IL PROBLEMA DEL MATRIMONIO STABILE

Beatrice Bernardini

INTRODUZIONE

- Il problema del matrimonio stabile è uno tra i più importanti problemi di matching.
- Possiamo considerare i problemi di matching come particolari problemi di ottimizzazione che richiedono la suddivisione in coppie distinte di elementi, legati da una relazione, rispettando dei vincoli ben determinati.
- I problemi di matching possono essere suddivisi in varie classi: problema del matching massimo, del matching di peso massimo/ minimo e problemi di matching stabili.
- L'origine di questo nome deriva dal fatto che nella sua formulazione più nota si richiedeva di definire una serie di matrimoni tra un gruppo di ragazzi e uno di ragazze in modo tale che queste unioni fossero **stabili**, ovvero al riparo da possibili tradimenti che possano mettere in crisi i matrimoni stessi.
- La formulazione di questo problema si presta molto bene ad essere adattata nell'ambito della teoria dei grafi, noi ci occuperemo principalmente di matching in grafi bipartiti poiché si modellano meglio alle nostre necessità.

Una possibile formulazione del problema

Consideriamo un gruppo di 24 ragazzi composto da 12 maschi e 12 femmine.
Ad ogni ragazzo viene chiesto di esprimere una preferenza solamente per 4 ragazze.

Quello che ci domandiamo è: *«È possibile associare ad ogni ragazzo una ragazza in modo tale che ad ognuno vada una delle sue predilette?»*

È chiaro che non sempre questo è possibile, ad esempio non lo è se una delle ragazze non è nelle preferenze di alcun ragazzo. Un'altra domanda che ci possiamo porre è: *«È* possibile fare tutto ciò senza tener conto in alcun modo delle preferenze delle ragazze?».

Questo problema si può tradurre in modo molto semplice in un problema di teoria dei grafi creando un **grafo bipartito** nel seguente modo:

- identificando i ragazzi con i **vertici del grafo** suddivisi in due insiemi di vertici disgiunti, quello dei ragazzi e quello delle ragazze;
- indentificando **i lati del grafo** come le preferenze di ciascun ragazzo.

Nel nostro particolare esempio si ha che ogni vertice dell’insieme dei ragazzi sarà adiacente a quattro vertici dell’insieme delle ragazze.

Il **problema originale** quindi si riduce nel trovare 12 lati che accoppiano tutti e 24 i vertici del grafo.

Abbiamo quindi osservato che un’ovvia **condizione necessaria** al fatto che ad ogni ragazza sia assegnato un diverso ragazzo è che ognuna sia tra le favorite di almeno uno dei ragazzi. Quindi estendendo questa osservazione, si ha che una condizione necessaria affinché possa essere fatta una assegnazione completa è che per ogni sottoinsieme S di ragazze un numero almeno uguale di ragazzi indichi una ragazza nell’insieme S tra le proprie favorite.

Vedremo con il teorema di Hall che **questa ovvia condizione necessaria risulterà anche sufficiente** affinché si possa stabilire un matching completo.

Esempio di matching in un grafo non bipartito:

Vedremo ora un problema di matching che può essere formulato utilizzando un **grafo non bipartito**.

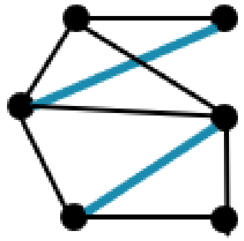
Consideriamo una compagnia aerea che dispone di una flotta di aeromobili e di un insieme di piloti. Per ogni aeromobile occorre decidere la coppia di piloti che lo possa guidare, in modo che i piloti di ciascun aereo siano in grado, per carattere, intelligenza, ecc., di collaborare in maniera efficace tra di loro.
In questo caso il problema può essere modellizzato con un grafo in cui i vertici corrispondono ai piloti, mentre uno spigolo tra due vertici esiste se e solo se i due piloti possono costituire un equipaggio.

DEFINIZIONI

Dato un grafo $G = (V,E)$,

- Un sottoinsieme $M \subseteq E$ è un **matching** in un grafo G se è un insieme di lati a due a due non incidenti, ovvero non aventi estremi in comune. Si ha che se un lato $(i,j) \in M$ si dice **accoppiato** mentre un lato $(i,j) \notin M$ si dice **libero**. Un vertice su cui incide un lato di M si dice accoppiato mentre un vertice su cui non incide nessun lato di M si dice **esposto**.

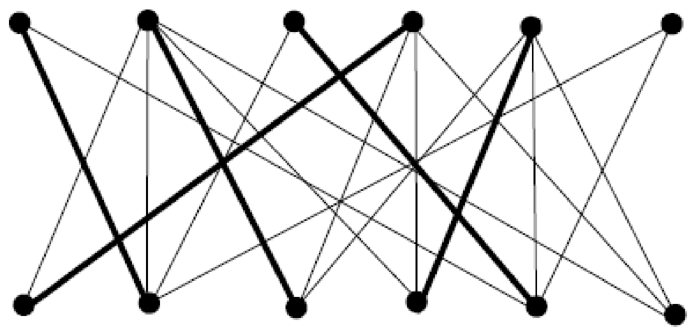
G



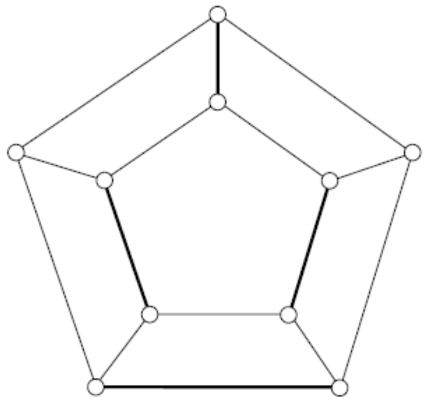
M

Un matching M di G

- Un matching M di G è **massimo** se ha cardinalità massima, ovvero un è matching tale che per ogni altro matching M' di G si ha: $|M| \geq |M'|$
- Un matching M di G è **massimale** se non esiste nessun matching di G che lo contiene propriamente.

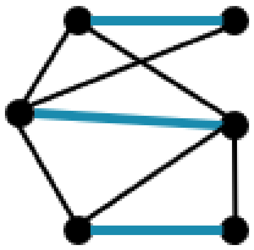


Un matching massimo in un grafo bipartito

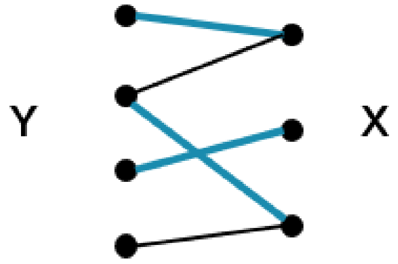


Un matching massimale

- Un grafo bipartito G = (X∪Y, E) ammette un **matching perfetto** se esiste un matching M ⊆E tale che ogni vertice v ∈ X∪Y è estremo di un lato del matching, ovvero tutti i vertici del grafo sono accoppiati.
  Naturalmente un grafo bipartito può ammettere un matching perfetto se e solo se $|X| = |Y|$.
  Nei casi in cui $|X| \neq |Y|$ allora parliamo di matching **X-completi** oppure matching **Y-completi.**
  Si ha quindi che G ammette un matching X-completo se esiste un matching M ⊆E tale che $|M| = |X|$



Un matching perfetto



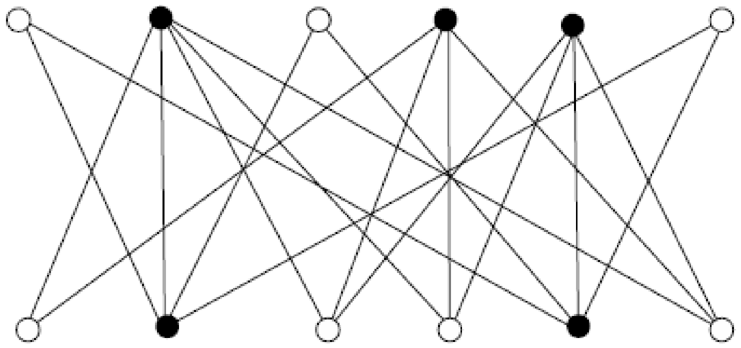
Un matching X-completo

TEOREMA DI HALL

Per dimostrare questo teorema passeremo attraverso il teorema di König, che fornisce una formulazione equivalente al teorema di Hall. Dobbiamo prima però introdurre una nuova nozione:

Definizione: Dato un grafo non orientato G = (V,E) una **copertura** (o vertex cover) è un sottoinsieme di vertici $U \subset V$ tale che, per ogni lato $(i,j) \in E$, $i \in U$ oppure $j \in U$ (oppure entrambi), vale a dire che ogni lato è «coperto» da almeno un vertice di U.

Una copertura U è detta **minimale** se $U \setminus \{i\}$ non è più una copertura per ogni i appartenente a U.

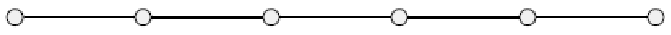


copertura minima in un grafo bipartito

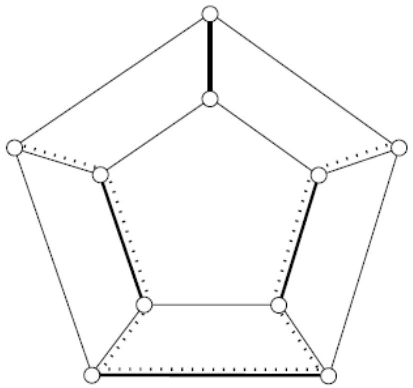
Teorema di König:
In un grafo bipartito G = ((A,B), E) la cardinalità del massimo matching in G è uguale alla cardinalità del minimo vertex cover: $\max|M| = \min|U|$.

Definizioni: Dato un grafo G = (V, E) ed un suo matching M, un cammino $v_1, e_1, \dots, e_{k-1}, v_k$ su G si dice **M-alternante** se, per $1 \le i \le k-2$, e_i appartiene al matching se e solo se e_{i+1} non appartiene al matching, cioè i suoi lati appartengono alternativamente ad M ed a E\M.

Un cammino M-alternante si dice **M-aumentante** se i suoi vertici all'estremità sono esposti, equivalentemente se il primo e l'ultimo lato del cammino solo lati liberi.



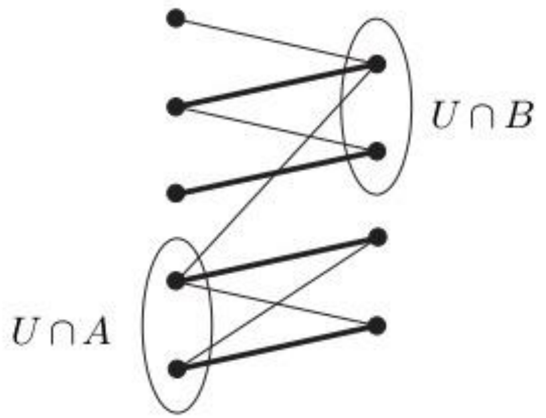
Un cammino M-alternante



Un cammino M-aumentante

Dimostrazione: Sia M un matching massimo di G. Per ogni lato in M scegliamo una delle sue estremità: la sua estremità in B se qualche cammino alternante finisce in quel vertice, la sua estremità in A altrimenti.

Ora dobbiamo far vedere che l'insieme U di questi |M| vertici copre il grafo G. Poiché ogni vertex cover di G deve coprire M, non ci può essere nessuno con meno di |M| vertici, e da questo seguirà il teorema.



Il vertex cover U

Sia $ab \in E$ un lato, facciamo vedere che o a o b stanno in U. Se $ab \in M$ questo è vero per definizione di U. Supponiamo quindi che $ab \notin M$. Siccome M è un matching massimale, esso contiene un lato $a'b'$ con $a= a'$ oppure $b=b'$. Infatti, possiamo assumere che $a = a'$: perché se cosi non fosse e quindi a è non accoppiato (e $b= b'$), allora ab è un cammino alternante, e quindi l'estremità di $a'b' \in M$ scelto per U era il vertice $b' =b$. Ora se $a' = a$ non è in U, allora $b' \in U$ e qualche cammino alternante P finisce in b'. Ma allora c'è anche un cammino alternante P' che termina in b: o $P':=Pb$ (se $b \in P$) oppure $P' := P\, b'a'b$. Ma dalla massimalità di M sappiamo che comunque P' non è un cammino aumentante. Quindi b deve essere abbinato, ed è stato scelto per U dal lato di M che lo contiene.

TEOREMA DI HALL

Definizione:

Sia G= (V,E) un grafo e sia Q ⊆ V un sottoinsieme di vertici del grafo.
Chiamiamo i vicini di Q l'insieme di tutti i vertici *u* che non appartengono a Q tali che esista un lato (q,u) per qualche q ∈ Q.

$$N(Q) = \{u \in V \backslash \mathrm{Q} \mid \exists (q, u) \in E \ per \ qualunque \ q \in Q\}.$$

In altre parole, i vicini di un certo sottoinsieme di vertici di Q è l'insieme dei vertici adiacenti ai vertici di Q.

Enunciato:

Un grafo bipartito G=((V_1, V_2), E), con $|V_1| \leq |V_2|$ ammette un matching V_1- *perfetto* se e solo se vale la seguente condizione, che chiameremo condizione di Hall:

$$\forall Q \subseteq V_1, \ |Q| \leq |N(Q)|$$

Dimostrazione:

$\Longrightarrow$ Supponiamo di avere un matching M che sia V_1- *perfetto.* Sia *n* una funzione che associa ad ogni vertice di V_1 il vertice a cui è associato secondo il matching M.

$$n: V_1 \rightarrow V_2 \ \textit{tale che}\ n(x) = y \ \textit{se e solo se}\ (x, y) \in M \ \ \forall x \in V_1 \ \forall y \in V_2$$

Consideriamo un qualsiasi sottoinsieme Q di vertici di V_1 e definiamo $\widetilde{N}$ l'insieme dei vertici di V_2 che sono accoppiati a qualche vertice di Q.
Per definizione di matching si ha che $|\widetilde{N}| = |Q|$. Inoltre sicuramente è vero che $\widetilde{N} \subseteq N(Q)$ perché un vertice di V_2 non può essere accoppiato ad un vertice di Q se non appartiene al suo insieme di vicini. Da questo deriva che:

$$\forall\, Q \subseteq V_1 \ |N(Q)| \geq |\widetilde{N}| = |Q| \quad \longrightarrow |N(Q)| \geq |Q|.$$

$\Longleftarrow$ Facciamo ora vedere che se vale la condizione di Hall allora il grafo bipartito G ammette un matching V_1- *perfetto.*
Supponiamo $\forall Q \subseteq V_1 \ |Q| \leq |N(Q)|$.
Sia C^* un vertex cover minimo di G. Consideriamo $Q = V_1 \setminus C^*$.

Per ogni lato in E si ha che almeno uno dei due estremi è in C^*, quindi $N(Q) \subseteq C^* \cap V_2$, allora si ha che: $|C^* \cap V_2| \geq |N(Q)| \geq |Q| = |V_1 \setminus C^*| = |V_1| - |C^* \cap V_1|$.

Sommando ad ambo i membri $|C^* \cap V_1|$ si ottiene:

$$|C^* \cap V_1| + |C^* \cap V_2| \geq |V_1|.$$

Dobbiamo però notare che $|C^* \cap V_1| + |C^* \cap V_2| = |C^*|$, otteniamo che:

$|C^*| = |M^*|$ per il teorema di König e quindi $|M^*| = |C^*| \geq |V_1|$, si può concludere che esiste un matching V_1- *perfetto.*

■

Osservazione: Il teorema di Hall ha applicazioni in varie parti della matematica e, a seconda del contesto, può assumere diverse formulazioni, a volte apparentemente lontane dalla teoria dei grafi.

Vediamo ora una formulazione equivalente del teorema di Hall molto usata in combinatoria.

Definizione: Sia $\mathcal{F}$ = $(\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_\mathrm{n})$ una famiglia di sottoinsiemi (non necessariamente distinti) di un insieme X. Un **sistema di rappresentanti distinti** (SDR) di $\mathcal{F}$ è una *n*-upla $(a_1, a_2, \ldots, a_n)$ di elementi distinti di X tale che $a_i \in A_i$ per ogni *i*= 1, …, *n*

Formulazione equivalente del teorema di Hall:

Una famiglia $\mathcal{F}$ = $(\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_\mathrm{n})$ di sottoinsiemi dell'insieme X ammette un sistema di rappresentanti distinti se e soltanto se

$$\left|\bigcup_{i\in S} A_i\right| \geq |S| \text{ per ogni } \mathrm{S} \subseteq \{1,2,\ldots,n\}.$$

Esempio:

Consideriamo $\mathcal{F} = (A_1, A_2, A_3, A_4, A_5)$ con

- A_1 = {a,b,c}
- A_2 = {b}
- A_3 = {a,c,d,e,f}
- A_4 = {c,d,e}
- A_5 = {c,d,e,f}

$\longrightarrow$ abbiamo che {a,b,c,d,e} è un SDR, ma non è l'unico, ad esempio anche {c,b,d,e,f} è un sistema di rappresentanti distinti.

Dimostrazione:

$\Longrightarrow$ Se $(a_1, a_2, \dots, a_n)$ è un sistema di rappresentanti distinti della famiglia di insiemi, allora per ogni $\emptyset \neq$ S $\subseteq \{1,2, \dots, n\}$ si ha $\bigcup_{i \in S} A_i \supseteq \{a_i$ tale che $i \in$ S}, e poiché gli a_i sono a due a due distinti allora la condizione di Hall è soddisfatta.

$\Longleftarrow$ Consideriamo il grafo bipartito G sui vertici V_1 = X e V_2 = {1,2, …, n}, possiamo supporre V_1 e V_2 disgiunti. Inoltre abbiamo che due vertici $x \in X$ e $i \in \{1, \dots, n\}$ sono adiacenti se e solo se $x \in A_i$. Si ha che per ogni S$\subseteq \{1, \dots, n\}$, $|\bigcup_{i \in S} A_i| = |N_G(\text{S})|$, e quindi per la condizione di Hall, vista nel teorema precedente, si ha che esiste un matching V_1-*perfetto.*
I vertici in X dei lati che formano tale matching costituiscono un sistema di rappresentanti distinti per la famiglia $A_1, A_2, \dots, A_n$.

TEOREMA DI TUTTE

Vogliamo ora far vedere una generalizzazione del teorema di Hall che ci caratterizza l'esistenza di un matching perfetto in un grafo arbitrario, fornendoci una condizione necessaria e sufficiente affinché un grafo qualsiasi abbia un matching perfetto.

Notazione: Dato un grafo G = (V,E) indichiamo con o(G) il numero di componenti connesse dispari di G.

Teorema: Un grafo G = (V,E) ha un matching perfetto se e solo se per ogni sottoinsieme $U \subseteq V$, il sottografo indotto da $V \setminus U$ ha al massimo $|U|$ componenti connesse con un numero dispari di vertici.

Dimostrazione: Vogliamo quindi far vedere che: $\forall\, U \subseteq V,\ o(G - U) \leq |U|$.

TEOREMA DI TUTTE

Dimostriamo la **condizione necessaria**:

Si consideri un grafo G, con un matching perfetto. Sia U un sottoinsieme arbitrario di V. Eliminiamo U da G e consideriamo C una componente dispari arbitraria in G-U. Poiché G aveva un matching perfetto, almeno un vertice in C deve essere accoppiato ad un vertice in U. Quindi ciascuna componente dispari ha almeno un vertice accoppiato con un vertice in U. Poiché ciascun vertice in U può essere in questa relazione con al massimo una componente connessa, perché essa viene accoppiata al massimo una sola volta in un matching perfetto, allora vale $o(G - U) \leq |U|$.

ALGORTIMO DI GALE E SHAPLEY

- Questo algoritmo permette di **trovare sempre una soluzione stabile** al problema del matrimonio, se il problema a cui viene applicato è formulato con liste di preferenza complete e insiemi di uguale cardinalità.
- Osserviamo che lo scopo dell'algoritmo non è creare delle coppie che corrispondono necessariamente all'ottimo per i due partner, ma una soluzione stabile.
- **L'idea di questo algoritmo** è quella di creare una situazione di asimmetria tra i due insiemi, sono infatti gli elementi di uno dei due insiemi a poter scegliere seguendo la propria configurazione ottima, secondo l'ordine stabilito dalla propria lista di preferenza, per poi andarsi a stabilizzare sulla configurazione ottimale poiché gli elementi dell'altro insieme cercano ad ogni iterazione di migliorare, se possibile, la propria situazione decidendo di volta in volta se accettare il cambio.

Consideriamo i seguenti insiemi X ed Y composti entrambi da n elementi:

$$X = \{ x_1, \dots, x_n \} \text{ e } Y = \{ y_1, \dots, y_n \}$$

Definizione: Un *matching* tra X ed Y è un sottoinsieme del prodotto cartesiano $X \times Y$, costituito da una famiglia di *n* coppie tale che ciascun elemento di X e ciascun elemento di Y appartengono *ad una ed una sola* coppia.

Abbiamo quindi che un matching M è:
$M = \{(x,y) : x \in X, \; y \in Y \}$ e per ogni $x \in X$ esiste un unico $y \in Y$ tali che $(x,y) \in M$ e viceversa.

Esempio:

Se $X = \{a,b,c\}$ e $Y = \{\alpha,\beta,\gamma\}$, un matching tra X e Y è $M = \{(a,\alpha), (b,\beta), (c,\gamma)\}$.

Dobbiamo quindi creare delle *liste di preferenza* ovvero ciascun elemento dell'insieme X deve esprimere una propria classifica sugli elementi di Y disponendoli in ordine decrescente di gradimento. In modo analogo faranno gli elementi di Y esprimendo le proprie preferenze sugli elementi di X.

Associamo quindi ad ogni elemento x_k ∈ X una *n*-pla ordinata costituita da una permutazione di tutti gli elementi di Y: P$(x_k) = (\, y_{k_1}, y_{k_2}, \ldots, y_{k_n})$ dove con $(k_1\,, \ldots, k_n)$ abbiamo indicato una permutazione degli indici (1,...,n).

Si ha quindi che la permutazione **P(x_k) rappresenta la lista di preferenza di x_k sugli elementi di Y.**

Analogamente, associamo a ciascun elemento y_h ∈ Y una n-pla ordinata costituita da una permutazione di tutti gli elementi di X: P$(y_h) = (\, x_{h_1}, x_{h_2}, \ldots, x_{h_n})$.

Notazione:

Sia $a \in X$ e α,β∈ Y abbiamo che se a preferisce α a β, scriveremo **α $<_a$β**, indicando così con il simbolo $<_a$ la relazione d'ordine totale definita da a sugli elementi di Y.

Una volta introdotte le «liste di preferenza» abbiamo che tra le coppie di un matching può venirsi a creare una **condizione di instabilità**.

Sia M un matching tra X e Y e siano (a, α), (b, β) ∈M due coppie del matching tale che β $<_a$ α e a $<_\beta$ b. Allora a e β avrebbero l'interesse reciproco di lasciare i propri compagni formando la nuova coppia (a,β) in cui entrambi migliorerebbero la propria condizione.

Il problema del matrimonio stabile ha l'obiettivo di definire un matching in cui la condizione di instabilità non si verifichi per nessuna coppia di elementi.

Da questo deriva la nozione di *stabilità*:

Definizione:

Un matching si dice **stabile** se la condizione di instabilità non si verifica tra le coppie del matching.

Osservazione:

È opportuno osservare che la preferenza dei partner di a e β è del tutto irrilevante ai fini della stabilità: b potrebbe preferire β ad ogni altro elemento di Y e analogamente α potrebbe preferire a ad ogni altro elemento di X ma questo non rende comunque il matching stabile.

Esempio:

Consideriamo i seguenti insiemi: $X = \{A,B,C\}$ e $Y = \{\alpha,\beta,\gamma\}$

Associamo ad ogni elemento di X ed Y la sua lista di preferenza:

$P(A) = (\beta, \alpha, \gamma)$

$P(B) = (\gamma, \alpha, \beta)$

$P(C) = (\alpha, \beta, \gamma)$

$P(\alpha) = (A,C,B)$

$P(\beta) = (C,A,B)$

$P(\gamma) = (A,B,C)$

Un matching possibile tra X e Y è $M = \{(A,\alpha), (B,\beta), (C,\gamma)\}$, ma questo *matching non è stabile*

Infatti la coppia (A, β) crea instabilità poiché A preferisce β rispetto al proprio partner α e anche β preferisce A rispetto al proprio partner B.

Un *matching stabile* è: $M = \{(A, \beta), (B, \gamma), (C, \alpha)\}$.

StableMarriage(X,Y)

Input: I due insiemi X e Y e le liste di preferenza complete

Output: Un matching stabile tra X e Y

```
k := 0, associa a $\Omega$ ogni y $\in$ Y
*fintanto che* k < n *ripeti*:
    sia $\bar{x} \coloneqq x_{k+1}$
    *fintanto che* $\bar{x} \neq \Omega$ *ripeti*:
        sia y la migliore scelta rimanente per $\bar{x}$ nella sua lista
        *se* y preferisce $\bar{x}$ al suo partner *allora*:
            associa ($\bar{x}$ ,y), sia $\bar{x}$ il precedente elemento associato ad y
        *fine-condizione*
        *se* $\bar{x} \neq \Omega$ *allora*:
            elimina y dalla lista di preferenze di $\bar{x}$
        *fine-condizione*
    *fine-ciclo*
k:= k+1
*fine-ciclo*
```

Teorema:

L'algoritmo StableMarriage **produce un matching stabile** per ogni istanza del problema formulata con $|X| = |Y|$ e le **liste di preferenza complete** per ciascun elemento di X e di Y.

Dimostrazione:

Analizziamo l'algoritmo nel dettaglio e vediamo perché riesce sempre a produrre un matching stabile:

- Il *passo 10* mi garantisce che se y viene rimosso dalla lista delle preferenze di $\bar{x}$ allora non esiste un matching stabile che contiene ($\bar{x}$,y). Infatti se y viene rimosso, vuol dire che ha preferito un altro elemento $\tilde{x} \in X$ *a* $\bar{x}$: dunque y preferisce $\tilde{x}$ *e* $\tilde{x}$ preferisce y ad ogni altro elemento presente nella propria lista, dato che le coppie del matching vengono costruite in base all'ordine di preferenza degli elementi di X.

StableMarriage(X,Y)

1: k := 0, associa a Ω ogni y $\in$ Y
2: *fintanto che* k < n *ripeti*:
3:     sia $\bar{x} := x_{k+1}$
4:     *fintanto che* $\bar{x} \neq \Omega$ *ripeti*:
5:         sia y la migliore scelta rimanente per $\bar{x}$ nella sua lista
6:         *se* y preferisce $\bar{x}$ al suo partner *allora*:
7:             associa ($\bar{x}$,y), sia $\bar{x}$ il precedente elemento associato ad y
8:         *fine-condizione*
9:         *se* $\bar{x} \neq \Omega$ *allora*:
10:             elimina y dalla lista di preferenze di $\bar{x}$
11:         *fine-condizione*
12:     *fine-ciclo*
13: k:= k+1
14: *fine-ciclo*

- Possiamo notare che la situazione di accoppiamento di ciascun elemento di Y non peggiora mai durante l'esecuzione dell'algoritmo. Infatti inizialmente ogni elemento di y si trova accoppiato con Ω l'elemento peggiore in assoluto e successivamente per la **condizione 6** l'accoppiamento di y cambia solo se la nuova proposta migliora la sua situazione attuale.
- La lista di preferenza di ogni elemento $x \in X$ **non diventa mai vuota**, infatti se così fosse vorrebbe dire che x è stato rifiutato da ogni $y \in Y$, il che non è possibile, dato che $|Y|=|X|$.
- Non è mai possibile che due elementi di Y vengano associati allo stesso elemento di X poiché ogni volta che rompo una coppia, vado a trovare un sostituto per l'elemento scoppiato.
- Possiamo quindi concludere **che l'algoritmo del matrimonio stabile è ben definito e termina sempre** individuando un matching stabile.

StableMarriage(X,Y)

1: k := 0, associa a Ω ogni $y \in Y$
2: *fintanto che* k < n *ripeti*:
3:     sia $\bar{x} \coloneqq x_{k+1}$
4:     *fintanto che* $\bar{x} \neq \Omega$ *ripeti*:
5:         sia y la migliore scelta rimanente per $\bar{x}$ nella sua lista
6:         *se* y preferisce $\bar{x}$ al suo partner *allora*:
7:             associa ($\bar{x}$,y), sia $\bar{x}$ il precedente elemento associato ad y
8:         *fine-condizione*
9:         *se* $\bar{x} \neq \Omega$ *allora*:
10:             elimina y dalla lista di preferenze di $\bar{x}$
11:         *fine-condizione*
12:     *fine-ciclo*
13: k:= k+1
14: *fine-ciclo*

Osservazione:
Dobbiamo osservare che l'ottimalità per l'insieme X, l'insieme dei ragazzi, avviene a scapito delle preferenze dell'insieme delle ragazze, infatti ciascun elemento dell'insieme Y riceve il peggior partner valido, infatti possiamo affermare che l'algoritmo conduce alla peggiore configurazione tra quelle accettabili (stabili) per le ragazze.

Proposizione:
L'algoritmo di Gale-Shapley trova l'abbinamento stabile M^* **pessimo per l'insieme Y**, l'insieme formato dalle ragazze.

Dimostrazione:
Supponiamo per assurdo che x e y sono abbinati in M^* in modo tale che *x* è diverso dal peggior partner valido per *y*. Allora esiste un abbinamento stabile M in cui *y* è abbinato ad un elemento di *X*, sia esso *x'* che *y* preferisce meno di *x.* Allora abbiamo che: ***y* preferisce *x* ad x'.**

Sia *y'* il partner di *x* in M, dall'ottimalità per gli elementi dell'insieme X, sappiamo che *y* è il miglior partner valido per *x*. Allora abbiamo che **x preferisce y ad y'.**

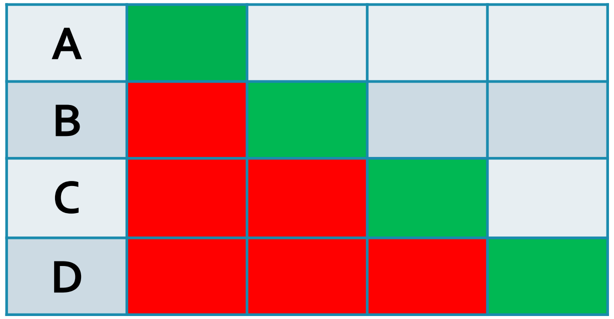
Da questo deriva che *(x,y)* **è una coppia instabile in M^***, e questa è una contraddizione.

LA COMPLESSITA' DELL'ALGORITMO

Per studiare la complessità dell'algoritmo ci dobbiamo concentrare sull'insieme dei ragazzi poiché è l'insieme su cui si basa il ciclo principale dell'algoritmo.

Il **caso peggiore** è quando ogni elemento si deve proporre a tutti gli altri elementi dell'altro insieme e questo accade quando tutti gli elementi dell'insieme X condividono lo stesso ordine di preferenza, quindi si propongono prima alla stessa ragazza la quale sceglie il suo preferito, successivamente gli altri n-1 elementi di X si propongono alle restanti n-1 ragazze e così via…

Si viene quindi a creare la configurazione che vediamo in figura:



da cui deriva che il numero di operazioni è pari a $\frac{n(n+1)}{2}$,

e si ha quindi che la **complessità è dell'ordine di n^2**.

ESEMPIO

- Sia X un insieme di 4 ragazzi: Alessio, Bruno, Claudio e Dario
- Sia Y un insieme di 4 ragazze: Elisa, Francesca, Giulia e Hilary
- Per brevità indicheremo: X= {A,B,C,D} e Y = {E,F,G,H}.
- Definiamo nelle seguenti tabelle le liste di preferenza (in ordine decrescente) degli elementi di X ed Y:

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

- **Passo 1:** Associamo a tutti gli elementi di Y l'elemento Ω che rappresenta l'elemento che occupa l'ultimo posto nella lista di preferenze di ogni elemento di Y.
- Ottenendo le coppie (Ω,E), (Ω,F), (Ω,G) e (Ω,H)

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

→ Alessio sceglie Francesca

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

→ Francesca accetta (poiché era abbinata con Ω, che sta all'ultimo posto nella sua lista di preferenza

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Bruno sceglie Giulia

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Giulia accetta (poiché era abbinata con Ω, che sta all'ultimo posto nella sua lista di preferenza)

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Claudio sceglie Giulia

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Giulia rifiuta Claudio, poiché preferisce Bruno

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Claudio sceglie Francesca

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Francesca accetta Claudio lasciando Alessio

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Alessio sceglie Elisa

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Elisa accetta (poiché era abbinata con Ω, che sta all'ultimo posto nella sua lista di preferenza)

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Dario sceglie Elisa

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Elisa accetta Dario e lascia Alessio

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Alessio sceglie Giulia

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Giulia accetta Alessio lasciando Bruno

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Bruno sceglie Hilary

A	F	E	G	H
B	G	H	F	E
C	G	F	H	E
D	E	H	G	F

E	C	D	B	A
F	C	B	A	D
G	A	B	C	D
H	D	C	B	A

Hilary accetta (poiché era abbinata con Ω, che sta all'ultimo posto nella sua lista di preferenza)

Tutti gli elementi dell'insieme X hanno un match e quindi l'algoritmo si ferma. Abbiamo quindi ottenuto il seguente matching stabile formato dalle coppie: (A,G),(B,H), (C,F), (D,E).

GENERALIZZAZIONI DEL PROBLEMA

Il modo in cui abbiamo definito il problema del matrimonio stabile presenta alcuni vincoli come il richiedere che la cardinalità dei due insiemi sia la stessa e che ciascun elemento di ogni insieme esprima una lista di preferenza completa rispetto a tutti gli elementi dell'altro insieme, che ne limitano l'applicabilità a contesti pratici più realistici, ma garantiscono che il problema abbia sempre una soluzione stabile.

Vogliamo ora introdurre due generalizzazioni di questo problema in modo da poterlo applicare a:

- Insiemi di cardinalità differente
- Liste di preferenza incomplete

Prima generalizzazione: insiemi di cardinalità differente

Supponiamo che i due insiemi **non hanno la stessa cardinalità**, ad esempio $|X| < |Y|$.
In questo modo abbiamo che qualche elemento di Y non avrà associato nessun elemento di X, quindi **l'idea è quella di consentire ad uno stesso elemento di X di essere associato a più di un elemento dell'insieme Y.**

In questo modo ci possiamo ricondurre al problema del matrimonio stabile con insiemi di uguale cardinalità, **costruendo un'estensione di X, ripetendo n_k volte ciascun elemento $x_k \in \mathbf{X}$, fino a costruire un insieme X' con la stessa cardinalità di Y:**

$$X' = \{x_k^{(i)}, i = 1,2, \ldots, n_k, \mathrm{k} = 1,2, \ldots, |\mathrm{X}|\}, \text{ con } \sum_{k=1}^{|X|} n_k = |Y|$$

dove la variabile i rappresenta il numero di volte che è ripetuto un elemento x_k, che può essere totalmente arbitrario.

Esempio:

Applicare questa generalizzazione ci permette di poter utilizzare l'algoritmo del matrimonio stabile per costruire un modello **per il problema dell'assegnazione del medico di base ai pazienti** di una determinata zona: ovviamente il numero di medici è molto inferiore rispetto al numero di persone, ma è anche vero che uno stesso medico può assistere più di un paziente.

Abbiamo quindi che **X = { insieme dei medici }** e **Y = { insieme dei pazienti },** ed il medico k-esimo dell'insieme X può assistere al massimo n_k pazienti.

Risulta quindi che $|\boldsymbol{X}| < |\boldsymbol{Y}|$ $\boldsymbol{e}$ $\boldsymbol{n_1 + n_2 + \ldots + n_k = |Y|}$.

Ciascun paziente indica la propria preferenza per i diversi medici basandosi su un *criterio soggettivo*, mentre la preferenza dei medici per i pazienti viene costruita applicando un *criterio oggettivo*, come ad esempio la distanza del domicilio del paziente dallo studio del medico.

L'algoritmo del matrimonio stabile ci darà la configurazione che scontenti il minor numero di pazienti avvantaggiando tra i pazienti chi preferisce un certo medico ed abita più vicino al suo studio.

Seconda generalizzazione: liste di preferenza incomplete

Supponiamo che a ciascun elemento $x_k \in X$ del primo insieme sia associata una lista di preferenze parziale P(x_k) sugli elementi dell'altro insieme e che lo stesso avvenga per ogni elemento $y_h \in Y$ del secondo insieme.

- **Obiettivo:** ricondurre ogni istanza del problema con liste di preferenza incomplete ad una *istanza equivalente* del problema del matrimonio stabile con liste di preferenza complete.
- **Idea:** aggiungere ad ognuno dei due insiemi un elemento, che chiameremo il *vedovo* e la *vedova* in modo tale da avere: $X = X \cup \{v_X\}, \mathrm{Y} = \mathrm{Y} \cup \{v_Y\}$.

Le liste di preferenza P(x_k) e P(y_h) degli elementi di X e di Y vengono completate nel seguente modo, in modo tale da ricondursi al caso del problema con liste di preferenza complete, che sappiamo già trattare:

- si colloca v_X come ultima preferenza di v_Y e viceversa;
- si colloca v_X come ultima preferenza di ciascun elemento y∈ Y, e viceversa;
- si aggiungono in un ordine arbitrario, dopo v_X tutti gli elementi mancanti alla lista di ciascun elemento y∈ Y, e viceversa.

Osservazione: La trasformazione effettuata nel riportare un'istanza del problema con liste incomplete ad un'istanza equivalente del problema con liste complete è corretta e non altera i termini del problema originale.

Vedremo ora che la soluzione dell'istanza del problema originale, con liste incomplete, esiste se e solo se esiste una soluzione all'istanza ottenuta completando le liste come descritto in precedenza, in cui gli elementi v_X e v_Y sono associati in una coppia del matching.

Teorema: Per la versione completa del problema esiste un matching stabile in cui v_X e v_Y sono associati se e solo se esiste un matching stabile nella versione incompleta del problema.

Dimostrazione:

$\Longrightarrow$ Se esiste un matching stabile per la versione completa, in cui (v_X, v_Y) costituiscono una coppia, allora lo stesso matching è stabile anche eliminando v_X e v_Y, questo perché visto che v_X è la scelta peggiore per v_Y e viceversa, evidentemente affinché l'istanza completa del problema abbia un matching stabile con v_X e v_Y accoppiati, significa che nessun altro elemento di X e Y preferisce rispettivamente v_X e v_Y al proprio partner.

Facciamo vedere ora che vale anche il viceversa:

$\Longleftarrow$ Se esiste un matching stabile per il problema con le liste incomplete allora lo stesso matching, con l'aggiunta della coppia (v_X, v_Y) è un matching stabile per il problema con le liste complete.

$\blacksquare$

Questo permette di poter prendere in considerazione tutti quei casi, applicazioni pratiche, in cui è veramente molto irrealistico che ciascun elemento dei due insiemi esprima una lista di preferenza completa su tutti gli elementi dell'altro insieme. Ad esempio, tornando all'esempio iniziale sulle coppie di ragazzi e ragazze è molto più probabile che ciascuno esprima la propria preferenza su un sottoinsieme di elementi dell'altro insieme, preferendo rimanere da solo piuttosto che finire in coppia con una persona non gradita.

POSSIBILI SVILUPPI

- Applicazione dei problemi di matching ai problemi di flusso;
- Equivalenza del teorema di Hall con il teorema di Menger;
- Algoritmi per matching massimi e matching pesati per la ricerca di matching di peso massimo/minimo;
- Approccio matroidale al problema;
- Dimostrazione della condizione sufficiente del teorema di Tutte.

Bibliografia:

- J. A. Bondy, U.S.R. Murty: *Graph theory*, Springer GTM 244.R
- R. Wilson: I*ntroduction to Graph theory*, Prentice Hall
- Diestel: *Graph theory*, Spriger GTM 173.

Sitografia:

- Il problema del matrimonio stabile di M. Liverani

GRAZIE PER L'ATTENZIONE!