si ha quindi che $|M'_J| \leq |VI - I'|$ e anche $|M_J| = |VI|$. In conclusione sommando le disuguaglianze si ottiene allora che: $|M'| = |M'_{G \setminus J}| + |M'_J| + |M'_C| \leq |M_1| + |VI - I'| + |I'| = |M_1| + |VI| = |M_1| + |M_J| = |M_1 \cup M_J|$, da cui segue la conclusione. \square

Quest'ultimo teorema mostra di fatto che se alla fine del processo algoritmico non è stato possibile trovare un albero aumentante, e quindi un cammino M -aumentante, allora il matching M di partenza era massimo in J , albero ora ungherese, per cui è poi possibile considerare separatamente la parte rimanente $G \setminus J$.

Riassumendo, a partire da un qualsiasi matching M del grafo $G = (V, E)$, eventualmente anche $M = \emptyset$, se $|M| = \frac{|V|}{2}$ allora M è perfetto e quindi massimo in G . Altrimenti sia $S = \{v \in V(G) \mid v \text{ è } M\text{-esposto}\}$, allora l'insieme S forma di per sè una foresta alternante dove ogni albero alternante corrisponde ad un singolo vertice M -esposto. "Applicando" quindi il *teorema 3.8* a qualsiasi albero della foresta è possibile ottenere un matching massimo di G . Informalmente l'ALGORITMO DEL MATCHING DI CARDINALITÀ MASSIMA DI EDMONDS può essere facilmente generalizzato al caso di foreste alternanti come segue: sia v un vertice esterno di un qualsiasi albero alternante J della foresta e sia P_v l'unico cammino che collega v alla radice r della stessa componente connessa. Successivamente si consideri un vertice u adiacente a v . Esistono quindi tre casi interessanti, che corrispondono alle tre operazioni "grow", "augment" e "shrink":

- **CASO 1:** u è un vertice M -coperto che non appartiene a J , ed a nessun altro albero alternante in G . Allora l'albero J cresce, e quindi anche tutta la foresta, quando si aggiungono (v, u) e l'arco del matching che copre u .
- **CASO 2:** u è un vertice esterno di una diversa componente connessa della foresta. Allora il cammino $P_v \cup (u, v) \cup P_u$ risulta essere M -aumentante in G , e pertanto è possibile aumentare il matching M .
- **CASO 3:** u è un vertice esterno di J . Allora è possibile estendere J ad un albero fiorito e si procede quindi con la contrazione del blossom $B(M)$ in esso contenuto. Sia infatti b il primo vertice di P_v (che inizia in v) e che appartiene anche a P_u (b può eventualmente essere uno tra i vertici v e u). Come dimostrato nel *lemma 3.4* si ha che il sottografo $B = P_{v,b} \cup (v, u) \cup P_{u,b}$ forma un blossom con almeno tre vertici in J .

Se non è possibile applicare nessuno dei tre casi precedenti, questo significa che tutti i vicini dei vertici esterni di J sono interni per J o per altre componenti connesse della foresta. Se questa condizione è verificata per ogni albero della foresta, di fatto si è ottenuta una foresta ungherese. È quindi facilmente possibile estendere il *teorema 3.9* per dimostrare che il matching ottenuto è massimo in G e pertanto l'algoritmo termina.

UN ESEMPIO

In conclusione alla presente trattazione si è deciso di mostrare un esempio chiarificatore, in cui si è cercato di riassumere tutte le possibili occorrenze dell'algoritmo per il calcolo del matching massimo di Edmonds.

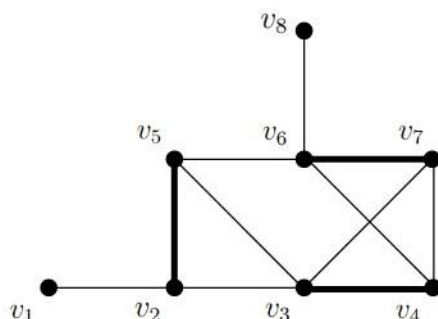
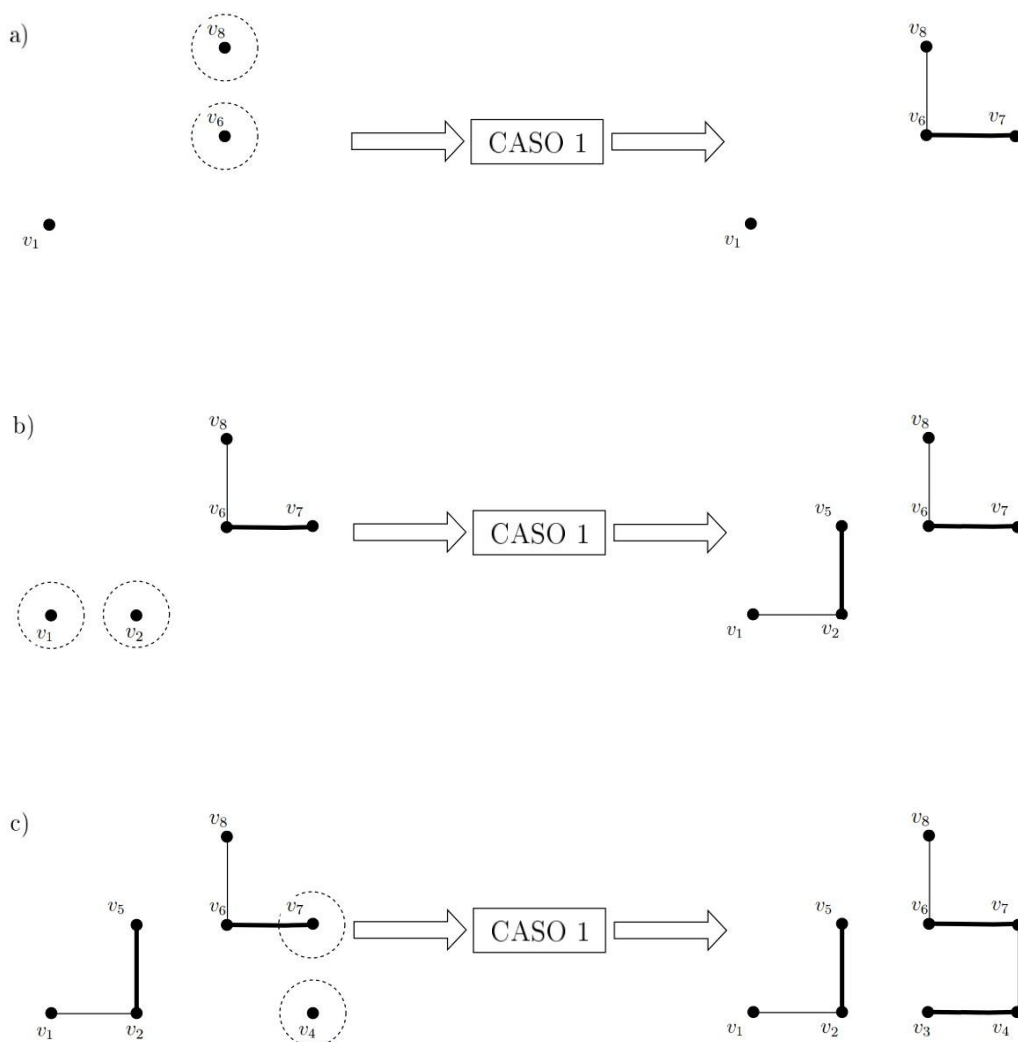


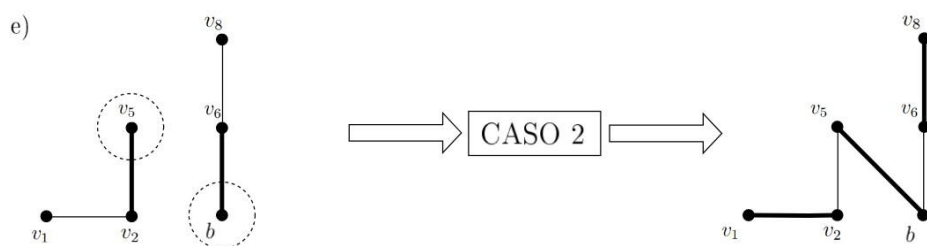
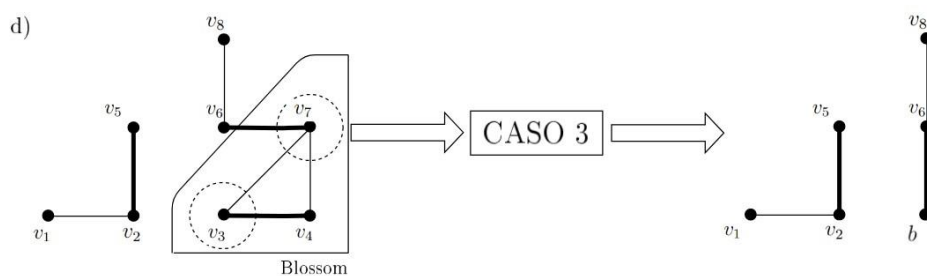
Figura 3.2

44CAPITOLO 3. ALGORITMO DEL MATCHING MASSIMO DI EDMONDS

Input: siano dati un grafo G e un matching di partenza M di G come in figura 3.2, dove gli archi di M sono stati evidenziati mediante linee in grassetto.

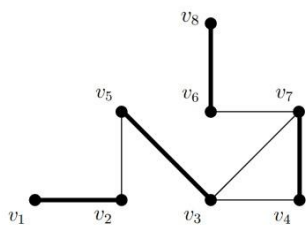
Svolgimento: dato che $|M| = 3 < 4 = \frac{|V(G)|}{2}$, allora M non è perfetto in G e pertanto ci si chiede se è possibile aumentare tale matching. Sia allora $S = \{v \in V(G) \mid v \text{ è } M\text{-esposto}\} = \{v_1, v_8\}$ la foresta alternante da cui cominciare. Ad ogni passo dell'algoritmo i due vertici presi in considerazione saranno evidenziati con un cerchio tratteggiato. Inoltre la modalità con cui procedere sarà indicata come caso 1, 2 o 3 rispettivamente con lo stesso significato di quanto illustrato nel paragrafo precedente.





STOP

Decontrazione



Output: il matching $M = \{(v_1, v_2), (v_3, v_5), (v_4, v_7), (v_6, v_8)\}$ trovato è perfetto il G e quindi a maggior ragione è anche massimo, come si voleva.

Bibliografia

- [1] Manuel F. Aprile. Matchings - note supplementari. March 8, 2022.
- [2] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.
- [3] Michele Conforti and Yuri Faenza. Note del corso di matematica discreta. A.A. 2011/12.
- [4] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.
- [5] Bernhard Korte and Jens Vygen. *Ottimizzazione Combinatoria*. Springer, 2011.
- [6] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [7] Paolo Serafini. *Ricerca operativa*. Springer Science & Business Media, 2009.
- [8] U Zwick. Maximum matching in bipartite and non-bipartite graphs, 2009.