



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

Università degli Studi di Padova

---

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

Corso di Laurea Triennale in Matematica

Stabilità rilassata in problemi di matching bipartiti

Relatore:  
Prof. Marco Di Summa

Laureando: Jacopo Caldana  
Matricola: 2017590

---

Anno Accademico 2022/2023

15/12/2023



# Indice

<b>Introduzione</b>	<b>iii</b>
<b>1 Introduzione ai problemi di matching stabili</b>	<b>1</b>
1.1 Algoritmo di Gale-Shapley . . . . .	1
1.2 Da HR a HRSS . . . . .	2
1.3 Approssimazione MAX HRSS . . . . .	4
<b>2 Casi speciali di HRSS</b>	<b>13</b>
2.1 $(2, \infty)$ -MAX SMISS . . . . .	13
2.2 HRSS con $ S  = k$ costante . . . . .	17
2.3 HRSS con $ C  = k$ costante . . . . .	18
<b>3 Matching (in)stabili con costi di blocco</b>	<b>23</b>
3.1 Rilassare pagando . . . . .	23
3.2 Quando pagare non aiuta . . . . .	25
3.2.1 Il problema Binary Unstable Matching (BUM) . . . . .	25
3.2.2 Il problema Preference-Concordant Unstable Matching (PCUM) . . . . .	27
3.3 Quando pagare diventa difficile . . . . .	29
3.4 Il problema Red-Blue Unstable Matching (R-B UM) . . . . .	35



# Introduzione

Il concetto di stabilità introdotto da Gale e Shapley è stato studiato approfonditamente nella teoria dei matching, dando vita a un vasto numero di estensioni del problema originale modellate sulle varie esigenze applicative.

Nella sua forma classica, il problema coinvolge  $n$  uomini ed  $n$  donne (che chiameremo agenti), ognuno dei quali esprime una lista di preferenze, che determina una relazione d'ordine totale sui membri del sesso opposto. Un *matching*  $M$  è un insieme di coppie (uomo,donna) tale che ogni agente appartiene rigorosamente a una sola coppia. Se  $(u, d) \in M$ , diciamo che  $d$  è la partner di  $u$ , e viceversa, inoltre scriviamo  $M(u) = d, M(d) = u$ .

Diciamo che un agente  $x$  preferisce  $y$  a  $y'$  se  $y$  precede  $y'$  nella lista di preferenze di  $x$ . Un matching  $M$  è *stabile* se non ammette *coppie bloccanti*, ovvero una coppia  $(u, d)$  tali che  $u$  preferisca  $d$  a  $M(u)$  e  $d$  preferisca  $u$  a  $M(d)$ .

Nonostante il concetto di stabilità sia importante in varie applicazioni di problemi di matching, ci sono classi di problemi per le quali non è assicurato che un'istanza ammetta un matching stabile. Peraltro, imporre il concetto di stabilità tende a ridurre le dimensioni dei matching ottenuti. Ciò risulta essere un problema in particolar modo nelle applicazioni in cui si desidera trovare un matching il più possibile grande.

Pertanto, in certi contesti, è auspicabile rilassare la definizione di stabilità. In questo scritto mostreremo delle strategie di rilassatezza computazionalmente trattabili. In primo luogo affronteremo il problema MAX HRSS, presentando un algoritmo di approssimazione per combattere la NP-difficoltà del problema. Successivamente, tratteremo casi speciali dello stesso problema MAX HRSS interessanti per la loro risolvibilità in tempo polinomiale.

Per poi concludere studiando problemi in matching bipartiti con pesi sugli archi, dove i pesi rappresentano l'importanza di includere o meno tale arco nel matching. Presenteremo una strategia di rilassatezza della stabilità basata sul pagare un prezzo per gli archi bloccanti del matching, massimizzando il valore dato dal peso degli archi del matching meno il costo degli archi bloccanti. Infine, mostreremo che ci sono casi in cui questa strategia porta ad algoritmi risolutivi polinomiali.



# Capitolo 1

## Introduzione ai problemi di matching stabili

In questo capitolo partiremo ricordando il classico Algoritmo di Gale-Shapley. Seguiremo introducendo il problema HR per poi condurci alla variante con Stabilità Sociale (HRSS). Infine, tratteremo il problema MAX HRSS, cercando un algoritmo di approssimazione per combatterne la NP-difficoltà.

### 1.1 Algoritmo di Gale-Shapley

In questa sezione verrà introdotto l'algoritmo di Gale-Shapley nella sua forma originale, facendo un richiamo alle definizioni principali, per permettere al lettore di riacquisire familiarità con l'argomento e le relative notazioni.

Ricordiamo che nella sua forma classica, il problema coinvolge  $n$  uomini ed  $n$  donne (che chiameremo *agenti*), ognuno dei quali esprime una *lista di preferenze*, che determina una relazione d'ordine totale sui membri del sesso opposto.

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

**Definizione.** Diciamo che un agente  $x$  preferisce  $y$  a  $y'$  se  $y$  precede  $y'$  nella lista di preferenze di  $x$ .

**Definizione 1.** Dato un grafo  $G(V, E)$  e una famiglia  $\{>_v\}_v \in V$  dove  $>_v$  è un ordinamento totale degli agenti, la coppia  $(G, >)$  è chiamata *sistema di preferenze*.

**Definizione 2.** Un matching  $M$  di  $G$  è *stabile* se non esistono *coppie bloccanti*  $(a, b')$ , ovvero non esiste una coppia che si preferisce a vicenda rispetto al suo partner attuale nel matching ( $b' >_a M(b)$ ,  $a >_{b'} M(b')$ , con potenzialmente  $M(b) = \emptyset$  o  $M(b') = \emptyset$  se non accoppiati).

**Definizione 3.** Un matching è un *matrimonio stabile* (stable marriage) se è stabile e perfetto ( $|M| = |V|/2$ ).

Gale e Shapley [Gale and Shapley, 1962] provarono che ogni istanza di matrimonio stabile ammette almeno un matching stabile, e descrissero un algoritmo che trova questo matching in tempo lineare.

**Algoritmo di Gale-Shapley:**

- **Input:**  $K_{n,n}$  con partizione  $(A, B) = (uomini, donne), \{>_v\}_{v \in A \cup B}$
- **Output:** matrimonio stabile  $M$
- **Ad ogni passo:** ogni uomo “si propone” alla donna che preferisce tra quelle che non l’hanno ancora rifiutato.
  - Se ogni donna riceve al più una proposta, l’algoritmo termina e restituisce il matching corrispondente.
  - Altrimenti, ogni donna dice “forse” all’uomo che preferisce tra quelli che le si sono proposti, e rifiuta gli altri.

dove  $K_{n,n}$  è il grafo bipartito completo di  $2n$  nodi.

Notiamo che: almeno un uomo viene rifiutato ad ogni step, e possono verificarsi al più  $|E| \leq n^2$  rifiuti, pertanto la complessità computazionale dell’algoritmo è  $O(n^2)$  nel caso peggiore. Questo sembrerebbe contraddire quanto affermato in precedenza sulla linearità dell’algoritmo ma, dato che le liste di preferenza in input hanno grandezza proporzionale a  $n^2$ , si ha che il tempo di esecuzione è proporzionale alla dimensione dell’input. Un uomo cambia proposta solo se viene rifiutato ed egli non può essere rifiutato da tutte le donne. Perché, dal momento che ci sono  $n$  donne e  $n$  uomini ed esso si è proposto a tutte le donne, questo implicherebbe la presenza di una coppia bloccante, ma questo contraddice il fatto che il matching ottenuto è stabile. Infine, nell’ultimo step, ogni donna riceve esattamente una proposta. Infatti essendoci  $n$  uomini e  $n$  donne e volendo ottenere un matching perfetto, non possono esistere due uomini che vogliono sposare la stessa donna all’ultimo step. L’algoritmo termina con un matching perfetto  $M$  dove ogni donna è sposata con l’uomo che preferisce tra quelli che le si sono proposti.

Concludiamo la sezione presentando l’Algoritmo di Gale-Shapley Esteso [Gusfield and Irving, 1989], che ci sarà utile più avanti, questo algoritmo ci permetterà di trovare matching stabili per istanze di problemi con liste incomplete in tempo polinomiale.

## 1.2 Da HR a HRSS

Il problema Ospedali/Residenti (HR), è anche conosciuto come ‘matrimonio poligamo’ o ‘ammissione al college’ ed è un problema many-to-one che può essere trasformato in uno one-to-one clonando gli ospedali con  $n$  posti in  $n$  ospedali diversi con un solo posto ciascuno, portando così il problema in forma classica.

In un’istanza del problema HR gli agenti sono partizionati in ospedali e residenti, ogni agente desidera essere accoppiato con un agente dell’altro insieme e ha una lista di preferenze sui possibili accoppiamenti. Inoltre ogni residente forma una coppia accettabile con ogni ospedale della sua lista e ogni ospedale ha una capacità, che indica il numero



---

```
Poniamo ogni persona come single ( $M = \emptyset$ );
while esiste un uomo  $m$  single do
   $w :=$  la prima donna nella lista di  $m$ ;
  if  $w$  è già accoppiata in  $M$  e  $m >_w M(w)$  then
    |  $M = M \setminus \{(M(w), w)\}$ 
  end
  if  $w$  è single in  $M$  then
    |  $M = M \cup \{(m, w)\}$ ;
  end
  for ogni uomo  $m'$  che viene dopo  $m$  nella lista di preferenze di  $w$  do
    | cancella la coppia  $(m', w)$  dall'istanza;
  end
end
Restituisce un matching stabile  $M$ .
```

---

massimo di residenti a cui può essere assegnato. Un matching è un insieme di coppie accettabili in cui non ci sono residenti accoppiati con più di un ospedale, e gli ospedali non eccedono la loro capacità. Il modello HR ha vari ambiti di applicazione, tra cui l'assegnazione di studenti di medicina agli ospedali, l'assegnazione di studenti alle high schools/college, ma veniva usato anche su piattaforme come AirBnB e su vari siti di incontri. In questi tipi di applicazioni è immediato notare che la stabilità è una caratteristica desiderata. Tuttavia esistono classi di problemi per cui l'esistenza di un matching stabile non è garantita. Inoltre, richiedere che il matching trovato sia stabile ne diminuisce la grandezza [Biró et al., 2010]. Questo risulta essere un problema, in particolar modo nelle applicazioni in cui è auspicabile trovare un matching il più possibile grande. Trattandosi di problemi da mettere in pratica nella vita reale è necessario sottolineare che una coppia residente-ospedale è più incline a bloccare un matching se esiste un legame sociale tra di loro. Queste osservazioni hanno portato a ricercare definizioni alternative, più deboli di stabilità.

Arcaute e Vasilvitskii [Arcaute and Vassilvitskii, 2009] contestualizzarono il problema HR nell'ambito dell'assegnamento di candidati lavoratori ad aziende. Osservarono che i candidati raccomandati da altri candidati amici che erano già riusciti a farsi assumere venivano scelti con più probabilità. Nel loro modello, una coppia candidato-azienda  $(c, a)$  poteva bloccare un matching  $M$  se era una coppia bloccante nel senso classico e  $c$  era amica di un agente  $c'$  già assegnato ad  $a$  in  $M$ . Strutturalmente il problema era composto dal classico HR e da una rete sociale, rappresentata da un grafo in cui i nodi rappresentavano i candidati e gli archi indicavano la presenza di legami sociali tra i vertici agli estremi. I matching che non ammettono coppie bloccanti nel senso nuovo vengono definiti *localmente stabili*. Il problema HR+SN fu studiato in maniera più approfondita da Cheng e McDermid [Cheng and McDermid, 2013] che stabilirono varie proprietà algoritmiche e risultati sulla complessità. Mostrarono che i matching localmente stabili potevano essere di varie grandezze, ma il problema di trovare il massimo era NP-difficile. Tuttavia trovarono dei casi speciali in cui il problema poteva essere risolto in tempo polinomiale e diedero delle

limitazioni dall'alto e dal basso sull'approssimabilità.

Il modello HR+SN assume che il candidato  $c'$  sia sempre disposto a fare una raccomandazione. Tuttavia potrebbe non essere così, infatti una raccomandazione potrebbe portare  $c'$  a essere rifiutato dall'azienda, facendolo accoppiare con un'azienda meno desiderata o addirittura lasciandolo disoccupato. Mentre è vero che se i legami sociali fossero tra azienda e candidato potrebbero portare a una coppia bloccante. Pertanto, un candidato per bloccare un matching non ha bisogno di conoscere un altro candidato che è stato assunto, gli basta conoscere un qualsiasi dipendente dell'azienda.

Sulla base di queste idee è stata introdotta una variante del problema HR chiamata *problema Ospedale/Residente con Stabilità Sociale* (HRSS). In questo modello, che descriviamo nel contesto dell'assegnazione di studenti di medicina laureati a posizioni in ospedale, assumiamo che una coppia residente-ospedale possa essere bloccante solo se esiste una relazione sociale tra loro. Due agenti che hanno una relazione sociale vengono chiamati *coppia conosciuta*, e sono rappresentati da un arco nel *grafo della rete sociale*. Viceversa due agenti che non hanno tale relazione li chiamiamo *coppia sconosciuta*. Tale coppia può essere parte di un matching  $M$  ma non può bloccarlo. Pertanto, nonostante una coppia residente-ospedale possa bloccare  $M$  nel senso classico, se è una *coppia sconosciuta*, non bloccherà il matching per il problema HRSS. Un matching senza coppie bloccanti in questo nuovo contesto viene definito *socialmente stabile*. Denotiamo la restrizione one-to-one del problema HRSS come il *problema di Matrimonio Stabile con liste Incomplete e Stabilità Sociale* (SMISS). Fino ad ora abbiamo sempre fatto riferimento a problemi con liste complete. Tuttavia, anche per i problemi di matrimonio stabile con liste incomplete (SMI), in cui il numero di uomini e donne può essere diverso e le loro liste di preferenza sono di lunghezza variabile senza necessariamente includere tutti i membri del sesso opposto, è possibile estendere l'algoritmo di Gale-Shapley (vedi [Gusfield and Irving, 1989]) in modo tale che restituisca un matching stabile. Data un'istanza del problema SMI. Un matrimonio  $M$  è stabile se è stabile nel senso classico e non esistono coppie  $(u, d)$  unmatched tali che  $u$  e  $d$  sono entrambi disaccoppiati in  $M$  nonostante  $(u, d)$  sia una coppia accettabile. Considerando che, come mostrato da Askalidis et al, il problema di trovare il massimo matching socialmente stabile data un'istanza  $I$  di HRSS, è NP-difficile [Askalidis et al., 2013]. Nella prossima sezione ci occuperemo dell'approssimabilità del problema MAX HRSS, dando un algoritmo che approssima di un fattore  $3/2$  il problema.

### 1.3 Approssimazione MAX HRSS

Come accennato nella sezione precedente, è possibile convertire un'istanza  $I$  di un problema HR in un'istanza  $I'$  di un problema di Matrimonio Stabile con liste Incomplete, in tempo polinomiale, tramite una tecnica chiamata *clonazione* [Gusfield and Irving, 1989][Sotomayor, 1990], in modo tale da ottenere una corrispondenza uno-a-uno tra l'insieme dei matching stabili in  $I$  e  $I'$ . In maniera analoga, è possibile passare da un'istanza HRSS a una SMISS in tempo polinomiale se si assume che le capacità degli ospedali sono limitate da una costante.

Sia  $(I, G)$  un'istanza di HRSS dove  $R = \{r_1, r_2, \dots, r_{n_1}\}$  è l'insieme dei residenti e  $H = \{h_1, h_2, \dots, h_{n_2}\}$  è l'insieme degli ospedali. Definiamo con  $c_j$  la capacità dell'o-

spedale  $h_j \in H$ .

Ora possiamo costruire un'istanza SMISS nel modo seguente:

- Ogni residente in  $(I, G)$  corrisponde a un uomo in  $(I', G')$ .
- Ogni ospedale  $h_j \in H$  dà origine a  $c_j$  donne chiamate  $h_{j,1}, h_{j,2}, \dots, h_{j,c_j}$  ognuna delle quali ha la stessa lista di preferenze di  $h_j$  in  $(I', G')$ , ma con capienza uno.
- Le liste di preferenza di ogni  $r_i \in R$  rimangono le stesse. Tuttavia, in questo passaggio, ogni entrata della lista verrà sostituita dalle  $c_j$  donne  $h_{j,1}, h_{j,2}, \dots, h_{j,c_j}$ .

$G'$  ora sarà composto dai vertici  $R \cup H$ , e dagli archi  $A' = \{(r_i, h_{j,k}) : (r_i, h_j) \in A, 1 \leq k \leq c_j\}$ , e denoteremo con  $C = E(G)$  l'insieme delle coppie conosciute in  $(I, G)$ .

**Teorema 4.** *Data un'istanza  $(I, G)$  di HRSS, è possibile costruire in tempo  $O(n_1 + c_{max}m)$  un'istanza  $(I', G')$  di SMISS tale che un matching socialmente stabile  $M$  di  $(I, G)$  può essere trasformato, in tempo  $O(c_{max}m)$ , in un matching socialmente stabile  $M'$  di  $(I', G)$  con  $|M| = |M'|$  e viceversa.*

*Dove  $n_1$  è il numero di residenti,  $c_{max}$  è il massimo delle capacità degli ospedali e  $m$  è il numero di coppie residente-ospedale accettabili in  $I$ .*

*Dimostrazione.* Siano  $(I, G)$  un'istanza di HRSS,  $(I', G')$  un'istanza di SMISS clonata da  $(I, G)$  e  $M$  un matching socialmente stabile in  $(I, G)$ . Costruiamo un matching  $M'$  in  $(I', G')$  nel modo seguente:

Per ogni  $h_j \in H$ , sia  $r_{j,1}, r_{j,2}, \dots, r_{j,x_j}$  l'insieme dei residenti assegnati a  $h_j$  in  $M$  dove  $x_j \leq c_j$ , e  $k < l$  implica che  $h_j$  preferisce  $r_{j,k}$  a  $r_{j,l}$ . Aggiungiamo  $(r_{j,k}, h_{j,k}^k)$  a  $M' \forall 1 \leq k \leq x_j$ . A questo punto è evidente che  $M'$  è un matching in  $(I', G')$  tale che  $|M'| = |M|$ , ed è immediato verificare che  $M'$  è socialmente stabile in  $(I, G)$ .

Viceversa, sia  $M'$  un matching socialmente stabile in  $(I', G')$ . Costruiamo un matching  $M$  in  $(I, G)$  nel modo seguente:

Per ogni  $(r_i, h_{j,k}^k) \in M'$ , aggiungiamo  $(r_i, h_j)$  a  $M$ . Anche in questo caso è evidente che  $M$  è un matching socialmente stabile in  $(I, G)$  tale che  $|M| = |M'|$ .

Le affermazioni sulla complessità computazionale derivano dal fatto che  $I'$  ha  $O(n_1 + K)$  agenti e  $O(c_{max}m)$  coppie accettabili, dove  $K$  è la capacità totale degli ospedali in  $I$ .  $\square$

Questo Teorema ci permette di passare da un tipo di problema all'altro senza particolari intoppi. Quindi se riuscissimo a trovare un algoritmo  $\alpha$  che approssimi il problema MAX SMISS lo potremmo usare per ottenere un'approssimazione di MAX HRSS con la stessa efficacia. Infatti, clonando l'istanza HRSS  $(I, G)$  per formare un'istanza SMISS  $(I', G')$  e applicando  $\alpha$  a  $(I', G')$  otterremo un matching  $M'$ . Questo matching può essere trasformato in un matching  $M$  di  $(I, G)$  tale che  $|M| = |M'|$ .

**Definizione.** Sia  $G = (V, E)$  un grafo, definisco con  $\beta^+(G)$  la cardinalità di un matching massimo di  $G$ .

Utilizzando il fatto che ogni matching stabile è almeno la metà di un massimo matching socialmente stabile si ottiene una prima stima del limite superiore per MAX HRSS. Questo fatto deriva da un noto risultato che mostreremo di seguito.

**Lemma.** *Dati due matching stabili qualunque, il rapporto tra le loro cardinalità è al massimo 2.*

*Dimostrazione.* Dato  $M$  matching stabile, supponiamo  $|M| = k$ . Siano  $n_u$  e  $n_d$  il numero di uomini e donne,  $|M| = k$  implica che tra gli  $n_u - k$  uomini single e le  $n_d - k$  donne single non c'è una coppia che si piace a vicenda, cioè tale che ciascuno dei due ha l'altro nella propria lista. Infatti, essendo  $M$  stabile se aggiungessimo una coppia di questo tipo non si creerebbero coppie bloccanti, poichè se tali agenti sono rimasti single significa che nessuno degli agenti accoppiati li preferisce al proprio partner. In altre parole, tra gli  $n_u - k$  uomini single e le  $n_d - k$  donne single non ci sono archi. Questo implica che ogni matching (generico anche non stabile) può avere al massimo  $2k$  archi.  $\square$

Tale argomentazione non cambia se si parla di due matching socialmente stabili, o di uno stabile e uno socialmente stabile.

**Proposizione 5.** *Il problema MAX HRSS è approssimabile entro un fattore 2.*

*Dimostrazione.* Sia  $M$  un matching socialmente stabile massimo di un'istanza  $(I, G)$  del problema SMISS e sia  $M'$  un matching stabile in  $I$ . Dunque  $M'$  è un matching massimale, nel grafo bipartito soggiacente  $G'$  in  $I$ . Perchè se si potesse aggiungere un arco a  $M'$ , vorrebbe dire che i due estremi sono un uomo e una donna compatibili e single in  $M'$ , ma questi formerebbero una coppia bloccante, il che contraddirebbe la stabilità. Quindi dal fatto che ogni matching massimale ha cardinalità almeno la metà di quella di un matching massimo (noto risultato della teoria dei matching che verrà dimostrato nel capitolo 3)  $|M'| \geq \beta^+(G')/2$  e da  $\beta^+(G') \geq |M|$  otteniamo  $|M'| \geq |M|/2$ .  $\square$

Mostreremo ora un algoritmo che fornisce un'approssimazione di un fattore  $3/2$  per un problema di MAX SMISS. Esso si basa in gran parte su un altro algoritmo già esistente, proposto da Kiraly [Király, 2013] e da McDermid [McDermid, 2009], che approssima dello stesso fattore il problema di trovare un matching di cardinalità massima per un'istanza del problema Ospedale-Residente con Legami (HRT).

L'algoritmo funziona chiamando una versione modificata dell'algoritmo di Gale-Shapley esteso, dove agli uomini non accoppiati viene data una seconda chance di trovare una compagna promuovendoli nelle liste di preferenza in cui sono presenti. Siano  $C$  e  $S$  rispettivamente gli insiemi delle coppie conosciute (ovvero con legami sociali) e sconosciute (senza legami) di un'istanza  $(I, G)$  di SMISS.

**Definizione.** Data una donna  $d_j$ , un uomo  $u_i$  nella sua lista di preferenze è rispettivamente un *uomo conosciuto* o un *uomo sconosciuto* se  $(u_i, d_j) \in C$  o  $(u_i, d_j) \in S$ .

Durante l'esecuzione dell'algoritmo se un uomo non ha più donne a cui proporsi per la prima volta, egli viene *promosso*, in questo modo gli viene permesso di proporsi alle

donne rimaste nella sua lista partendo dalla prima. Un uomo può essere promosso una volta sola durante l'esecuzione dell'algoritmo. Se anche dopo essere stato promosso ed essersi proposto alle restanti donne un uomo rimane single, egli viene rimosso dall'istanza e non farà parte del matching finale.

**Definizione.** Dato un uomo  $u_i$ , denotiamo con  $next(u_i)$  la donna successiva all'ultima a cui si è proposto nella sua lista di preferenze o la prima donna nella sua lista nel caso fosse appena stato promosso o non si fosse ancora proposto.

**Definizione.** Data una donna  $d_j$  e un uomo  $u_i$ , denotiamo con  $rank(d_j, u_i)$  la posizione occupata da  $u_i$  nella lista di preferenze di  $d_j$ .

*OSSERVAZIONE.* Nell'algoritmo di Gale-Shapley classico [Gale and Shapley, 1962] una donna  $d_j$  preferisce un uomo  $u_i$  a un altro  $u_k$  se  $rank(d_j, u_i) < rank(d_j, u_k)$ .

Nella versione modificata dell'algoritmo di Gale-Shapley esteso che verrà presentata, le donne accettano o rifiutano un uomo non solo sulla base della posizione che esso occupa nella loro lista di preferenze, ma anche in base al suo status di uomo conosciuto o sconosciuto e al fatto che sia stato promosso o meno. Prima di introdurre l'algoritmo è utile definire delle relazioni che serviranno a capire se la proposta di un uomo è stata accettata o rifiutata.

**Definizione 6.** Siano  $u_i$  e  $u_k$  due uomini nella lista di preferenza di una donna  $d_j$ . Allora

1.  $u_i \triangleleft_{d_j} u_k$  se vale una delle seguenti:
  - $(u_i, d_j) \in S$ ,  $(u_k, d_j) \in S$ , e  $u_i$  è stato promosso mentre  $u_k$  no, oppure
  - $(u_i, d_j) \in C$ ,  $(u_k, d_j) \in S$ , e  $u_k$  non è stato promosso.
2.  $u_i \triangleleft'_{d_j} u_k$  se  $u_i \not\triangleleft_{d_j} u_k$ ,  $u_k \not\triangleleft_{d_j} u_i$  e  $d_j$  preferisce  $u_i$  a  $u_k$  nel senso classico.
3. Definiamo ora  $\prec_{d_j} = \triangleleft_{d_j} \cup \triangleleft'_{d_j}$

La relazione  $\prec_{d_j}$  verrà usata per determinare se una proposta di un uomo viene accettata o rifiutata da  $d_j$ .

Presentiamo ora l'algoritmo *mod-EXGS*, modifica della versione estesa di Gale-Shapley, che verrà chiamato nell'algoritmo di approssimazione del problema SMISS.

In questo algoritmo viene fatta partire una sequenza di proposte permettendo a ogni uomo di proporsi alla prima donna della sua lista. Se un uomo  $u_i$  si propone a una donna  $d_j$  della sua lista ed essa è matched e  $u_i \prec_{d_j} M(d_j)$ , allora  $d_j$  lascia il suo partner  $u_k$  per accoppiarsi con  $u_i$  e  $u_k$  è libero di continuare a proporsi alle donne della sua lista. Se invece  $M(d_j) \prec_{d_j} u_i$  allora  $d_j$  rifiuta la proposta di  $u_i$ . Nel caso in cui  $d_j$  non è accoppiata con nessuno quando  $u_i$  le si propone, i due formeranno una nuova coppia. Indipendentemente dal fatto che la proposta di  $u_i$  venga accettata o meno, se  $(u_i, d_j) \in C$  allora tutte le coppie  $(u_k, d_j)$  tali che  $rank(d_j, u_k) > rank(d_j, u_i)$  vengono cancellate dall'istanza. Tuttavia se  $(u_i, d_j) \in S$ , ovvero non si conoscono, questa cancellazione non avviene.

---

**Algorithm 1:** mod-EXGS

---

```
while esiste un uomo  $u_i$  unmatched che ha ancora donne nella sua lista do
   $d_j = \text{next}(u_i)$ ;
  if  $d_j$  è matched in  $M$  e  $u_i \prec_{d_j} M(d_j)$  then
    |  $M = M \setminus \{(M(d_j), d_j)\}$ ;
  end
  if  $d_j$  è unmatched in  $M$  then
    |  $M = M \cup \{(u_i, d_j)\}$ ;
  end
  if  $(u_i, d_j) \in C$  then
    | for ogni  $u_k$  tale che  $(u_k, d_j) \in C$  e  $\text{rank}(d_j, u_k) > \text{rank}(d_j, u_i)$  do
    | | cancella  $(u_k, d_j)$  dall'istanza;
    | end
  end
end
end
```

---

---

**Algorithm 2:** Approx-SMISS

---

```
 $M = \emptyset$ ;
while esiste un uomo unmatched con una lista di preferenze non vuota do
  chiama mod-EXGS;
  for tutti gli  $u_i$  tali che  $u_i$  è unmatched e promosso do
  | rimuovi  $u_i$  dall'istanza
  end
  for tutti gli  $u_i$  tali che  $u_i$  è unmatched, non è stato promosso e ha una lista di
  preferenze non vuota do
  | promuovi  $u_i$ ;
  end
end
ritorna il matching  $M$ ;
```

---

Questa sequenza di proposte continua finchè ogni uomo è accoppiato o non ha più donne a cui proporsi.

L'algoritmo Approx-SMISS inizia chiamando mod-EXGS. Se dopo ogni sequenza di proposte un uomo promosso ha esaurito le donne a cui proporsi senza riuscire ad accoppiarsi esso viene rimosso dall'istanza. Nel caso in cui invece egli aveva esaurito la lista di donne a cui proporsi ma non era stato ancora promosso gli viene data una seconda chance promuovendolo e permettendogli di iniziare una nuova sequenza di proposte (chiamando mod-EXGS).

L'algoritmo termina quando ogni uomo o (i) è assegnato a una donna, (ii) non ha più donne nella sua lista di preferenze o (iii) è stato promosso e si è proposto a tutte le donne nella sua lista di preferenze per la seconda volta.

**Lemma 7.** *Per un'istanza  $(I, G)$  di SMISS l'algoritmo Approx-SMISS termina con un matching socialmente stabile  $M$  in  $(I, G)$ .*

*Dimostrazione.* Supponiamo che  $M$  non sia socialmente stabile e sia  $(u_i, d_j)$  una coppia "socialmente" bloccante di  $M$  in  $(I, G)$ . Allora  $(u_i, d_j) \in C$ .

(i) Se  $d_j$  è unmatched in  $M$  allora non ha mai ricevuto una proposta da  $u_i$ , altrimenti non avrebbe potuto diventare unmatched dopo. Questo significa che  $u_i$  preferisce la sua partner a  $d_j$  in  $M$ . Pertanto  $(u_i, d_j)$  non può bloccare "socialmente"  $M$  in questo caso.

(ii) Se invece supponiamo che  $d_j$  sia accoppiata in  $M$  ma preferisce  $u_i$  a

$M(d_j) = u_k$ , e che  $u_i$  è unmatched in  $M$  o preferisce  $d_j$  a  $M(u_i)$ . Allora  $u_i$  si era proposto a  $d_j$  durante l'esecuzione dell'algoritmo o  $(u_i, d_j)$  era stata cancellata. In entrambi i casi, tutti i gli uomini successivi a  $u_i$  nella lista di  $d_j$  vengono cancellati, quindi  $(u_k, d_j) \notin M$ , contraddizione.  $\square$

**Lemma 8.** *Durante ogni esecuzione dell'algoritmo mod-EXGS, se  $u_i$  si propone a  $d_j$  e  $(u_i, d_j) \in C$  allora  $d_j$  non rifiuterà mai  $u_i$  se  $\text{rank}(d_j, u_i) < \text{rank}(d_j, M(d_j))$ .*

*Dimostrazione.* Segue dalla definizione della relazione  $\prec_{d_j}$ . Supponiamo che  $d_j$  rifiuti  $u_i$  per un altro uomo  $u_k$  e  $\text{rank}(d_j, u_i) < \text{rank}(d_j, u_k)$ , perciò abbiamo  $u_k \prec_{d_j} u_i$ . Questo implica una delle seguenti relazioni  $u_k \triangleleft_{d_j} u_i$  o  $u_k \triangleleft'_{d_j} u_i$ . Poichè  $(u_i, d_j) \in C$  allora per come è definita la relazione  $u_k \not\triangleleft_{d_j} u_i$  perciò la relazione che vale è  $u_k \triangleleft'_{d_j} u_i$  la quale implica che  $\text{rank}(d_j, u_k) < \text{rank}(d_j, u_i)$ , che è una contraddizione alla nostra assunzione iniziale.  $\square$

**OSSERVAZIONE.** Il tempo di esecuzione dell'algoritmo mod-EXGS è  $O(m)$  dove  $m = |A|$  è il numero di coppie accettabili. L'algoritmo può essere eseguito al massimo  $2n_1$  volte, dove  $n_1$  è il numero di uomini, poichè a un uomo sono date al massimo due possibilità di proporsi alle donne della sua lista. Pertanto la complessità finale dell'algoritmo è  $O(n_1 m)$ .

I risultati riportati in precedenza ci permettono ora di introdurre questo teorema riguardante l'efficienza dell'approssimazione del problema MAX HRSS con l'utilizzo dell'algoritmo mostrato sopra.

**Teorema 9.** *Il problema MAX HRSS è approssimabile entro un fattore  $3/2$ .*

*Dimostrazione.* Consideriamo i cammini alternanti di lunghezza dispari, tra  $M_{opt}$  e  $M$ , nelle componenti connesse dell'unione  $M \cup M_{opt}$ . Ricordiamo che l'unione tra due matching ha come componenti connesse solo cicli alternanti di lunghezza pari e cammini alternanti. Tuttavia trascuriamo le componenti di lunghezza pari perchè in queste il rapporto tra la gli archi di  $M$  e  $M_{opt}$  è 1. Si nota che per cammini alternanti di lunghezza maggiore di 3, il numero di archi in  $M_{opt}$  è al massimo  $3/2$  volte il numero di archi in  $M$ . Se esistesse un cammino alternante di lunghezza 3 si avrebbe  $|M_{opt}| = 2|M|$ , il che contraddirebbe la nostra ipotesi. Mostriamo dunque che non può esistere un cammino alternante di lunghezza 3 in  $M \cup M_{opt}$ .

Consideriamo un cammino alternante di lunghezza 3  $\langle (u, d'), (u, d), (u', d) \rangle$  tale che  $(u, d') \in M_{opt}$ ,  $(u, d) \in M$  e  $(u', d) \in M_{opt}$ . Se  $d'$  fosse accoppiata in  $M$  avremmo un cammino alternante di lunghezza maggiore, pertanto possiamo affermare che  $d'$  è unmatched in  $M$ , perciò nessuno le si è mai proposto durante l'esecuzione dell'algoritmo. Quindi  $d'$  non ha mai cancellato nessun uomo dalla sua lista di preferenze e  $u$  non è stato promosso (altrimenti si sarebbe proposto a  $d'$ ). Dunque  $u$  preferisce  $d$  a  $d'$ . Essendo che anche  $u'$  è unmatched in  $M$  per lo stesso motivo (esisterebbe un cammino di lunghezza maggiore), le possibilità sono due: (i) la coppia  $(u', d)$  è stata cancellata dall'istanza durante l'esecuzione dell'algoritmo o (ii)  $d$  ha rifiutato  $u'$  due volte durante l'esecuzione dell'algoritmo.

**Caso (i):** se  $(u', d)$  è stato cancellato durante l'esecuzione significa che  $d$  ha ricevuto una proposta da un uomo  $u''$  tale che  $(u'', d) \in C$  e  $d$  preferisce  $u''$  a  $u'$ . Perciò tutti i successori di  $u''$  nella lista di  $d$  verrebbero cancellati, quindi possiamo concludere che  $d$  preferisce  $u$  a  $u''$  e di conseguenza anche a  $u'$ . Per il Lemma 8,  $d$  accetterebbe temporaneamente la proposta di  $u''$ , tuttavia alla fine la troviamo accoppiata con  $u$  un uomo non promosso. Supponiamo di non essere nel caso  $u = u''$ , allora dopo la proposta di  $u''$ ,  $d$  ha accettato un'altra serie di proposte  $\langle u''_0, u''_1, \dots, u''_k \rangle$  per qualche  $k \geq 0$  prima di essere assegnata definitivamente a  $u$ . Quindi  $u''_0 \prec_d u$ , che implica  $u''_0 \triangleleft_d u''$  o  $u''_0 \triangleleft'_d u''$ . Poichè  $(u'', d) \in C$ ,  $u''_0 \not\triangleleft_d u''$  che implica  $u''_0 \triangleleft'_d u''$ . Per definizione della relazione  $\triangleleft'$  quindi vale anche  $u'' \not\triangleleft_d u''_0$ . Affinchè valga questa relazione  $u''_0$  non deve essere un uomo sconosciuto e non promosso nella lista di  $d$ . Pertanto  $u''_0$  può essere un uomo conosciuto o un uomo sconosciuto promosso nella lista di  $d$ . Se  $u = u''_0$  allora sappiamo che  $u$  non è stato promosso e perciò  $(u, d) \in C$ . Se invece esiste un altro uomo  $u''_1$  nella sequenza, allora vale la stessa argomentazione che  $u''_1 \prec_d u''_0$ , implica  $u''_1 \triangleleft_d u''_0$  o  $u''_1 \triangleleft'_d u''_0$ . Come osservato in precedenza,  $u''_0$  è un uomo conosciuto o sconosciuto promosso, nella lista di  $d$ . In entrambi i casi,  $u''_1 \not\triangleleft_d u''_0$  che implica  $u''_1 \triangleleft'_d u''_0$ . Per definizione quindi vale anche  $u''_0 \not\triangleleft_d u''_1$ . Pertanto, come visto prima,  $u''_1$  può essere un uomo conosciuto o uno sconosciuto promosso nella lista di  $d$ . Lo stesso processo può continuare per tutti gli uomini nella sequenza finchè non si arriva ad  $u$ . Poichè sappiamo già che  $u$  non è stato promosso, segue che  $(u, d) \in C$ .

**Caso(ii):**  $d$  rifiuta  $u'$  anche quando è stato promosso perchè le si è proposto un altro uomo  $u''$ . Quindi  $u'' \prec_d u'$  che significa  $u'' \triangleleft_d u'$  o  $u'' \triangleleft'_d u'$ . Sapendo che  $u'$  è stato promosso, allora  $u'' \not\triangleleft_d u'$  quindi  $u'' \triangleleft'_d u'$ . Questo significa che  $u' \not\triangleleft_d u''$  e  $d$  preferisce  $u''$  a  $u'$ . Quindi  $u''$  deve essere un uomo promosso o conosciuto nella lista di  $d$ . Se  $u''$  fosse un uomo conosciuto nella lista di  $d$ , la coppia  $(u', d)$  verrebbe cancellata e ne conseguirebbe quanto mostrato nel Caso(i). Ora supponiamo  $u''$  sia un uomo sconosciuto e promosso nella lista



di  $d$ . Sappiamo che  $u \neq u''$  perchè  $u$  non era stato promosso in  $M$ . Allora  $d$  ha accettato un'altra serie di proposte  $\langle u''_0, u''_1, \dots, u''_k \rangle$  per qualche  $k \geq 0$  prima di essere assegnata definitivamente a  $u$ . Quindi si ha  $u''_0 \prec_d u''$ , che implica  $u''_0 \triangleleft_d u''$  o  $u''_0 \triangleleft'_d u''$ . Essendo  $u''$  un uomo promosso e sconosciuto nella lista di  $d$  possiamo affermare che  $u''_0 \not\triangleleft_d u''$  e quindi la relazione che sussiste è  $u''_0 \triangleleft'_d u''$ . Questo implica, per definizione, che  $u'' \not\triangleleft_d u''_0$ . Affinchè valga questa relazione  $u''_0$  non deve essere un uomo sconosciuto e non promosso nella lista di  $d$ . A questo punto possiamo utilizzare un'argomentazione del tutto simile a quella del Caso(i) per concludere che  $(u, d) \in C$ .

Le seguenti condizioni: 1)  $u$  preferisce  $d$  a  $d'$ , 2)  $d$  preferisce  $u$  a  $u'$  e 3)  $(u, d) \in C$  implicano che  $(u, d)$  blocca socialmente il matching  $M_{opt}$ , il che è una contraddizione.  $\square$

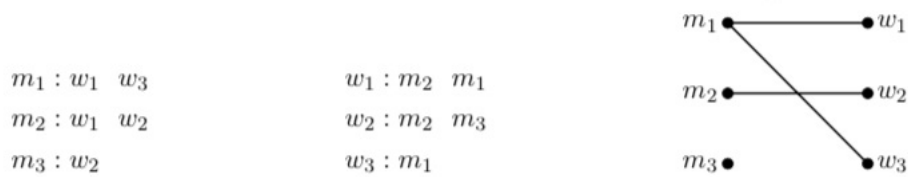


Figura 1.1:  $|M_{opt}| = (3/2)|M|$

L'istanza SMISS mostrata in figura ci dimostra la strettezza del limite di approssimazione  $3/2$  per l'algoritmo. Infatti,  $M_{opt} = \{(m_1, w_3), (m_2, w_1), (m_3, w_2)\}$  è l'unico matching socialmente stabile massimo, e l'algoritmo di approssimazione restituisce il matching  $M = \{(m_1, w_1), (m_2, w_2)\}$ .



# Capitolo 2

## Casi speciali di HRSS

Nella **Sezione 1.3** del capitolo precedente abbiamo introdotto un algoritmo di approssimazione di MAX HRSS per combattere la NP-difficoltà del problema. Tuttavia esistono casi speciali di HRSS che risultano particolarmente simpatici nella loro trattabilità. In questo capitolo presenteremo tre casi speciali di MAX HRSS risolvibili in tempo polinomiale.

### 2.1 $(2, \infty)$ -MAX SMISS

Data un'istanza del problema SMISS  $(I, G)$ , dove gli uomini possono avere al massimo due donne nella loro lista di preferenze e la lista delle donne può essere illimitata, mostreremo che si può trovare un matching socialmente stabile massimo in tempo polinomiale. Chiameremo  $(2, \infty)$ -MAX-SMISS-*alg* l'algoritmo, introdotto in [Askalidis et al., 2013], che ci permetterà di trovare tale matching. Esso è suddiviso in tre fasi che ora introdurremo.

**Fase1:** le coppie che non possono fare parte di un matching socialmente stabile in  $(I, G)$  vengono *cancellate* dall'istanza. Una coppia  $(u_i, d_j)$  viene cancellata dall'istanza rimuovendo  $u_i$  dalla lista di preferenze di  $d_j$  e viceversa. Le cancellazioni seguono una strategia precisa. Per ogni uomo  $u_i$ , se la prima donna nella sua lista  $d_j$  soddisfa  $(u_i, d_j) \in C$ , dove  $C = E(G)$  è l'insieme delle coppie conosciute in  $(I, G)$ , cancelliamo tutte le coppie  $(u_k, d_j)$  per tutti gli  $u_k$  che seguono  $u_i$  nella lista di preferenze di  $d_j$ . Infatti le coppie  $(u_k, d_j)$  non possono fare parte di un matching socialmente stabile perché  $(u_i, d_j)$  lo bloccherebbe. Una volta completato questo processo per tutte queste coppie si ottengono le *liste di preferenza ridotte*.

**Fase2:** dall'istanza con liste di preferenza ridotte ottenuta nella Fase1, costruiamo un grafo bipartito con pesi  $G'$ . Questo sarà costruito ponendo gli uomini da una parte e le donne dall'altra, come nell'istanza generale, e aggiungendo archi tra  $u_i$  e  $d_j$  qualora  $d_j$  appaia nella lista di preferenze di  $u_i$ . I pesi sugli archi rappresenteranno la posizione di  $u_i$  nella lista di preferenze di  $d_j$  (denotata con  $rank(u_i, d_j)$ ). Sotto introdurremo l'algoritmo *buildGraph* che descrive questo processo. A questo punto utilizzeremo l'algoritmo descritto

in [Gabow and Tarjan, 1989] per ottenere un matching di massima cardinalità con minimo costo  $M_{G'}$ .

A questo punto non è garantito che  $M_{G'}$  sia socialmente stabile. Infatti esso potrebbe avere una coppia bloccante  $(u_i, d_j)$ , dove  $(u_i, d_j) \in C$ , con  $u_i$  che viene assegnato alla sua seconda scelta  $d_k$  in  $M_{G'}$  nonostante la sua prima scelta  $d_j$  sia libera in  $M_{G'}$ .

---

**Algorithm 3:** buildGraph
 

---

```

 $V := U \cup D$ ; /*U e D sono gli insiemi di uomini e donne in  $I^*$ */
 $E' := \emptyset$ ;
for ogni uomo  $u_i \in U$  do
  | for ogni donna  $d_j$  nella lista ridotta di  $u_i$  do
  | |  $E' := E' \cup \{(u_i, d_j)\}$ ;
  | |  $peso(u_i, d_j) := rank(d_j, u_i)$ ;
  | end
end
 $G' := (V, E')$ ;
ritorna  $G'$ ;

```

---

**Fase3:** per rimuovere tali coppie bloccanti, in questa fase assegnamo  $u_i$  a  $d_j$  lasciando  $d_k$  disaccoppiata. Ora anche  $d_k$  potrebbe essere la prima scelta di un altro uomo  $u_l$  tale che  $(u_l, d_k) \in C$ . Il processo continua finchè non ci sono più uomini accoppiati con la loro seconda scelta nonostante essi abbiano un legame con la prima scelta in  $G$  ed essa sia libera.

Adesso mostreremo che alla fine dell'algorithmo, per istanze  $(2, \infty)$ -MAX-SMISS, il matching ottenuto è sia socialmente stabile che massimo.

**Lemma 10.** *L'algorithmo  $(2, \infty)$ -MAX-SMISS-*alg* termina.*

*Dimostrazione.* È immediato vedere che le Fasi 1 e 2 terminano. Nella Fase 3, ad ogni iterazione un uomo passa da essere accoppiato con la sua seconda preferenza alla sua prima senza che altri uomini peggiorino i loro accoppiamenti o rimangano disaccoppiati. Poichè il numero di questi miglioramenti è finito, ne deriva che anche questa fase termina.  $\square$

**Lemma 11.** *La Fase1 del  $(2, \infty)$ -MAX-SMISS-*alg* non cancella mai una coppia socialmente stabile, ovvero una coppia uomo-donna che appartiene a un qualunque matching socialmente stabile in  $(I, G)$ .*

*Dimostrazione.* Supponiamo che la coppia  $(u_i, d_j)$  venga cancellata durante un'esecuzione del  $(2, \infty)$ -MAX-SMISS-*alg* e  $(u_i, d_j) \in M$  dove  $M$  è un qualche matching socialmente stabile in  $(I, G)$ . Allora  $u_i$  viene cancellato dalla lista di preferenze di  $d_j$ , perché  $d_j$  è la prima scelta di un altro uomo  $u_k$  tale che  $(u_k, d_j) \in C$  e  $d_j$  preferisce  $u_k$  a  $u_i$ . Questo significa che  $(u_k, d_j)$  blocca socialmente  $M$ , il che è una contraddizione.  $\square$

---

**Algorithm 4:**  $(2, \infty)$  – MAX – SMISS – alg
 

---

```

/*Fase1*/
while qualche uomo  $u_i$  ha come prima preferenza una donna  $d_j$  tale che
   $(u_i, d_j) \in C$  do
  for ogni successore  $u_k$  di  $u_i$  nella lista di  $d_j$  do
  |   cancella la coppia  $(u_k, d_j)$ ;
  end
end
/*Fase2*/
 $G' := \text{buildGraph}()$ ;
 $M_{G'} := \text{massimo matching di peso minimo in } G'$ ;
/*Fase3*/
 $M := M_{G'}$ ;
while esiste un uomo  $u_i$  che è accoppiato con la sua seconda scelta  $d_k$  e la sua
  prima scelta  $d_j$  è una donna disaccoppiata tale che  $(u_i, d_j) \in C$  do
  |    $M := M \setminus \{(u_i, d_k)\}$ ;
  |    $M := M \cup \{(u_i, d_j)\}$ ;
end
ritorna  $M$ ;
    
```

---

**Lemma 12.** *Il matching ritornato dal  $(2, \infty)$ -MAX-SMISS-alg è socialmente stabile in  $(I, G)$ .*

*Dimostrazione.* Supponiamo che il matching  $M$  prodotto dall’algoritmo  $(2, \infty)$ -MAX-SMISS-alg non sia socialmente stabile in  $(I, G)$ . Allora una qualche coppia  $(u_i, d_j) \in C$  blocca socialmente  $M$  in  $I$ . Se accade questo significa che ci troviamo in uno dei seguenti casi.

Caso(i):  $u_i$  e  $d_j$  sono entrambi non accoppiati in  $M$ . Allora  $u_i$  non è accoppiato neanche in  $M_{G'}$  e lo stesso vale per  $d_j$ , essa può essere unmatched dall’inizio o lo può essere diventata a causa di alcune operazioni eseguite nella Fase3 del  $(2, \infty)$ -MAX-SMISS-alg. Se  $d_j$  fosse inizialmente unmatched in  $M_{G'}$  allora  $M_{G'}$  potrebbe aumentare la sua grandezza aggiungendo la coppia  $(u_i, d_j)$  il che contraddice il fatto che  $M_{G'}$  sia di massima cardinalità. Supponiamo ora che  $d_j$  sia diventata unmatched a causa della Fase3. Chiamiamo  $u_{p1}$  l’uomo con cui era accoppiata in  $M_{G'}$ . Durante la Fase3,  $u_{p1}$  è stato assegnato alla sua prima preferenza  $d_{q1}$ . Se fino a quel momento  $d_{q1}$  fosse stata unmatched in  $M_{G'}$ , allora si potrebbe aumentare  $M_{G'}$  lungo il cammino  $(u_i, d_j), (d_j, u_{p1}), (u_{p1}, d_{q1})$  contraddicendo il fatto che  $M_{G'}$  sia massimo. Perciò  $d_{q1}$  è diventata unmatched solo a seguito di operazioni eseguite nella Fase3. Se  $d_{q1}$  era accoppiata con  $u_{p2}$  in  $M_{G'}$ , durante la Fase3  $u_{p2}$  deve essere stato assegnato alla sua prima preferenza. Usando un’argomentazione del tutto analoga possiamo dire che anche  $d_{q2}$  era inizialmente assegnata a un uomo in  $M_{G'}$ , e che esso l’ha lasciata per accoppiarsi con la sua prima preferenza. Essendo un numero limitato i possibili miglioramenti, questo processo in cui ogni uomo migliora la sua partner e non la peggiora mai può andare avanti un numero finito di volte, pertanto esiste un numero finito di donne che possono rimanere disaccoppiate in questo modo nella Fase3. Quindi ad un certo punto

ci sarà un uomo  $u_{p_s}$  che migliorerà la sua posizione accoppiandosi con una donna  $d_{q_s}$  che fino a quel momento era unmatched in  $M_{G'}$ . A questo punto potremmo costruire un cammino aumentante in  $G'$   $(u_i, d_j), (d_j, u_{p_1}), (u_{p_1}, d_{q_1}), (d_{q_1}, u_{p_2}), (u_{p_2}, d_{q_2}), \dots, (u_{p_s}, d_{q_s})$  che contraddirebbe il fatto che  $M_{G'}$  è di massima cardinalità.

Caso(ii):  $u_i$  è disaccoppiato in  $M$  e  $d_j$  preferisce  $u_i$  a  $u_l = M(d_j)$ . Allora  $u_i$  è unmatched anche in  $M_{G'}$ . Supponiamo che  $d_j$  sia accoppiata con  $u_l$  in  $M_{G'}$ . Questo implicherebbe che potremmo trovare un matching della stessa misura di  $M_{G'}$  ma di peso minore accoppiando  $u_i$  a  $d_j$  e lasciando  $u_l$  disaccoppiato, contraddicendo il fatto che  $M_{G'}$  è un matching di massima cardinalità e di peso minimo. Perciò  $d_j$  non è assegnato a  $u_l$  in  $M_{G'}$ . Allora  $d_j$  può essere unmatched in  $M_{G'}$  oppure è accoppiata con un altro uomo  $u_p$ , con  $u_p \neq u_l$  e  $u_p \neq u_i$ . Se fosse unmatched in  $M_{G'}$ , staremmo contraddicendo la massimalità di  $M_{G'}$ . Mentre se fosse assegnata a  $u_p$  in  $M_{G'}$ , dato che  $u_p$  non è più assegnato a  $d_j$  in  $M$ , significa che  $u_p$  ha migliorato la sua posizione accoppiandosi con la sua donna preferita  $d_q$  durante la Fase3. Quindi  $d_q$  era unmatched in  $M_{G'}$  oppure lo è diventata dopo che un uomo l'ha lasciata per la sua prima preferenza. Usando un'argomentazione simile a quella del caso(i), possiamo costruire un cammino aumentante che contraddice il fatto che  $M_{G'}$  è di massima cardinalità.

Caso(iii):  $u_i$  è assegnato a  $d_k$  in  $M$  e  $u_i$  preferisce  $d_j$  a  $d_k$  e  $d_j$  è unmatched in  $M$ . Dato che la lista di preferenza di  $u_i$  è di lunghezza 2,  $d_j$  è la sua prima preferenza e  $(u_i, d_j) \in C$  e questo caso soddisfa la condizione del ciclo while della Fase3, questa situazione non può mai verificarsi una volta che la Fase3 è terminata.

Caso(iv):  $u_i$  è assegnato a  $d_k$  in  $M$ ,  $u_i$  preferisce  $d_j$  a  $d_k$ ,  $d_j$  è assegnata a  $u_l$  in  $M$  e  $d_j$  preferisce  $u_i$  a  $u_l$ . Come prima, dato che la lunghezza della lista di  $u_i$  è 2, allora  $d_j$  è la prima scelta di  $u_i$  e  $(u_i, d_j) \in C$ . Pertanto  $u_l$  sarebbe dovuto essere cancellato dalla lista di preferenze di  $d_j$  nel ciclo while della Fase1.  $\square$

Dal Lemma 11 sappiamo che la Fase1 non cancella mai una coppia socialmente stabile, quindi un matching socialmente stabile massimo deve essere composto da coppie che appartengono alle liste ridotte. Poichè  $M_{G'}$  è un matching massimo e la Fase3 non riduce mai la grandezza di un matching, segue che il matching prodotto dall'algoritmo è di massima cardinalità. Inoltre dal Lemma 12 sappiamo che tale matching è socialmente stabile, quindi possiamo concludere affermando che l'algoritmo produce un matching socialmente stabile massimo in  $(I, G)$ .

*OSSERVAZIONE.* La complessità dell'algoritmo è dettata per la maggior parte dalla Fase2. Infatti, la complessità dell'algoritmo che trova il matching massimo di peso minimo in  $G' = (V, E')$  è  $O(\sqrt{|V|}|E'| \log|V|)$  [Gabow and Tarjan, 1989]. Sia  $n = |V| = n_1 + n_2$  il numero totale di uomini e donne. Dato che  $A$  l'insieme delle coppie accettabili soddisfa  $|A| \leq 2n_1 = O(n)$ , segue che la complessità dell'algoritmo  $(2, \infty)$ -MAX-SMISS-*alg* è  $O(n^{3/2} \log n)$ .

Con questa osservazione abbiamo dimostrato il teorema seguente.

**Teorema 13.** *Data un'istanza  $(I, G)$  di  $(2, \infty)$ -MAX-SMISS, l'algoritmo  $(2, \infty)$ -MAX-SMISS-*alg* genera un matching socialmente stabile massimo in  $O(n^{3/2} \log n)$  operazioni, dove  $n$  è il numero totale di residenti e ospedali in  $(I, G)$ .*

## 2.2 HRSS con $|S| = k$ costante

Consideriamo il caso particolare in cui l'insieme di coppie sconosciute  $S$  è esattamente l'insieme delle coppie accettabili  $A$  nell'istanza HR sottostante, si ha che l'insieme delle coppie conosciute soddisfa  $C = \emptyset$ , pertanto ogni matching trovato è socialmente stabile. Anche nel caso opposto in cui  $C = A$  e  $S = \emptyset$ , si ha che tutti i matching stabili nel senso classico sono socialmente stabili. In entrambi i casi, un matching socialmente stabile massimo può essere generato in tempo polinomiale. Ora mostreremo che, anche nel caso in cui il numero di coppie sconosciute è costante, è possibile generare un massimo matching socialmente stabile in tempo polinomiale.

Sia  $(I, G)$  un'istanza di HRSS e sia  $B \subseteq A$  un sottoinsieme delle coppie accettabili in  $I$ . Denotiamo con  $I \setminus B$  l'istanza HR ottenuta da  $I$  cancellando le coppie appartenenti a  $B$  dalle liste di preferenza in  $I$ . Presentiamo ora una proposizione che avrà un ruolo cruciale per stabilire la correttezza dell'algoritmo.

**Proposizione 14.** *Siano  $(I, G)$  un'istanza di HRSS e  $M$  un matching socialmente stabile in  $(I, G)$ . Allora esiste un insieme di coppie sconosciute  $\emptyset \neq S' \subseteq S$  tale che  $M$  è stabile in  $I' = I \setminus S'$ .*

*Viceversa supponiamo che  $M$  sia un matching stabile in  $I' = I \setminus S'$  per qualche  $S' \subseteq S$ . Allora  $M$  è socialmente stabile in  $(I, G)$ .*

*Dimostrazione.* Supponiamo  $M$  sia socialmente stabile in  $(I, G)$ . Sia  $S' = S \setminus (M \cap S) = S \setminus M$ . Dimostriamo che  $M$  è stabile in  $I' = I \setminus S'$ . Supponiamo che  $(r_i, h_j)$  blocchi  $M$  in  $I'$ , allora  $(r_i, h_j) \notin M$ . Da  $S' = S \setminus (M \cap S) = S \setminus M$  possiamo osservare che gli archi di  $I' = I \setminus S' = (C \cup S) \setminus (S \setminus M)$  sono quelli in  $C \cup M \cap S$ , e banalmente  $(r_i, h_j) \notin M \cap S$ . Quindi  $(r_i, h_j) \in C$ , di conseguenza  $M$  non può essere socialmente stabile in  $I$ , il che è una contraddizione.

Viceversa, supponiamo che  $M$  sia stabile in  $I' = I \setminus S'$  per qualche  $S' \subseteq S$ . Vogliamo affermare che  $M$  è socialmente stabile in  $(I, G)$ . Supponiamo che  $(r_i, h_j)$  blocchi  $M$  in  $(I, G)$ , allora  $(r_i, h_j) \in C$  quindi  $(r_i, h_j) \notin S$ . Perciò  $(r_i, h_j)$  è una coppia di  $I'$  e quindi essa blocca  $M$  anche in  $I'$ , il che è una contraddizione.  $\square$

Considerando tutti i sottoinsiemi  $S' \subseteq S$ , che formano i rispettivi  $I'$ , cerchiamo ora i matching stabili in ogni  $I'$ . Tenendo traccia del massimo matching stabile trovato otterremo un matching socialmente stabile massimo in  $(I, G)$ . Questa argomentazione ci porta al seguente teorema.

**Teorema 15.** *[Askalidis et al., 2013] Data un'istanza  $(I, G)$  di HRSS dove l'insieme delle coppie sconosciute  $S$  è di grandezza costante, un matching socialmente stabile massimo può essere generato in  $O(m)$  operazioni, dove  $m = |A|$  è il numero di coppie accettabili.*

*Dimostrazione.* Considerando tutti i sottoinsiemi  $S' \subseteq S$ , che formano i rispettivi  $I'$ , cercando i matching stabili in ogni  $I'$  e tenendo traccia del massimo matching stabile trovato otterremo un matching socialmente stabile massimo in  $(I, G)$ . Sia  $M$  il più grande matching stabile trovato tra tutti gli  $I' = I \setminus S'$  dove  $S' \subseteq S$ . Allora per quanto visto nella proposizione precedente,  $M$  è socialmente stabile in  $(I, G)$ . Supponiamo per assurdo

che  $M'$  sia un matching socialmente stabile in  $I$  con  $|M'| > |M|$ . Allora per la stessa proposizione esiste un sottoinsieme  $S'' \subseteq S$  tale che  $M'$  è stabile in  $I'' = I \setminus S''$ . Ciò significa che abbiamo trovato un sottoinsieme  $\bar{S}' = S''$  di  $S$  tale per cui in  $\bar{I}' = I \setminus \bar{S}'$  esiste un matching  $M'$  con  $|M'| > |M|$ . Ma questo contraddice il fatto che al variare di tutti gli  $S' \subseteq S$ ,  $M$  è il più grande matching stabile esistente tra tutti gli  $I' = I \setminus S'$ .

Per trovare tutti i sottoinsiemi di  $S$  è necessario un tempo  $O(2^k)$  dove  $k = |S|$ . Dalla dimostrazione dell'Algoritmo di Gale-Shapley esteso [Gusfield and Irving, 1989] sappiamo che formare  $I'$  e trovare un matching stabile in  $I'$  può essere fatto in  $O(|A|)$  operazioni dove  $A$  è l'insieme delle coppie accettabili in  $I$ . Perciò la complessità totale dell'algoritmo è  $O(2^k|A|)$  che è polinomiale nella grandezza dell'istanza essendo  $k$  costante.  $\square$

## 2.3 HRSS con $|C| = k$ costante

In questa sezione tratteremo la restrizione di HRSS in cui l'insieme delle coppie conosciute  $C$  è di grandezza costante  $k$ . Data un'istanza  $(I, G)$  di questo problema dimostreremo che è possibile trovare un matching socialmente stabile massimo in tempo polinomiale se si assume che le capacità degli ospedali sono limitate da una costante.

Sia  $C = \{e_1, e_2, \dots, e_k\}$  dove  $e_i$  rappresenta una coppia conosciuta  $(r_{s_i}, h_{t_i})$  ( $1 \leq i \leq k$ ). Si costruisce un albero  $T$  di profondità  $k$  dove tutti i nodi a profondità  $i$  vengono chiamati  $e_{i+1}$  ( $i \geq 0$ ). I rami sotto  $e_i$  corrispondono a condizioni poste su  $r_{s_i}$  o  $h_{t_i}$  in relazione a un matching  $M$ . Il ramo sinistro corrisponde alla condizione che  $r_{s_i}$  è accoppiato in  $M$  e preferisce il suo partner a  $h_{t_i}$ , mentre il ramo destro corrisponde alla condizione che  $h_{t_i}$  ha riempito i suoi posti e non ha partner peggiori di  $r_{s_i}$ . La realizzazione di una di queste due condizioni ci assicura che  $M$  non ammette coppie bloccanti che coinvolgono  $(r_{s_i}, h_{t_i})$ . In questo albero i nodi  $e_k$  a profondità  $k - 1$  si diramano verso nodi foglia fittizi  $e_{k+1}$  che non rappresentano coppie conosciute.

Un cammino  $P$  dalla radice  $e_1$  a una foglia  $e_{k+1}$  visita tutte le coppie in  $C$  esattamente una volta. Ogni ramo sinistro dà una condizione sul residente e ogni ramo destro ne dà una sull'ospedale. Siano  $R'$  e  $H'$  l'insieme rispettivamente dei residenti e degli ospedali coinvolti nelle condizioni in  $P$ . Dato un matching  $M$ , per far rispettare tutte le condizioni presenti lungo  $P$  si cancellano dall'istanza  $I$  tutte le coppie che potrebbero potenzialmente violare tali condizioni. Se, per esempio, una condizione sul residente in  $P$  afferma che un residente  $r_{s_i}$  deve essere accoppiato in  $M$  con un ospedale che preferisce rispetto a  $h_{t_i}$  allora la lista di preferenze di  $r_{s_i}$  viene troncata a partire da  $h_{t_i}$ . Allo stesso modo, se una condizione sull'ospedale afferma che un ospedale  $h_{t_i}$  deve essere al completo in  $M$  e che non deve avere partner peggiori di  $r_{s_i}$  allora la lista di preferenze di  $h_{t_i}$  viene troncata a partire dal residente successivo a  $r_{s_i}$ . A seguito di queste riduzioni delle liste basate sulle condizioni lungo  $P$ , si ottiene una nuova istanza  $I'$  del problema HR.

**Proposizione 16.** *Se  $M$  è un matching in  $I'$  calcolato in un nodo foglia di un cammino  $P$  e tutti i residenti in  $R'$  sono accoppiati in  $M$  e tutti gli ospedali in  $H'$  hanno riempito i loro posti in  $M$ , allora  $M$  è un matching socialmente stabile in  $(I, G)$ .*

*Dimostrazione.* Supponiamo che una coppia residente-ospedale blocchi socialmente  $M$  in  $(I, G)$ . Allora questa coppia appartiene a  $C$  pertanto corrisponde a un nodo  $e_i = (r_{s_i}, h_{t_i})$



in  $T$  per qualche  $i$  ( $1 \leq i \leq k$ ). Quindi in  $M$  si ha che (i)  $r_{s_i}$  è unmatched o preferisce  $h_{t_i}$  a  $M(r_{s_i})$  e (ii)  $h_{t_i}$  non è al completo oppure preferisce  $r_{s_i}$  a  $M(h_{t_i})$ . Supponiamo che, lungo un cammino  $P$  dalla radice a una foglia di  $T$ , all'altezza di  $e_i$  sia stato scelto di percorrere il ramo sinistro. Questo significa  $r_{s_i} \in R'$ , per ipotesi  $r_{s_i}$  è accoppiato e, per le troncature apportate, ha un partner migliore di  $h_{t_i}$  ( $M(r_{s_i}) >_{r_{s_i}} h_{t_i}$ ), una contraddizione. Quindi nel percorso di  $P$  è stato scelto il ramo destro di  $e_i$ . Allora  $h_{t_i} \in H'$ ,  $h_{t_i}$  per ipotesi è al completo e dalle troncature fatte possiamo affermare che  $h_{t_i}$  non ha partner peggiori di  $r_{s_i}$ . Ma  $(r_{s_i}, h_{t_i}) \notin M$ , perciò  $h_{t_i}$  ha un partner migliore di  $r_{s_i}$  ( $M(h_{t_i}) >_{h_{t_i}} r_{s_i}$ ), il che contraddice il fatto che  $(r_{s_i}, h_{t_i})$  sia una coppia bloccante.  $\square$

Con  $I'$  ottenuto dalle troncature legate alle condizioni da soddisfare lungo il cammino  $P$  dalla radice a una foglia, vogliamo ottenere un matching in cui tutti i residenti in  $R'$  sono accoppiati e tutti gli ospedali in  $H'$  sono al completo. Procediamo clonando ogni ospedale  $h_j$  creando l'insieme  $\{h_{j,1}, h_{j,2}, \dots, h_{j,c_j}\}$  corrispondente al numero di posti  $c_j$  disponibili in  $h_j$ . Sia  $H'' = \{h_{j,k} : h_j \in H', 1 \leq k \leq c_j\}$  l'insieme di tutti i cloni ottenuti dagli ospedali in  $H'$ . Definiamo un grafo bipartito  $G'$  dove un insieme di nodi è rappresentato dall'insieme dei residenti in  $I'$  e l'altro insieme dai cloni degli ospedali in  $I'$ . Se  $h_j$  risulta accettabile per  $r_i$  in  $I$ , allora aggiungiamo un arco da  $r_i$  a  $h_{j,q}$  per ogni  $q$  ( $1 \leq q \leq c_j$ ). Definiamo ora un nuovo grafo  $G''$  che contiene gli stessi nodi e archi di  $G'$  avente però pesi sugli archi. A questo punto segniamo tutti i nodi che rappresentano i residenti in  $R'$  e i cloni degli ospedali in  $H''$  di colore rosso lasciando i rimanenti agenti incolore. In questo modo possiamo assegnare i pesi nel modo seguente: (i) un arco tra un nodo rosso e uno non colorato ha peso 1; (ii) un arco tra due nodi rossi ha peso 2; (iii) un arco tra due nodi non colorati ha peso 0. Ora troviamo un matching di peso massimo  $M'$  nel grafo bipartito ottenuto  $G''$ . Denotiamo con  $wt(M')$  il peso di un matching  $M'$  in  $G''$ . Allora

$$wt(M') = |\{(r_i, h_{j,q}) \in M' : r_i \in R'\}| + |\{(r_i, h_{j,q}) \in M' : h_{j,q} \in H''\}| \leq |R'| + |H''|$$

Inoltre  $wt(M') = |R'| + |H''|$  se e solo se ogni agente in  $R' \cup H''$  è accoppiato in  $M'$ . Quindi, per costruzione, un tale matching  $M'$  in  $G''$  è anche un matching in  $G'$  e un matching di massima cardinalità  $M''$  in  $G'$  può essere ottenuto aumentando in continuazione  $M'$  finchè non si riesce più a trovare un cammino aumentante. Dal momento che ogni nodo accoppiato in  $M'$  rimane accoppiato anche in  $M''$  segue che tutti gli agenti in  $R' \cup H''$  saranno accoppiati in  $M''$  e tale matching, per la Proposizione 16, sarà socialmente stabile in  $(I, G)$ . Tuttavia se,  $wt(M') < |R'| + |H''|$  allora in ogni matching massimo in  $G'$  qualche agente in  $R' \cup H''$  rimane disaccoppiato rendendo così possibile l'esistenza di una coppia bloccante di  $M'$  in  $(I, G)$ . In questo caso,  $P$  viene considerato *impercorsibile* e si considera un altro cammino *percorsibile*.

Ci sono  $2^k$  cammini dalla radice a una foglia nell'albero  $T$ . La seguente proposizione ci assicura che almeno uno di questi è *percorsibile*.

**Proposizione 17.** *Deve esistere almeno un cammino percorsibile in  $T$ .*

*Dimostrazione.* Sia  $M$  un matching stabile in  $I$ , allora  $M$  è socialmente stabile in  $(I, G)$ . Consideriamo ogni nodo  $e_i = (r_{s_i}, h_{t_i})$  in  $T$ . Se  $(r_{s_i}, h_{t_i}) \in M$ , diramiamo a destra di  $e_i$ .

Se  $(r_{s_i}, h_{t_j}) \notin M$ , e  $r_{s_i}$  è accoppiato ed ha un partner migliore di  $h_{t_j}$ , diramiamo a sinistra di  $e_i$ . Se  $(r_{s_i}, h_{t_j}) \notin M$ , e  $h_{t_j}$  è al completo e non ha partner peggiori di  $r_{s_i}$ , diramiamo a destra di  $e_i$ . Ogni altra condizione implicherebbe che  $(r_{s_i}, h_{t_j})$  blocchi  $M$  in  $I$ , una contraddizione. Seguire questo processo a partire dalla radice  $e_1$  ci permette di ottenere un cammino  $P$  percorribile in  $T$  fino a un nodo foglia  $v$  dove, per l'insieme dei residenti  $R'$  e degli ospedali  $H'$  coinvolti in  $P$ , si ha  $wt(M) = |R'| + |H'|$ .  $\square$

Per generare un matching socialmente stabile massimo  $M$  in un istanza  $(I, G)$  di HRSS, si considerano tutti i  $2^k$  cammini possibili in  $T$  dalla radice a una foglia tenendo traccia del massimo matching  $M''$  calcolato all'altezza della foglia di ogni cammino percorribile. Il matching desiderato  $M$  può essere costruito considerando

$$M = \{(r_i, h_j) : (r_i, h_{j,k}) \in M'' \text{ per qualche } k (1 \leq k \leq c_j)\}$$

**Proposizione 18.** *Se  $M$  è un matching ottenuto col processo descritto sopra,  $M$  è un matching socialmente stabile massimo in  $(I, G)$ .*

*Dimostrazione.* La Proposizione 16 ci mostra che  $M$  è socialmente stabile in  $(I, G)$ . Supponiamo che  $M'$  sia un matching socialmente stabile in  $(I, G)$  tale che  $|M'| > |M|$ . Costruiamo un cammino percorribile  $P$  in  $T$  dalla radice a un nodo foglia  $v$ , diramando a sinistra o a destra di ogni nodo  $e_i = (r_{s_i}, h_{t_j})$  in base a quanto segue. Se  $r_{s_i}$  è accoppiato in  $M'$  a un ospedale migliore di  $h_{t_j}$ , diramiamo a sinistra. Altrimenti  $(r_{s_i}, h_{t_j})$  non è una coppia bloccante di  $M'$ , quindi  $h_{t_j}$  è al completo in  $M'$  e non ha partner peggiori di  $r_{s_i}$ , in tal caso diramiamo a destra. Costruiamo ora gli insiemi  $R'$  e  $H'$ . Per ogni ramo sinistro in  $P$  che coinvolge  $r_{s_i}$ , si aggiunge  $r_{s_i}$  a  $R'$  e per ogni ramo destro in  $P$  che coinvolge un ospedale  $h_{t_j}$ , si aggiunge  $h_{t_j}$  a  $H'$ . In questo modo il matching  $M'$  soddisfa la condizione che ogni residente  $r_{s_i} \in R'$  è accoppiato in  $M'$  con un ospedale migliore di  $h_{t_j}$  e ogni ospedale  $h_{t_j} \in H'$  è al completo e non è accoppiato con residenti peggiori di  $r_{s_i}$ . Perciò  $|M''| \geq |M'|$  e poichè  $|M'| > |M|$ , segue che  $M''$  contraddice la scelta di  $M$  come matching massimo.  $\square$

Da questa proposizione segue il risultato principale di questa sezione.

**Teorema 19.** *[Askalidis et al., 2013] Data un'istanza  $(I, G)$  di HRSS dove l'insieme  $C$  delle coppie conosciute soddisfa  $|C| = k$  per qualche costante  $k$ , è possibile generare un matching socialmente stabile massimo in tempo  $O(c_{max}m\sqrt{n_1 + K})$  dove  $n_1$  è il numero di residenti,  $m$  il numero di coppie accettabili,  $c_{max}$  è la capacità più grande degli ospedali e  $K$  è la capacità totale di tutti gli ospedali nell'istanza del problema.*

*Dimostrazione.* Da  $|C| = k$  costante si ha che anche il numero di foglie (e di conseguenza il numero di cammini da radice a foglia) è costante ( $2^k$ ). Per eseguire le troncature imposte dalle condizioni da rispettare lungo il cammino sono necessarie  $O(m)$  operazioni dove  $m$  è il numero di coppie accettabili in  $I$  (da Gale-Shapley esteso). Il numero di nodi  $n'$  e

archi  $m'$  in  $G'$  sono dati da:

$$m' = \sum_{j=1}^{n_2} |pref(h_j)c_j| \leq c_{max} \sum_{j=1}^{n_2} |pref(h_j)| \leq c_{max}m$$

$$n' = n_1 + \sum_{j=1}^{n_2} c_j = n_1 + K$$

dove  $n_1$  è il numero di residenti,  $n_2$  il numero degli ospedali,  $pref(h_j)$  è l'insieme dei residenti nella lista di preferenze di  $h_j$ ,  $c_{max}$  è la più grande capacità di un ospedale e  $K$  il numero totale di posti nell'istanza del problema. Dato che i pesi sugli archi sono di grandezza  $O(1)$ , il matching di peso massimo in  $G''$  può essere trovato in tempo  $O(m'\sqrt{n'})$  [Duan and Su, 2012]. Aumentare tale matching fino a trovarne uno di massima cardinalità in  $G'$  può essere fatto in tempo  $O(m'\sqrt{n'})$  [Hopcroft and Karp, 1973]. Perciò la complessità computazionale dell'algoritmo è  $O(c_{max}m\sqrt{n_1 + K})$ .  $\square$

**Definizione.** Un problema parametrizzato è un insieme  $L \subseteq \Sigma^* \times \Sigma^*$  dove  $\Sigma$  è un alfabeto fissato.

**Definizione.** Un problema parametrizzato  $L$  è FTP (fixed-parameter-tractable) se esiste una costante  $\alpha$  e un algoritmo che determina se  $(x, y) \in L$  in tempo  $f(|y|)|x|^\alpha$ , dove  $f: \mathbb{N} \rightarrow \mathbb{N}$  è una funzione arbitraria.

Dai risultati dei Teoremi 15 e 19, concludiamo mostrando l'esistenza di algoritmi FPT per MAX HRSS con due diverse parametrizzazioni. Intuitivamente un problema è FPT se è trattabile in modo efficiente (tempo polinomiale) per valori costanti di un fissato parametro.

**Teorema 20.** [Askalidis et al., 2013] MAX HRSS è FPT con parametro  $k$ , dove  $k = |C|$  o  $k = |S|$ , e  $C$  e  $S$  sono gli insiemi di coppie rispettivamente conosciute e sconosciute.



## Capitolo 3

# Matching (in)stabili con costi di blocco

Nei capitoli precedenti abbiamo visto che, a meno di casi particolari, il problema di trovare il più grande matching socialmente stabile è NP-difficile ma approssimabile entro un fattore di  $3/2$ . Questi e altri esempi in letteratura, vedi [Biró et al., 2010] [Hamada et al., 2009], sembrano suggerirci che voler rilassare la stabilità mentre si cerca di mantenere un controllo sugli archi bloccanti ci porta a ottenere modelli computazionalmente intrattabili. In questo capitolo presenteremo un approccio che smentisce questa affermazione. Infatti, tratteremo problemi che si occupano di rilassare la stabilità con metodi computazionalmente efficaci e di restringere i costi dovuti alla presenza di archi bloccanti.

### 3.1 Rilassare pagando

Il nostro obiettivo è quello di scegliere il matching  $M$  che massimizza il ricavo prodotto dagli archi in  $M$ , meno il costo derivato dagli archi che bloccano  $M$ .

**Definizione.** Dato un matching  $M$  in un grafo bipartito a liste complete  $G(V, E)$ , definiamo con  $B_M$  l'insieme di archi di  $G$  che bloccano  $M$ .

**Definizione.** Dato un insieme di archi  $S$ , definiamo  $p(S) := \sum_{e \in S} p(e)$ , dove  $p(e)$  è il peso associato all'arco  $e$ .

Per prima cosa consideriamo il seguente problema di *General Unstable Matching*:

**Dati:** un sistema di preferenze  $(G(V, E), >)$ ,  $p, c: E \rightarrow \mathbb{Z}$ .

**Trovare:** un matching  $M$  di  $G$  che massimizza  $p(M) - c(B_M)$ .

Nella formulazione generale entrambe le funzioni possono assumere valori negativi. Tuttavia se  $p(e) = \alpha$ ,  $c(e) = \beta$  per ogni  $e$ , dal momento che ogni istanza di un problema che cerca un matching di massima cardinalità con il minor numero di archi bloccanti (MMSBP) può essere trasformata in un istanza di *General Unstable Matching* ponendo la condizione  $\alpha \gg \beta > 0$ , si ha che tale problema è NP-Difficile (essendolo MMSBP[Biró et al., 2010]). Pertanto è necessario cercare delle versioni semplificate.

Supponiamo che a ogni arco è associato un unico peso  $p(e) \geq 0$ , che rappresenta sia il guadagno se  $e \in M$  che il costo subito se  $e$  blocca  $M$ . Quindi il valore del matching  $M$  è  $f(M) := p(M) - p(B_M)$ .

A questo punto consideriamo tre diversi problemi. Primo, restringiamo l'immagine di  $p$  in  $\{0, 1\}$ , in questo modo otterremo il problema *Binary Unstable Matching*, che include il caso speciale in cui  $p(e) = 1$  per ogni  $e \in E$ .

**Dati:** un sistema di preferenze  $(G(V, E), >)$  e una funzione  $p: E \rightarrow \{0, 1\}$ .

**Trovare:** un matching  $M$  di  $(G(V, E), >)$  che massimizza  $f(M)$ .

Nel secondo problema interpretiamo  $p(e)$  come il grado di importanza dell'arco  $e$  nella lista di preferenze degli endpoint.

**Definizione.** Diciamo che  $p: E \rightarrow \mathbb{Z}_{\geq 0}$  è a *preferenze concordanti* se, per ogni  $xy, xy' \in E$ ,  $y >_x y'$  quando  $p(xy) > p(xy')$ .

Il problema *Preference Concordant Unstable Matching* è definito nel modo seguente:

**Dato:** un sistema di preferenze  $(G(V, E), >)$  e una funzione  $p: E \rightarrow \mathbb{Z}_{\geq 0}$ .

**Trovare:** un matching  $M$  di  $(G(V, E), >)$  che massimizza  $f(M)$ .

Procediamo enunciando il Teorema 21 che ci fornisce i primi due risultati interessanti.

**Teorema 21.** *I problemi Binary Unstable Matching e Preference Concordant Unstable Matching sono risolvibili in tempo polinomiale.*

Alla luce di questi risultati, siamo tentati di rilassare le assunzioni sulla funzione  $p$ . Tuttavia, procediamo mostrando che anche un minimo tentativo di modifica del problema *Binary Unstable Matching* ci conduce a un problema difficile da approssimare. Consideriamo il seguente problema  $\beta$ -*Binary Unstable Matching*:

**Dati:** un sistema di preferenze  $(G(V, E), >)$ ,  $\beta \in (0, 1)$  e una funzione  $p: E \rightarrow \{\beta, 1\}$ .

**Trovare:** un matching  $M$  di  $(G(V, E), >)$  che massimizza  $f(M)$ .

**Teorema 22.** *Il problema  $\beta$ -Binary Unstable Matching è APX-Difficile.*

Concludiamo la sezione considerando un modo alternativo di restringere il problema *General Unstable Matching*, dove ogni arco soddisfa  $c(e) = \infty$ ,  $p(e) \geq 0$ , oppure  $p(e) = -\infty$ ,  $c(e) = 0$ . Gli archi di  $E$  possono dunque essere partizionati in "blu"- che possono fare parte del matching  $M$ , ma non possono bloccarlo - e "rossi"- che possono bloccare  $M$  con un costo, ma non possono farne parte. Chiamiamo quindi il modello risultante col nome *Red-Blue Unstable Matching* e lo definiamo usando la funzione peso  $p$ .

**Dati:** un sistema di preferenze  $(G(V, E), >)$ , una funzione  $p: E \rightarrow \mathbb{Z}_{\geq 0}$  e un insieme  $F \subseteq E$  ("archi rossi").

**Trovare:** un matching stabile  $M$  di  $(G(V, E \setminus F), >)$  che massimizza  $f(M)$ .

**Teorema 23.** *Il problema Red-Blue Unstable Matching può essere risolto in tempo polinomiale.*

I teoremi presentati in questa sezione verranno dimostrati nelle sezioni seguenti.

*OSSERVAZIONE* (Notazione). Introduciamo delle notazioni utili alla comprensione del capitolo. Dati  $M$  e  $S$  due matching di  $G(V, E)$ , denotiamo con  $B_M^S := S \cap B_M$  e  $V(S)$  l'insieme di nodi accoppiati in  $S$ . Per  $v \in V$  diciamo che  $v$  preferisce  $S$  a  $M$  se  $S(v) >_v M(v)$ . Per  $n \in \mathbb{N}$ , scriviamo  $[n] := \{1, \dots, n\}$  e  $[n]_0 = [n] \cup \{0\}$ . Infine per descrivere la differenza simmetrica utilizziamo il simbolo classico  $\Delta$ .

## 3.2 Quando pagare non aiuta

### 3.2.1 Il problema Binary Unstable Matching (BUM)

Presentiamo ora un lemma che mette in relazione la grandezza di un matching con il numero di archi bloccanti.

**Lemma 24.** *Siano  $M, S$  due matching di  $(G(V, E), >)$ , con  $S$  stabile.*

*Allora  $|B_M^S| \geq |M| - |S|$ .*

*Dimostrazione.* Dati  $M', M''$  matching di  $G$  e  $x \in V$ , definiamo  $voto(x, M', M'') = 1$  (risp.  $-1$ ) se  $x$  preferisce  $M'$  a  $M''$  (risp.  $M''$  a  $M'$ ), e  $voto(x, M', M'') = 0$  altrimenti. Per  $e = mw \in E$  definiamo  $voto(e, M', M'') := voto(m, M', M'') + voto(w, M', M'')$ . Infine definiamo  $voto(M', M'') := \sum_{v \in V} voto(v, M', M'')$ . Da [Huang and Kavitha, 2013] sappiamo che  $voto(S, M) \geq 0$ . Quindi, decomponendo  $voto(S, M)$  nella partizione di  $V$  data da

$(V(S), V(M) \setminus V(S), V \setminus (V(S) \cup V(M)))$ , otteniamo:

$$\begin{aligned} 0 &\leq \underbrace{\sum_{v \in V(S)} voto(v, S, M)}_{=\sum_{e \in S} voto(e, S, M)} + \underbrace{\sum_{v \in V(M) \setminus V(S)} voto(v, S, M)}_{=-2|M \setminus S|} + \underbrace{\sum_{v \in V \setminus (V(S) \cup V(M))} voto(v, S, M)}_{=0} \\ &\leq \sum_{e \in S} voto(e, S, M) - 2(|M| - |S|), \end{aligned}$$

dove l'ultima disequazione deriva dal fatto che  $|M \setminus S| \geq |M| - |S|$ .

Poichè  $voto(e, S, M) \in \{0, \pm 2\}$  per ogni  $e \in S$ , devono esistere almeno  $|M| - |S|$  archi tali che  $voto(e, S, M) = 2$ . Tutti questi sono archi di  $S$  che bloccano  $M$ .  $\square$

Come conseguenza di questo lemma, sappiamo che, quando  $p(e) = 1$  per ogni  $e \in E$ , è ottimale avere come output un qualsiasi matching stabile, poichè l'aumento di peso dovuto all'aumento della grandezza del matching viene bilanciato dalla presenza degli archi bloccanti introdotti. Col prossimo lemma mostreremo che il Lemma 24 ci rivela anche una strategia ottima per il problema *Binary Unstable Matching* che consiste nel non pagare per tutti gli archi  $e$  con  $p(e) = 1$  e non includere nel matching gli archi  $e$  con  $p(e) = 0$ .

**Lemma 25.** *Sia  $I$  un'istanza del problema Binary Unstable Matching ed  $E_1 := \{e \in E : p(e) = 1\}$ . Allora ogni matching stabile di  $G(V, E_1)$  è una soluzione ottima di  $I$ .*

*Dimostrazione.* Definiamo  $E_0 := E \setminus E_1$  e, per un insieme  $F \subseteq E_0$ ,  $G_F := G(V, E_1 \cup F)$ . Sia  $M$  un matching di  $G$ , definiamo  $F(M) := M \cap E_0$  e denotiamo con  $S'$  un matching stabile di  $G_{F(M)}$ . Allora:

$$f(M) = p(M) - P(B_M) = |M \cap E_1| - |B_M \cap E_1| \quad (3.1)$$

$$= |M| - |F(M)| - |B_M \cap E(G_{F(M)})| \quad (3.2)$$

$$\leq |M| - |F(M)| + |S'| - |M| \quad (3.3)$$

$$\leq |S' \cap E_1| \quad (3.4)$$

$$= p(S'), \quad (3.5)$$

dove l'uguaglianza della seconda riga deriva direttamente dalla definizione di  $F(M)$  e da  $F(M) \cap B_M = \emptyset$ , la disuguaglianza della terza riga si ottiene applicando il Lemma 24 al grafo  $G_{F(M)}$  e ai matching  $M, S'$ , la disuguaglianza della quarta riga deriva da  $S' \cap E_0 \subseteq F(M)$ , mentre l'ultima uguaglianza viene dalla definizione di  $p$ . Osservando che, essendo  $S'$  stabile in  $G_{F(M)}$ , nessuno degli archi di  $F(M) \cup E_1$  blocca  $S'$ , ne deduciamo che:

$$p(S') = p(S') - p(B_{S'} \cap E_0) = p(S') - p(B_{S'}) = f(S').$$

Questa catena di uguaglianze, insieme alle relazioni precedenti (vedi (3.5)), implica l'esistenza di una soluzione ottima per  $I$  nella forma di un matching stabile di  $G_F$ , per qualche  $F \subseteq E_0$ . Sia ora  $S_\emptyset$  un matching stabile di  $G(V, E_1) = G_\emptyset$  e  $S_F$  un matching stabile di  $G_F$ , con  $F \subseteq E_0$ . Notiamo che, per  $S \in \{S_\emptyset, S_F\}$ , abbiamo  $|V_S| = 2f(S)$ , dove  $V_S := \{v \in V : E_1 \cap \delta(v) \cap S = \emptyset\}$ . Questo deriva dal fatto che, come visto in precedenza, per  $S = S_F$  (per  $S_\emptyset$  è banalmente uguale) la funzione  $f(S)$  ci restituisce il numero di archi di peso 1.

A questo punto, per concludere la dimostrazione, è sufficiente mostrare che  $|V_{S_F}| \leq |V_{S_\emptyset}|$ , così facendo mostreremo che effettivamente è conveniente scegliere matching composti solo da archi di peso 1 perché, per la relazione  $|V_S| = 2f(S)$ , si massimizzerà il valore di  $f(S)$ . Supponiamo per assurdo che non sia così, e definiamo  $G' := (V, (E_1 \cap S_F) \Delta (E_1 \cap S_\emptyset))$ . Deve esistere una componente connessa di  $G'$  con strettamente più nodi appartenenti a  $V_{S_F}$  rispetto a  $V_{S_\emptyset}$ . Poichè ogni nodo di  $V$  ha grado al massimo 2 in  $G'$ , questa componente connessa è un cammino  $P = v_0, v_1, \dots, v_t$  con  $t$  dispari, dove  $v_0, v_t$  sono accoppiati da  $E_1 \cap S_F$  e lasciati esposti da  $E_1 \cap S_\emptyset$ , mentre gli altri archi del cammino alternano tra  $E_1 \cap S_F$  e  $E_1 \cap S_\emptyset$ . Da  $E_1 \cap S_\emptyset = S_\emptyset$  si deduce che  $v_0$  e  $v_t$  non sono accoppiati in  $S_\emptyset$ . Quindi  $v_0$  e  $v_t$  preferiscono  $S_F$  a  $S_\emptyset$ . Se anche  $v_1$  preferisce  $S_F$  a  $S_\emptyset$ , allora  $v_0 v_1$  blocca  $S_\emptyset$ , contraddicendo il fatto che nessun arco di  $E_1$  blocca  $S_\emptyset$ . Pertanto  $v_1$  preferisce  $S_\emptyset$  a  $S_F$ . Ora poichè, per costruzione, nessun arco di  $E_1$  può bloccare  $S_F$ , possiamo iterare il ragionamento deducendone che  $v_i$  preferisce  $S_F$  a  $S_\emptyset$  per gli  $i$  pari e  $S_\emptyset$  a  $S_F$  per gli  $i$  dispari. Perciò  $v_t$  preferisce  $S_\emptyset$  a  $S_F$ , contraddizione.  $\square$

Concludiamo la sottosezione osservando che, se  $p(e) = 1$  per tutti gli archi  $e \in E$  eccetto un arco  $f$  con  $p(f) = \beta \in (0, 1)$ , il Lemma 25 non è più vero. Infatti, la figura sotto ci mostra un'istanza del problema  $\beta$ -Binary Unstable Matching in cui l'unica soluzione ottima è bloccata da un arco  $e$  con  $p(e) = 1$ . Questo ci dà una visione intuitiva del perchè il problema  $\beta$ -BUM è sostanzialmente più difficile del problema BUM.



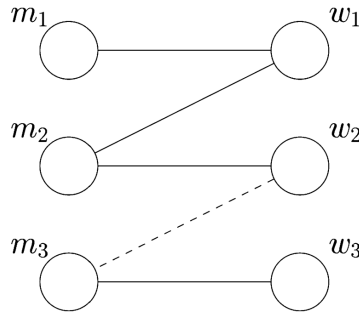


Figura 3.1: Poniamo  $p(e) = 1$  per tutti gli archi  $e$  ecceto l'arco tratteggiato  $f$  per il quale si ha  $p(f) = \beta \in (0, 1)$ .  $w_1$  e  $m_2$  si preferiscono a vicenda. Supponiamo che  $w_2$  preferisca  $m_2$  a  $m_3$ , e che  $m_3$  preferisca  $w_2$  a  $w_3$ . Sia  $M^* = \{m_1w_1, m_2w_2, m_3w_3\}$ . Allora  $B_{M^*} = m_2w_1$  e  $f(M^*) = p(M^*) - p(B_{M^*}) = 3 - 1 = 2$ .  $M^*$  è l'unica soluzione ottima, dal momento che ogni altro matching  $M$  soddisfa  $f(M) < 2$ . Quando  $\beta \in \{0, 1\}$ ,  $M^*$  continua ad essere una soluzione ottima ma, Lemma 24 e 25, implicano che essa non è più unica.

### 3.2.2 Il problema Preference-Concordant Unstable Matching (PCUM)

Sia  $(G, >)$  un sistema di preferenze con  $V(G) = \{m_1, w_1, m_2, w_2\}$  e  $E(G) = \{m_i, w_j\}_{i,j=1,2}$ . Poniamo  $p(m_1w_1) = 6$ ,  $p(m_1w_2) = p(m_2, w_1) = 5$  e  $p(m_2w_2) = 1$ . Utilizzando l'unico sistema di preferenze dato da  $>$  che rende  $p$  a preferenza concordante, il matching ottimo è  $\{m_1w_1, m_2w_2\}$ . Se invece avessimo  $w_2 >_{m_1} w_1$  e  $w_1 >_{m_2} w_2$ , il matching ottimo sarebbe  $\{m_1w_2, m_2w_1\}$ . Questo ci mostra che la preferenza concordante non può essere assunta senza perdita di generalità.

Ora presenteremo un algoritmo per il problema Preference-Concordant Unstable Matching che trova iterativamente un matching stabile nel sottografo con gli archi di peso massimo e rimuove i nodi accoppiati dal grafo. In seguito mostreremo che l'unione di tutti i matching costruiti dall'algoritmo ci dà una soluzione ottima del problema.

**Lemma 26.** *Per ogni istanza  $I$  del problema PCUM, l'Algoritmo 5 restituisce un Preference-Concordant Unstable Matching ottimo in  $I$ .*

*Dimostrazione.* Per provare questo lemma dimostreremo due Claim che ci saranno di supporto.

**Claim 1**  $S$  è un matching stabile di  $(G, >)$ .

*Dimostrazione.* Sappiamo banalmente che  $S$  è un matching di  $G$ . Assumiamo che  $S$  non sia stabile e sia  $mw$  un arco che blocca il matching, ovvero  $w >_m S(m)$  e  $m >_w S(w)$ . Sia  $i$  l'iterazione dell'Algoritmo 5 in cui il primo tra  $w$  e  $m$  si accoppia, ovvero  $i - 1$  è l'ultima iterazione in cui entrambi non sono accoppiati. Allora, senza perdita di generalità possiamo assumere  $mS(m) \in \bar{S}$  dove  $\bar{S}$  è un matching stabile nel grafo  $G'$  costruito all' $i$ -esima iterazione. Essendo  $mS(m)$  un arco di peso massimo in quell'iterazione ed essendo  $m, w$  entrambi unmatched prima dell'inizio di questa, deve per forza sussistere la relazione  $p(mS(m)) \geq p(mw)$ . Inoltre, dalla preferenza concordante sappiamo che  $w >_m S(m)$

**Algorithm 5:**

**Data:** Un'istanza  $I$  del problema Preference-Concordant Unstable Matching con:  
 un sistema di preferenze  $(G(V, E), >)$ , e una funzione a preferenza concordante  
 $p: E \rightarrow \mathbb{Z}_{\geq 0}$ .

**Restituisce:** Una soluzione ottima di  $I$ .

Sia  $S = \emptyset$ .

**while**  $G$  ha ancora un arco **do**

    Sia  $p^*$  il peso massimo di un arco di  $G$  e  $G'$  il sottografo di  $G$  composto dai  
     soli archi  $e$  con  $p(e) = p^*$ .

    Sia  $\bar{S}$  un matching stabile di  $G'$ .

$S = S \cup \bar{S}$ .

$V = V \setminus V(\bar{S})$ .

**end**

ritorna  $S$ .

implica  $p(mw) \geq p(mS(m))$ . Allora  $p(mw) = p(mS(m))$ , perciò  $mw$  è anche un arco del grafo  $G'$  nell' $i$ -esima iterazione e quindi blocca anch'esso  $\bar{S}$ , il che è una contraddizione.  $\square$

Fissato un matching  $M$  di  $G$ . Mostreremo che  $f(M) \leq f(S)$ . Consideriamo il grafo  $G_M(V, E_M) := G(V, S \Delta M)$  e sia  $C$  l'insieme delle sue componenti connesse non banali (no singoletti). La differenza simmetrica tra due matching ha come componenti connesse cicli e cammini alternanti. Infatti, per ciascun matching le componenti connesse sono formate solamente dai due vertici che formano ciascun arco. Perciò l'alternanza è necessaria per avere componenti connesse di  $S \Delta M$ . Dall'alternanza deriva che i cicli hanno lunghezza pari, poichè avremmo componenti connesse con due archi dello stesso matching adiacenti, assurdo. Per  $U \subseteq V$ , definiamo  $\Delta(U) := p(M(U)) - p(B_M^S(U)) - p(S(U))$ , e per  $F \subseteq E$ , poniamo  $F(U) := \{uv \in F : u, v \in U\}$ .

A questo punto vogliamo dimostrare che  $\Delta(U) \leq 0$  per tutti gli  $U \in C$ , dove con  $U$ , per semplificare, intendiamo sia i gli archi che i nodi. Il che dimostrerebbe che  $S$  fornisce il guadagno migliore. Usando il Claim 1 sappiamo che  $B_S = \emptyset$  e quindi:

$$\begin{aligned}
 f(M) - f(S) &= p(M) - p(B_M) - p(S) \\
 &\leq p(M) - p(B_M^S) - p(S) \\
 &= \sum_{U \in C} (p(M(U)) - p(B_M^S(U)) - p(S(U))) \\
 &= \sum_{U \in C} \Delta(U) \\
 &\leq 0.
 \end{aligned}$$

Supponiamo per assurdo che  $\Delta(U) > 0$  per qualche  $U \in C$ , prendiamo il controesempio minimo, ovvero un'istanza  $I$  e due matching  $M, S$  che violano questa affermazione, tali che il grafo sottostante  $G$  ha il numero minimo di nodi. Definiamo  $E_{max} := \{e \in E_M(U) : p(e) \geq p(f) \ \forall f \in E_M(U)\}$  e  $G_{max} := G(U, E_{max})$ . Se  $G_{max}$  è un ciclo, allora ha un numero pari di archi e  $\Delta(U) = p(M(U)) - p(B_M^S(U)) - p(S(U)) \leq p(M(U)) - p(S(U)) = 0$ ,

una contraddizione. Altrimenti, dato che  $G_{max}$  contiene almeno un arco ed ogni nodo in  $U$  ha grado al più 2 in  $G_M$ , e quindi anche in  $G_{max}$ , possiamo dedurre che  $G_{max}$  è un cammino  $P = v_0, v_1, \dots, v_t$ .

**Claim 2:**  $P \cap B_M^S \neq \emptyset$ .

*Dimostrazione.* Supponiamo  $v_0v_1 \in S$ , l'altro caso si svolge in maniera simile. Per prima cosa osserviamo che  $v_iv_{i+1}$  appartiene a  $S$  per gli  $i$  pari e a  $M$  per gli  $i$  dispari. Essendo  $v_0$  un estremo di  $P$ , allora o  $v_0$  è esposto in  $M$ , oppure  $p(v_0S(v_0)) > p(v_0M(v_0))$ . Nell'ultimo caso, dalla definizione di preferenza concordante, si ha  $S(v_0) >_{v_0} M(v_0)$ . Quindi, in entrambi i casi  $v_0$  preferisce  $S$  a  $M$ .

Supponiamo esista un indice  $j$  tale che sia  $v_j$  che  $v_{j+1}$  preferiscono lo stesso matching tra  $S$  e  $M$ , e poniamo  $j$  come il minimo indice in cui questo accade. Se  $j$  è dispari, poichè  $v_0$  preferisce  $S$  a  $M$ , si ha allora che  $v_j$  e  $v_{j+1}$  preferiscono  $M$  a  $S$ . Quindi come visto sopra  $v_jv_{j+1} \in M$ , pertanto essi bloccano  $S$ , il che contraddice il Claim 1. Se viceversa  $j$  è pari, allora  $v_j$  e  $v_{j+1}$  preferiscono  $S$  a  $M$ , e  $v_jv_{j+1} \in S$ . A questo punto abbiamo trovato l'arco cercato.

Assumiamo allora che i nodi di  $P$  preferiscno in modo alternato  $S$  a  $M$  e viceversa. Se  $t$  è dispari (risp. pari), allora  $v_{t-1}v_t \in S$  (risp.  $M$ ), mentre  $v_t$  preferisce  $M$  a  $S$  (risp.  $S$  a  $M$ ). In entrambi i casi, viene contraddetto il fatto che  $v_t$  sia un estremo di  $P$ .  $\square$

Sia  $mw \in P \cap B_M^S$ , e poniamo  $U'$  (ed eventualmente anche  $U''$ ) come le componenti connesse del sottografo di  $G_M$ , indotto da  $U \setminus \{m, w\}$ . Usando il Claim 2, abbiamo:

$$\begin{aligned} 0 &< \Delta(U) \\ &= \Delta(U') + \Delta(U'') - 2p(mw) + p(mM(m)) + p(M(w)w) \\ &\leq \Delta(U') + \Delta(U''). \end{aligned}$$

Dove la prima disuguaglianza deriva dal fatto che  $mw \in B_M^S$  e l'ultima sussiste poichè  $mw \in E_{max}$ . Notiamo che  $U''$  o  $e \in \{mM(m), M(w)w\}$  potrebbero non esistere, in tal caso poniamo  $\Delta(U'') = 0$  o  $p(e) = 0$ . In particolare, dalla catena di disequaglianze, possiamo assumere senza perdita di generalità che  $\Delta(U') > 0$ .

Sia  $V' = v \setminus \{m, w\}$  e consideriamo una sottoistanza  $I'$  di  $I$  in  $G'$ , il sottografo di  $G$  indotto da  $V'$ . Sia  $S' := S \setminus \{mw\}$ ,  $M' := M \setminus \{mM(m), M(w)w\}$ . Chiaramente,  $M'$  è un matching di  $G'$ . E' facile verificare che c'è un'esecuzione dell'Algoritmo 5 che, con input  $I'$ , ritorna  $S'$ . Per  $U \subseteq V'$ , definiamo  $\Delta'(\cdot)$  come  $\Delta(\cdot)$ , ma questa volta per l'istanza  $I'$  e i matching  $S', M'$ . Osserviamo che  $U'$  è una componente connessa di  $G'(V', S' \Delta M')$ , e  $\Delta'(U') = \Delta(U') > 0$ , il che contraddice la scelta di  $I$  come controesempio minimo. Perciò,  $\Delta(U) \leq 0$ .  $\square$

### 3.3 Quando pagare diventa difficile

Per convenienza, introduciamo una formulazione equivalente del problema  $\beta$ -Binary Unstable Matching.

**Dati:** un sistema di preferenze  $(G(V, E), >)$  e una funzione  $p: E \rightarrow \{\alpha, \beta\}$  dove  $\alpha > \beta > 0$ .

**Trovare:** un matching  $M$  di  $(G(V, E), >)$  che massimizza  $f(M)$ .

**Teorema 27.** *Il problema  $\beta$ -Binary Unstable Matching è NP-difficile per ogni  $\alpha > \beta > 0$  tale che  $\alpha > 4\beta$ , anche se la lista di preferenze è di lunghezza al massimo 3.*

Mostreremo il Teorema 27 utilizzando una riduzione dal caso speciale di SAT usato per il problema MMSBP[Biró et al., 2010]. Diversamente da MMSBP, dove gli input non hanno pesi, noi dobbiamo assegnare dei pesi  $\alpha$  o  $\beta$  agli archi in modo tale da ottenere, per l'istanza MMSBP, un matching ottimo che sia perfetto e non abbia archi bloccanti di peso  $\alpha$ .

Per una formula Booleana  $B$  in forma normale congiunta (CNF) e un'assegnazione di verità  $h$ ,  $t(h)$  è il numero di clausole di  $B$  soddisfatte da  $h$  e  $t(B)$  è il massimo valore di  $t(h)$  tra tutte le assegnazioni di verità. Sia MAX (2, 2)-E3-SAT [Berman et al., 2004] il problema NP-difficile che consiste nel trovare un'assegnazione di verità  $h$  tale che  $t(h) = t(B)$ , dove ogni clausola di  $B$  ha esattamente 3 letterali e ogni variabile compare esattamente due volte come letterale positivo (non-negato) e due volte come letterale negativo (negato).

Siano  $v_1, \dots, v_{n-1}$  e  $c_1, \dots, c_k$  rispettivamente gli insiemi delle variabili e delle clausole di un'istanza  $B$  di MAX (2, 2)-E3-SAT, dove ogni clausola ha esattamente tre letterali, e i letterali  $v_i$  e  $\bar{v}_i$  appaiono esattamente due volte. Creiamo da  $B$  un'istanza  $I$  del problema  $\beta$ -BUM introducendo 6 uomini e 6 donne per ogni variabile in  $B$  più 7 uomini e 7 donne per clausola, nel modo seguente. L'insieme degli uomini in  $I$  è  $P \cup U \cup X \cup Q$ :

- $P := \{p_j^s : j \in [k], s \in [3]\}$ ,
- $U := \{u_j^s : j \in [k], s \in [3]\}$ ,
- $X := \{x_{6i+r} : i \in [n-1]_0, r \in [6]\}$ ,
- $Q := \{q_j : j \in [k]\}$ .

L'insieme delle donne, invece, è  $C \cup W \cup Y \cup Z$ :

- $C := \{c_j^s : j \in [k], s \in [3]\}$ ,
- $W := \{w_j^s : j \in [k], s \in [3]\}$ ,
- $Y := \{y_{6i+r} : i \in [n-1]_0, r \in [6]\}$ ,
- $Z := \{z_j : j \in [k]\}$ .

Nella trattazione useremo anche i sottoinsiemi  $X_i := \{x_{6i+r} : r \in [6]\}$ ,  $Y_i := \{y_{6i+r} : r \in [6]\}$ ,  $P_j := \{p_j^1, p_j^2, p_j^3\}$ ,  $U_j := \{u_j^1, u_j^2, u_j^3\}$ ,  $C_j := \{c_j^1, c_j^2, c_j^3\}$ ,  $W_j := \{w_j^1, w_j^2, w_j^3\}$ , definiti per gli appropriati indici  $i, j$ . La lista di preferenze e i pesi degli archi sono indicati nella Tabella 1 in figura.

	$\alpha$	$\beta$	$\alpha$
$x_{6i+1}$ :	$y_{6i+1}$	$c(x_{6i+1})$	$y_{6i+2}$
$x_{6i+2}$ :	$y_{6i+2}$	$c(x_{6i+2})$	$y_{6i+3}$
$x_{6i+3}$ :	$y_{6i+4}$	$c(x_{6i+3})$	$y_{6i+3}$
$x_{6i+4}$ :	$y_{6i+5}$	$c(x_{6i+4})$	$y_{6i+4}$
$x_{6i+5}$ :		$y_{6i+5}$	$y_{6i+6}$
$x_{6i+6}$ :		$y_{6i+1}$	$y_{6i+6}$
	$\beta$	$\alpha$	$\alpha$
$y_{6i+1}$ :	$x_{6i+6}$	$x_{6i+1}$	
$y_{6i+2}$ :		$x_{6i+1}$	$x_{6i+2}$
$y_{6i+3}$ :		$x_{6i+3}$	$x_{6i+2}$
$y_{6i+4}$ :		$x_{6i+4}$	$x_{6i+3}$
$y_{6i+5}$ :	$x_{6i+5}$	$x_{6i+4}$	
$y_{6i+6}$ :		$x_{6i+5}$	$x_{6i+6}$
	$\beta$	$\beta$	$\alpha$
$p_j^1$ :	$w_j^1$	$c_j^1$	
$p_j^2$ :	$w_j^2$	$c_j^2$	
$p_j^3$ :	$w_j^3$	$c_j^3$	
$u_j^1$ :	$w_j^1$		$z_j$
$u_j^2$ :	$w_j^2$		$z_j$
$u_j^3$ :	$w_j^3$		$z_j$
$c_j^1$ :	$p_j^1$	$x(c_j^1)$	$q_j$
$c_j^2$ :	$p_j^2$	$x(c_j^2)$	$q_j$
$c_j^3$ :	$p_j^3$	$x(c_j^3)$	$q_j$
$w_j^1$ :	$u_j^1$	$p_j^1$	
$w_j^2$ :	$u_j^2$	$p_j^2$	
$w_j^3$ :	$u_j^3$	$p_j^3$	
	$\alpha$	$\alpha$	$\alpha$
$z_j$ :	$u_j^1$	$u_j^2$	$u_j^3$
$q_j$ :	$c_j^1$	$c_j^2$	$c_j^3$

Figura 3.2: Tabella 1

Nella lista di preferenze di un uomo  $x_{6i+r} \in X$ , con  $i \in [n]$  e  $r \in \{1, 2\}$ , il simbolo  $c(x_{6i+r})$  denota la donna  $c_j^s \in C$  tale che la  $r$ -esima ricorrenza della variabile  $v_i$  compare nella posizione  $s$  della clausola  $c_j$ . Similmente,  $c(x_{6i+r})$  per  $r \in \{3, 4\}$  denota la donna  $c_j^s \in C$  tale che la  $r - 2$ -esima ricorrenza della variabile  $\bar{v}_i$  compare nella posizione  $s$  di  $c_j$ . Di conseguenza,  $x(c_j^s)$  denota l'unico uomo nella lista di preferenze di  $c_j^s$ . Iniziamo la trattazione con un'osservazione che deriva dalla relazione  $\alpha > \beta > 0$ .

**OSSERVAZIONE 28.** Ogni soluzione ottima per  $I$  è un matching massimale.

A questo possiamo aggiungere due considerazioni importanti, sottoforma di osservazioni, sul peso degli archi nella nostra costruzione.

**OSSERVAZIONE 29.** Se un arco  $mw$  blocca un matching  $M$  e nessuno dei due agenti  $m, w$  è single in  $M$  allora  $p(mw) = \beta$ , poichè  $mw \in X \times C$  o  $mw \in U \times W$  o  $mw = x_{6i+5}y_{6i+5}$  o  $mw = x_{6i+6}y_{6i+1}$  per qualche  $i \in [n]$ .

**OSSERVAZIONE 30.** Per  $i \in [n]$ , gli unici matching perfetti di  $X_i \times Y_i$  sono  $T_i := \{x_{6i+l}y_{6i+l}\}_{l=1,\dots,6}$  e  $F_i := \{x_{6i+l}y_{6i+l+1}\}_{l=1,\dots,5} \cup \{x_{6i+6}y_{6i+1}\}$ .

Osserviamo che  $\bar{M} := \bigcup_{i=1}^n T_i \cup \{p_j^1 c_j^1, p_j^2 c_j^2, p_j^3, w_j^3, u_j^1 w_j^1, u_j^2 w_j^2, u_j^3 z_j, q_j c_j^3 : j \in [k]\}$  è un matching perfetto di  $I$  che contiene  $5n + 2k$  archi di peso  $\alpha$  e (dall'Osservazione 30) non è bloccato da archi di peso  $\alpha$ . Combinando le osservazioni precedenti con la relazione di partenza  $\alpha > 4\beta$  si ottengono i seguenti due risultati.

**Lemma 31.** *Un matching ottimo per  $I$  non contiene archi di  $X \times C$ .*

*Dimostrazione.* Sia  $M$  un matching ottimo e supponiamo, per assurdo, che esista un arco  $x_{6i+l}c_j^a \in M \cap (X \times C)$  per opportuni indici  $i, l, a, j$ . Consideriamo il matching  $M' := M \setminus ((X_i \times Y_i) \cup \{x_{6i+l}c_j^a\} \cup S^-) \cup T_i \cup S^+$ , dove,  $S^+ = \{c_j^a q_j\}$  e  $S^- = \{M(q_j)q_j\}$  se  $c_j^a >_{q_j} M(q_j)$ , e  $S^+ = S^- = \emptyset$  altrimenti. Vogliamo mostrare che  $f(M') > f(M)$ , per ottenere la contraddizione che ci permette di concludere.

Dimostriamo che  $p(M' \cap (X_i \times Y_i)) - p(M \cap (X_i \times Y_i)) \geq \alpha$ . Infatti, osservando che  $M' \cap (X_i \times Y_i) = T_i$  e  $p(T_i) = 5\alpha + \beta$ , e notando che  $M' \cap (X_i \times Y_i)$  contiene al massimo 4 archi di peso  $\alpha$  poichè  $x_{6i+l}$  non è accoppiato tramite archi di peso  $\alpha$ , entrambi  $x_{6i+5}, x_{6i+6}$  formano un arco di peso  $\alpha$  solo con  $y_{6i+6}$ . Dato che  $x_{6i+l}c_j^a \in M'$ , risulta che  $p(M \cap (X_i \times Y_i)) \leq 4\alpha + \beta$  il che dimostra la nostra tesi.

Ora dimostriamo che  $p(M' \setminus (X_i \times Y_i)) - p(M \setminus (X_i \times Y_i)) \geq -\beta$ . Se  $S^+ = \emptyset$ , non vengono rimossi archi di peso  $\alpha$  da  $M \setminus (X_i \times Y_i)$ , nel caso contrario  $M(q_j)q_j$  viene rimosso e viene aggiunto  $c_j^a q_j$ , entrambi archi di peso  $\alpha$ . Quindi il contributo degli archi di peso  $\alpha$  è lo stesso sia in  $p(M' \setminus (X_i \times Y_i))$  che in  $p(M \setminus (X_i \times Y_i))$ . L'unica rimozione di un arco di peso  $\beta$ , per passare da  $M' \setminus (X_i \times Y_i)$  a  $M \setminus (X_i \times Y_i)$ , è quella di  $x_{6i+l}c_j^a$ . A questo punto anche la seconda tesi è provata.

Mostriamo ora che  $p(B_{M'}) - p(B_M) \leq 3\beta$ . Sia  $e$  un arco che blocca  $M'$ . Esso deve essere incidente a un nodo che ha cambiato partner tra  $M$  ed  $M'$ . Se  $e$  è incidente a  $X_i \cup Y_i$ , allora  $e \in \{x_{6i+3}c(x_{6i+3}), x_{6i+4}c(x_{6i+4}), x_{6i+6}y_{6i+1}\}$ . Proviamo ora che  $e$  deve essere incidente a  $X_i \cup Y_i$ , il che dimostra la relazione introdotta. Supponiamo per assurdo non sia così, allora  $e \in \{p_j^a c_j^a, q_j c_j^a\} \cup \delta(q_j) \cup \delta(M(q_j))$ . Se  $e = p_j^a c_j^a$ , allora  $e$  blocca anche  $M$ , dato che  $p_j^a$  non ha cambiato partner. Per costruzione  $e \neq q_j c_j^a$ , Se  $e$  fosse incidente a  $q_j$  ma non a  $c_j^a$ , allora bloccherebbe  $M$ . Per ultima cosa, osserviamo che  $e$  non può essere incidente a  $M(q_j)$ , poichè  $q_j$  è il partner meno gradito da  $M(q_j)$ .

In sintesi, si ottiene  $f(M') - f(M) = p(M') - p(M) - p(B_{M'}) + p(B_M) \geq \alpha - \beta - 3\beta > 0$ , come richiesto.  $\square$

**Lemma 32.** *Un matching ottimo per  $I$  contiene esattamente  $5n + 2k$  archi di peso  $\alpha$ .*

*Dimostrazione.* Osserviamo che un matching può contenere fino a 5 archi di peso  $\alpha$  di  $X_i \times Y_i$ ,  $i \in [n]$ , poichè sia  $x_{6i+5}$  che  $x_{6i+6}$  formano un solo arco di peso  $\alpha$ , entrambi con  $y_{6i+6}$  come altro estremo: questo ci fornisce fino a  $5n$  archi di peso  $\alpha$  di  $X \times Y$ . Ogni altro arco di peso  $\alpha$  è incidente a un nodo di  $\{z_j, q_j\}_{j \in [k]}$ , questo produce fino a  $k$  archi di questo tipo che appartengono a  $Q \times C$  e altri  $k$  appartenenti a  $U \times Z$ . Consideriamo, ora, un matching ottimo  $M$  che contiene meno di  $5n + 2k$  archi di peso  $\alpha$ .

Se  $M$  contiene meno di 5 archi di peso  $\alpha$  di  $X_i \times Y_i$  per qualche  $i \in [n]$ , non può contenere archi di peso  $\beta$  di  $X_i \times C$  (Lemma 31). Definiamo il matching  $M' := M \setminus (X_i \times Y_i) \cup T_i$ . Allora  $p(M \cap (X_i \times Y_i)) \leq 4\alpha + 2\beta$ , mentre  $p(M' \cap T_i) = 5\alpha + \beta$ . Come nella dimostrazione del Lemma 31, esistono sono al più 3 archi di peso  $\beta$  che bloccano  $M'$  ma non  $M$ . Questo conduce alla contraddizione  $f(M') - f(M) \geq (5\alpha + \beta - 3\beta) - (4\alpha + 2\beta) = \alpha - 4\beta > 0$ .

Se  $M$  contiene meno di  $k$  archi di peso  $\alpha$  di  $Q \times C$ , allora esiste  $j \in [k]$  tale per cui  $q_j$  è single in  $M$ . Essendo  $M$  massimo dall'Osservazione 28, tutti i  $c_j^r$ ,  $r \in [3]$ , sono accoppiati in  $M$ . Poichè  $M$  non contiene archi di  $X \times C$  (Lemma 31), si ha  $M(c_j^r) = p_j^r$  per ogni  $r \in [3]$ . Il matching  $M' := M \setminus \{p_j^3 c_j^3\} \cup \{q_j c_j^3\}$  lascia  $p_j^3$  single, introducendo al più tre

archi che bloccano  $M'$  ma non  $M$ , ovvero  $p_j^3 w_j^3, p_j^3 c_j^3$  e  $x(c_j^3)c_j^3$ , tutti di peso  $\beta$ . Allora, si ottiene  $f(M') - f(M) \geq \alpha - 3\beta - \beta > 0$ , una contraddizione.

Se  $M$  contiene meno di  $k$  archi di peso  $\alpha$  di  $U \times Z$ ,  $z_j$  è single in  $M$  per qualche  $j \in [k]$ . Essendo  $M$  massimo per l'Osservazione 28, si ha  $M(u_j^r) = w_j^r$  per ogni  $r \in [3]$ . Il matching  $M' := M \setminus \{u_j^3 w_j^3\} \cup \{u_j^3 z_j\}$  lascia  $w_j^3$  single, introducendo così due archi che bloccano  $M'$  ma non  $M$ , che sono  $u_j^3 w_j^3$  e  $p_j^3 w_j^3$ , entrambi di peso  $\beta$ ; a questo punto otteniamo nuovamente  $f(M') - f(M) > 0$ .  $\square$

**Lemma 33.** *Esiste un matching di  $I$  che è perfetto e induce un matching perfetto in  $X_i \times Y_i \forall i \in [n]$ .*

La dimostrazione di questo lemma è simile a quella del Lemma 31, perciò verrà omessa. Arrivati a questo punto possiamo presentare la riduzione che dimostrerà il Teorema 27.

*Dimostrazione. (Teorema 27)* Sia  $f(I)$  il massimo valore di  $f(M)$  tra tutti i matching  $M$  di  $I$ . Affermiamo che  $f(I) = 5n\alpha + 3k\beta + 2k\alpha + t(B)\beta$ , con  $t(B)$  massimo valore di  $t(h)$  tra tutte le assegnazioni di verità  $h$  dell'istanza  $B$  del problema MAX (2, 2)-E3-SAT. Sia  $h$  un'assegnazione di verità di  $B$  tale che  $t(h) = t(B)$ . Creiamo un matching perfetto  $M$  in  $I$  come nella riduzione del problema (3, 3)-MMSBP[Biró et al., 2010]. Per ogni variabile  $v_i \in V$ , se  $v_i$  è vera per  $h$ , aggiungiamo gli archi di  $T_i$  a  $M$ , altrimenti aggiungiamo a  $M$  gli archi di  $F_i$ . In entrambi i casi, è immediato vedere che c'è esattamente un arco di  $X_i \times Y_i$  che blocca  $M$ , rispettivamente  $x_{6i+6}y_{6i+1}$  o  $x_{6i+5}y_{6i+5}$ . Pertanto si ha  $p(M \cap (X_i \times Y_i)) - p(B_M \cap (X_i \times Y_i)) = 5\alpha + \beta - \beta = 5\alpha$ .

Sia ora  $j \in [k]$ . Se  $c_j$  contiene un letterale vero per  $h$ , identifichiamo con  $s \in \{1, 2, 3\}$  la posizione di  $c_j$  la posizione occupata da tale letterale, in caso contrario poniamo  $s = 1$ . Aggiungiamo ad  $M$  gli archi  $p_j^t c_j^t, u_j^t w_j^t$  ( $1 \leq t \neq s \leq 3$ ),  $q_j c_j^s, p_j^s w_j^s$  e  $u_j^s z_j$ . Il peso totale di questi archi è  $2\alpha + 5\beta$ . Inoltre,  $u_j^s z_j$  è un arco bloccante con  $p(u_j^s z_j) = \beta$ . Ora, se  $c_j$  non è soddisfatta sotto  $h$ , allora, per costruzione di  $M$ , l'uomo  $x(c_j^1)c_j^1$  viene assegnato all'ultimo partner nella sua lista di preferenze. Quindi anche  $x(c_j^1)c_j^1$  è un arco bloccante con  $p(x(c_j^1)c_j^1) = \beta$ . Sapendo che  $M$  è perfetto, usando l'Osservazione 29, si può dedurre che, insieme agli  $n + k$  archi bloccanti di  $X \times Y$  e  $U \times W$  già identificati, questi formano l'insieme di tutti gli archi bloccanti di  $M$  in  $I$ . Perciò  $f(I) \geq f(M) = n(5\alpha) + k(2\alpha + 5\beta - \beta) - (k - t(B))\beta = 5n\alpha + 2k\alpha + 3k\beta + t(B)\beta$ .

Viceversa, sia  $M$  un matching ottimo per  $I$ . Dal Lemma 33 possiamo assumere che  $M$  è perfetto e che induce un matching perfetto di  $X_i \times Y_i$ , per ogni  $i \in [n]$ . Dall'Osservazione 30, questo matching può essere  $T_i$  o  $F_i$ . Inoltre, dal momento che non ci sono agenti single in  $M$ , dall'Osservazione 29 gli archi che bloccano  $M$  sono in  $X \times C$  o in  $U \times W$  oppure sono della forma  $x_{6i+5}y_{6i+5}$  e  $x_{6i+6}y_{6i+1}$  e sono tutti di peso  $\beta$ .

Costruiamo un'assegnazione di verità  $h$  in  $B$  attraverso  $M$  nel modo seguente. per ogni  $i \in [n]$ , se  $M \cap (X_i \times Y_i) = T_i$ , poniamo  $v_i$  vera per  $h$ ; altrimenti, se  $M \cap (X_i \times Y_i) = F_i$ , poniamo  $v_i$  falsa. Inoltre, sappiamo che  $|B_M| = n + k + (k - t(h))$  ( $n$  archi di  $X \times Y$ ,  $k$  di  $U \times W$  e  $k - t(h)$  di  $X \times C$ ), perciò  $f(M) \leq 5n\alpha + n\beta + 2k\alpha + 5k\beta - (n + k + (k - t(h)))\beta = 5n\alpha + 2k\alpha + 3k\beta + t(B)\beta$ . Vista la disuguaglianza opposta precedentemente dimostrata, segue che  $f(M) \leq 5n\alpha + 2k\alpha + 3k\beta + t(B)\beta$ . Quindi, una soluzione ottima di  $I$  viene trovata se e solo se MAX (2, 2)-E3-SAT viene risolto in maniera ottima.  $\square$

La difficoltà nell'approssimare il problema MAX (2, 2)-E3-SAT trattata in [Berman et al., 2004], combinata con i risultati precedenti, ci conduce al seguente corollario, la cui dimostrazione segue da [Biró et al., 2010].

**Corollario 34.** *Per ogni coppia di costanti  $\alpha, \beta$  con  $\alpha > 4\beta > 0$ , il problema  $\beta$ -BUM è APX-difficile.*

A questo punto è naturale cercare un algoritmo il cui coefficiente di approssimazione dipenda da  $\alpha$  e  $\beta$ . Un'idea pertinente è quella di esaminare la vicinanza tra un matching stabile di peso massimo (MWSM) e una soluzione ottima.

**Lemma 35.** *Se  $M'$  è una soluzione ottima per un'istanza del problema  $\beta$ -BUM dove l'immagine di  $p$  è  $\{\alpha, \beta\}$ , e  $M$  è un MWSM, allora  $\frac{f(M')}{f(M)} \leq 2\frac{\alpha}{\beta} - 1$ .*

*Dimostrazione.* Sia  $|M| = k$ . Ogni arco di  $M$  ha peso almeno  $\beta$  quindi  $f(M) \geq k\beta$ . E' noto che ogni matching stabile è massimale e ogni matching massimale ha cardinalità almeno metà di quella di un matching massimo. Per dimostrare la seconda affermazione si considera un matching massimo  $M^*$  e si assume per assurdo che esista un matching massimale  $M_a$  con  $|M_a| < (1/2)|M^*|$ . Vogliamo mostrare che possiamo aumentare il matching  $M_a$  aggiungendo un arco di  $M^*$ , contraddicendo la massimalità di  $M_a$ . Notiamo che per ogni  $(u, v) \in M_a$  ci sono al più due archi in  $M^*$  incidenti a  $u$  o a  $v$  (uno per ogni estremo). Pertanto, ci sono al più  $2|M_a|$  archi di  $M^*$  incidenti in vertici di  $M_a$ . Dal momento che  $2|M_a| < |M^*|$ , si ha che esiste un arco  $e \in M^*$  che non incide in nessun vertice di  $M_a$ . Quindi  $M_a \cup \{e\}$  è un matching, e  $M_a$  non è massimale, assurdo. Perciò,  $M'$  soddisfa  $|M'| \leq 2k$  poichè tale matching non può avere più archi di un matching massimo. Dal Lemma 24 sappiamo che almeno  $|M'| - k$  archi bloccano  $M'$  ( $|B_{M'}^M| \geq |M'| - |M|$ ). Osserviamo che  $f(M')$  è massimizzata se ogni arco appartenente a  $M'$  è di peso  $\alpha$  ed ogni arco bloccante è di peso  $\beta$ . Questo implica che

$$\begin{aligned} f(M') &\leq |M'|\alpha - (|M'| - k)\beta = |M'|(\alpha - \beta) + k\beta \\ &\leq 2k(\alpha - \beta) + k\beta = 2k\alpha - k\beta. \end{aligned}$$

Segue che  $\frac{f(M')}{f(M)} \leq \frac{2k\alpha - k\beta}{k\beta} = 2\frac{\alpha}{\beta} - 1$ . □

Presentiamo ora un esempio che ci mostra la precisione di questo coefficiente di approssimazione.

**Esempio 1.** *Siano  $M = \{m_1, m_2\}$  e  $W = \{w_1, w_2\}$ , dove  $m_1$  preferisce  $w_2$  a  $w_1$ ,  $w_2$  preferisce  $m_1$  a  $m_2$ , mentre  $m_2$  e  $w_1$  hanno rispettivamente solo  $w_2$  e  $m_1$  nella loro lista di preferenza. Siano  $m_1w_1$  e  $m_2w_2$  di peso  $\alpha$  e  $m_1w_2$  di peso  $\beta$ . L'unico matching stabile è  $M = \{m_1w_2\}$  ed il suo valore è  $f(M) = \beta$ . La soluzione ottima di  $\beta$ -BUM è  $M' = \{m_1w_1, m_2w_2\}$  che ha valore  $f(M') = 2\alpha - \beta$  essendo  $m_1w_2$  un arco bloccante. Pertanto  $\frac{f(M')}{f(M)} = 2\frac{\alpha}{\beta} - 1$ .*



### 3.4 Il problema Red-Blue Unstable Matching (R-B UM)

L'obiettivo del problema Red-Blue Unstable Matching è quello di trovare un matching  $M$  che sia stabile nel sottografo indotto dai soli vertici blu  $E \setminus F$  e massimizzi la somma dei pesi in  $M$  meno la somma dei pesi degli archi rossi  $F$  che bloccano  $M$ . Useremo concetti classici e risultati relativi alle rotazioni e agli insiemi parzialmente ordinati (poset) associati, rimandiamo il lettore a [Gusfield and Irving, 1989] per una presentazione più esaustiva. Consideriamo un'istanza  $I$  del problema R-B UM e definiamo  $G' := (V, E \setminus F)$ ,  $R$  l'insieme delle rotazioni di  $(G', >)$ , e  $M_0$  il matching stabile ottimo per gli uomini. Una rotazione  $\rho$  è un insieme ordinato di archi  $(m_0w_0, \dots, m_{r-1}w_{r-1})$  che, se identificato con un matching stabile  $M$  e poi eliminato ruotando in senso orario gli uomini in  $\rho$ , restituisce un altro matching stabile che denotiamo con  $M\Delta\rho := M \setminus \{m_iw_i : i \in [r-1]_0\} \cup \{m_iw_{i+1} : i \in [r-1]_0\}$ , con gli indici presi modulo  $r$ . Per ogni matching stabile  $M$ , esiste esattamente un insieme di rotazioni  $R' = \{\rho_1, \dots, \rho_j\} \subseteq R$  la cui successione di eliminazioni da un matching ottimo per uomini restituisce  $M$ , ovvero  $M = M_0\Delta R' := ((M_0\Delta\rho_1) \dots \Delta\rho_k)$ . L'insieme parzialmente ordinato delle rotazioni  $(R, \succeq)$  è definito nel modo seguente: per  $\rho, \rho' \in R$ , abbiamo che  $\rho \succeq \rho'$  se e solo se, per ogni  $R'$  tale che  $M_0\Delta R'$  è un matching stabile e  $\rho' \in R'$ , si ha  $\rho \in R'$ . Segue il teorema.

**Teorema 36.** ([Irving and Leather, 1986]Thm 4.1) *Esiste una biezione tra l'insieme dei matching stabili di  $(G', >)$  e gli insiemi chiusi di  $(R, \succeq)$ , che mappa ogni insieme chiuso  $R'$  di  $(R, \succeq)$  in un matching stabile  $M_0\Delta R'$ .*

**Definizione.** Un insieme chiuso in un poset  $(S, \geq)$  è un sottoinsieme  $S' \subseteq S$  chiuso sotto la relazione d'ordine, ovvero tale che se  $x \in S'$  è in relazione con  $y$ , allora anche  $y \in S'$ .

Ora estendiamo i pesi alle rotazioni [Irving et al., 1987]. Per  $\rho = (m_0w_0, \dots, m_{r-1}w_{r-1})$ , definiamo  $p(\rho) := \sum_{i=0}^{r-1} p(m_iw_{i+1}) - p(m_iw_i)$ . Allora, se  $M = M_0\Delta R'$ , abbiamo  $p(M) = p(M_0) + p(R')$ . Siano  $R^+ := \{\rho \in R : p(\rho) \geq 0\}$  e  $R^- := R \setminus R^+$ . Ricordiamo che eliminando una rotazione  $\rho$  identificata a  $M$ , ogni uomo (risp. donna) ottiene in  $M\Delta\rho$  un partner peggiore (risp. migliore), oppure non cambia partner. Possiamo allora associare determinate rotazioni a ciascun arco rosso.

**OSSERVAZIONE 37.** Sia  $m \in M$ ,  $w \in W$  ed  $e = mw \in F$ .

- Esiste al massimo una  $\rho^+(e) = (m_0w_0, \dots, m_{r-1}w_{r-1}) \in R$  tale che  $w = w_i$  e  $m_{i-1} >_w m >_w m_i$  per qualche  $i \in [r-1]_0$ .
- Non esiste una  $\rho = (m_0w_0, \dots, m_{r-1}w_{r-1}) \in R$  tale che  $w = w_i$  e  $m_i >_w m >_w m_{i+1}$  per qualche  $i \in [r-1]_0$ .
- Esiste al massimo una  $\rho^-(e) = (m_0w_0, \dots, m_{r-1}w_{r-1}) \in R$  tale che  $m = m_i$  e  $w_i >_m w >_m w_{i+1}$  per qualche  $i \in [r-1]_0$ .
- Non esiste una  $\rho = (m_0w_0, \dots, m_{r-1}w_{r-1}) \in R$  tale che  $m = m_i$  e  $w_{i+1} >_m m >_m w_i$  per qualche  $i \in [r-1]_0$ .

Introduciamo ora una nuova classificazione degli archi. Attribuiamo a  $mw \in F$  con  $m \in M$  e  $w \in W$  una classificazione data da una coppia ordinata  $(l_m, l_w)$ , definita nel modo seguente:

- Se  $M_0(m) >_m w$  allora poniamo  $l_m = -$ , altrimenti  $l_m = +$ ;
- Se  $M_0(w) >_w m$ , allora poniamo  $l_w = -$ , altrimenti  $l_w = +$ .

Scriveremo, per esempio,  $e = (+, +)$  (oppure diremo  $e$  è  $(+, +)$ ) per fare riferimento alla sua classificazione.

A questo punto abbiamo tutti gli strumenti necessari per abbozzare un'idea su come costruire il nostro algoritmo.  $M_0$  è bloccato solo dagli archi  $(+, +)$ . Ricordiamo che, quando eliminiamo una rotazione esposta (identificata), gli uomini non migliorano la loro partner e le donne non lo peggiorano. Perciò, eliminando iterativamente le rotazioni esposte a partire da  $M_0$ , alcuni archi  $(+, +)$  verranno neutralizzati, ovvero diventeranno  $(+, -)$  non bloccando più il matching attuale, mentre certi archi  $(-, +)$  si attiveranno, trasformandosi in  $(+, +)$  e quindi bloccando il matching attuale. Pertanto, dobbiamo trovare un matching stabile  $M$  in  $(G', >)$  - o equivalentemente un insieme chiuso di rotazioni di  $(R, \succeq)$  - tale che  $p(M)$  meno la somma dei  $p(e)$  per tutti gli archi  $e$  "attivi" più la somma dei  $p(e)$  per tutti gli archi  $e$  "deattivati" (neutralizzati) è massimizzato. Calcoliamo questo matching risolvendo un problema di taglio minimo per un grafo orientato.

Ora applichiamo queste idee per partizionare  $F$ .

- $F_0$  è l'insieme degli archi  $e \in F$  con  $e = (+, +)$  e tali che  $\rho^+(e)$  non esiste.
- $F_1$  è l'insieme degli archi  $e \in F$  con  $e = (+, +)$  e tali che  $\rho^+(e)$  esiste.
- $F_2$  è l'insieme degli archi  $e \in F$  con  $e = (-, +)$  e tali che sia  $\rho^+(e)$  che  $\rho^-(e)$  esistono.
- $F_3$  è l'insieme degli archi  $e \in F$  con  $e = (-, +)$  e tali che  $\rho^+(e)$  non esiste, mentre  $\rho^-(e)$  esiste.
- $F_4$  è l'insieme  $F \setminus \bigcup_{i=0}^3 F_i$ .

Mostriamo in maniera più chiara le caratteristiche degli elementi delle partizioni descritte sopra,  $e \in F_0$  è  $(+, +)$  e non viene mai "deattivato",  $e \in F_1$  è  $(+, +)$  e viene "deattivato" se e solo se eliminiamo  $\rho^+(e)$ ,  $e \in F_2$  (resp.  $F_3$ ) è  $(-, +)$  e viene "attivato" se e solo se eliminiamo  $\rho^-(e)$  ma non  $\rho^+(e)$  (resp., eliminiamo  $\rho^-(e)$ ), mentre gli archi di  $F_4$  non si attivano mai.

Formalizziamo quanto detto nel seguente lemma.

**Lemma 38.** *Sia  $M$  un matching stabile di  $(G', >)$  e poniamo  $M = M_0 \triangle R'$  per un qualche  $R' \subseteq R$ . Allora:*

- 0. *Tutti gli archi di  $F_0$  bloccano  $M$ ;*
- 1. *Un arco  $e \in F_1$  blocca  $M$  se e solo se  $\rho^+(e) \neq R'$ ;*
- 2. *Un arco  $e \in F_2$  blocca  $M$  se e solo se  $\{\rho^+(e), \rho^-(e)\} \cap R' = \{\rho^-(e)\}$ ;*

- 3. Un arco  $e \in F_3$  blocca  $M$  se e solo se  $\rho^-(e) \in R'$ ;
- 4. Nessun arco di  $F_4$  blocca  $M$ .

*Dimostrazione.* 0. Sia  $e = mw \in F_0$ . Per definizione di  $F_0$  si ha  $w \succ_m M_0(m)$  e  $m \succ_w M_0(w)$ . Dalla definizione di  $M_0$ , invece, segue  $M_0(m) \succeq_m M(m)$ . D'altro canto, dal momento che  $\rho^+(e)$  non esiste, abbiamo che  $m \succ_w M(w)$ , pertanto  $e$  blocca  $M$ .

1. Se  $e = mw \in F_1$  come sopra,  $w \succ_m M_0(m)$  e  $m \succ_w M_0(w)$  e dalla definizione di  $M_0$  segue  $M_0(m) \succeq_m M(m)$ . Se  $\rho^+(e) \neq R'$  (non vale  $m_{i-1} \succ_w m \succ_w m_i$  per  $w = w_i$ ), allora  $w$  non trova un partner migliore di  $m$  in  $M$  ( $m \succ_w M(w)$ ), quindi  $e$  blocca  $M$ .

2. Se  $e = mw \in F_2$ , per definizione abbiamo  $M_0(m) \succ_m w$  e  $m \succ_w M_0$ . Pertanto  $e$  blocca  $M$  se e solo se dopo aver eliminato  $R'$ ,  $m$  è accoppiato con una donna peggiore di  $w$  e  $w$  non è accoppiata con un uomo migliore di  $m$ , ovvero se  $\{\rho^+(e), \rho^-(e)\} \cap R' = \{\rho^-(e)\}$ .

3. Uguale al punto 2, col vantaggio che non ci dobbiamo preoccupare dell'esistenza di  $\rho^+(e)$ .

4. Per costruzione di  $F_4$ , un arco  $e \in F_4$  non si può mai attivare.  $\square$

Costruiamo ora un multigrafo direzionale con capacità  $D(T, K)$  con  $T = R \cup \{s, t\}$  e  $K$  unione disgiunta degli insiemi:

- $K_0 := \{(\rho, \rho')\}_{\rho, \rho' \in R, \rho \succ \rho'}$ , ognuno con capacità  $+\infty$ ;
- $K_1 := \{(\rho^+(e), t)\}_{e \in F_1}$ , ognuno con capacità  $p(e)$ ;
- $K_2 := \{(\rho^+(e), \rho^-(e))\}_{e \in F_2}$ , ognuno con capacità  $p(e)$ ;
- $K_3 := \{(s, \rho^-(e))\}_{e \in F_3}$ , ognuno con capacità  $p(e)$ ;
- $K_{R^+} := \{(\rho, t)\}_{\rho \in R^+}$ , ognuno con capacità  $p(\rho)$ ;
- $K_{R^-} := \{(s, \rho)\}_{\rho \in R^-}$ , ognuno con capacità  $-p(\rho)$ .

Sia  $w$  il vettore delle capacità definite sopra. Si noti che  $w \geq 0$  e in ognuno degli insiemi  $K_1, K_2, K_3$  possono essere presenti più copie dello stesso arco, si veda la Figura.

Ora presenteremo un algoritmo polinomiale e, col prossimo lemma, mostreremo che esso risolve il problema Red-Blue Unstable Matching. Per  $S \subseteq T$ , ricordiamo che  $\delta^+(S)$  rappresenta l'insieme degli archi di  $D$  uscenti dai nodi in  $S$ .

**Lemma 39.** *Sia  $S$  un taglio  $s - t$  di  $D(T, K)$  di capacità finita  $w$ . Allora  $R \setminus S$  è un insieme chiuso di  $(R, \succeq)$ . Inoltre,  $w(\delta^+(S)) = p(M_0) - p(F_0) + p(R^+) - f(M)$  dove  $M := M_0 \Delta (R \setminus S)$ . Viceversa, per ogni insieme chiuso  $R'$  di  $(R, \succeq)$ ,  $S := \{s\} \cup (R \setminus R')$  è un taglio  $s - t$ .*

*Dimostrazione.* Sia  $S$  un taglio  $s - t$  di  $D(T, K)$  con capacità finita, e sia  $\rho \succ \rho' \in R$ . Poichè l'arco  $(\rho, \rho')$  ha capacità infinita, abbiamo che  $\rho' \in (R \setminus S)$  implica  $\rho \in (R \setminus S)$ , il che dimostra  $R \setminus S$  chiuso. Perciò,  $M = M_0 \Delta (R \setminus S)$  è un matching stabile (vedi Teorema 36). Viceversa, dato che gli unici archi di capacità infinita sono  $(\rho, \rho')$  con  $\rho \succ \rho' \in R$ , ne deduciamo che, se  $R'$  è chiuso, allora  $\{s\} \cup (R \setminus R')$  è un taglio  $s - t$  di capacità finita. Dal

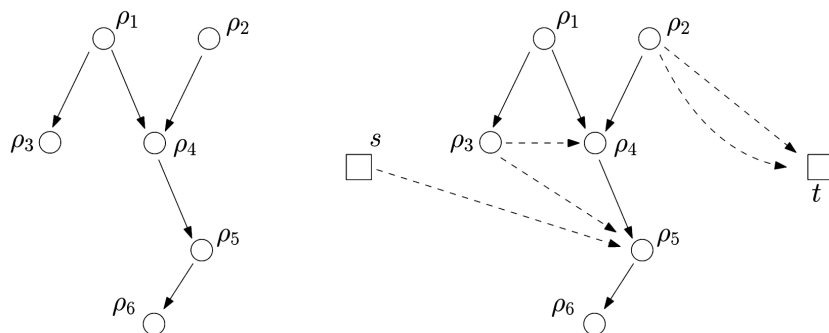


Figura 3.3: Sulla sinistra: Una rotazione. Sulla destra: Un esempio del multigrafo  $D(U, K)$  (gli archi di  $K_{R^+}, K_{R^-}$  e quelli dovuti alla transitività di  $\succeq$  sono stati omessi). Gli archi pieni hanno capacità infinita, mentre quelli tratteggiati hanno capacità  $p(e)$  per qualche  $e \in E$ . In questo caso  $F_1 = \{e_1, e_2\}$  con  $\rho^+(e_1) = \rho^+(e_2) = \rho_2$ ;  $F_2 = \{e_3, e_4\}$  con  $\rho^+(e_3) = \rho^+(e_4) = \rho_3, \rho^-(e_3) = \rho_4, \rho^-(e_4) = \rho_5$ ;  $F_3 = \{e_5\} = \rho_5$ . Il taglio  $\{s, \rho_3, \rho_5, \rho_6\}$  di capacità 1 corrisponde all'insieme chiuso  $\{\rho_1, \rho_2, \rho_4\}$ .

---

**Algorithm 6:**


---

**Data:** un'istanza  $I$  del problema Red-Blue Unstable Matching: un sistema di preferenze  $(G(V, E), >)$ , una funzione  $p: E \rightarrow \mathbb{Z}_{\geq 0}$  e un insieme  $F \subseteq E$ .

**Restituisc:** una soluzione ottima per  $I$ .

- 1: Calcolare il matching stabile ottimo per gli uomini  $M_0$  da  $(G(V, E \setminus F), >)$ , l'insieme  $R$  delle rotazioni, e il suo insieme parzialmente ordinato corrispondente  $(R, \succeq)$ , vedi [Gusfield and Irving, 1989].
  - 2: Classificare gli archi di  $F$ , costruire la partizione  $(F_0, \dots, F_4)$  di  $F$  e il digrafo  $D(T, K)$  con le capacità degli archi  $w$  definite in precedenza.
  - 3: Trovare  $S^*$  taglio minimo  $s - t$  in  $D(T, K)$ .
  - 4: **ritorna**  $M = M_0 \Delta (R \setminus S^*)$ .
- 

Teorema 36, questo definisce una biezione tra i tagli  $s - t$  di capacità finita  $S$  in  $D(T, K)$  e i matching stabili di  $(G, >)$ . Per un taglio  $s - t$  di capacità finita  $S$ , abbiamo:

$$\begin{aligned}
 w(\delta^+(S)) &= \underbrace{p(e \in F_1 : \rho^+(e) \in S)}_{(1)} + \underbrace{p(e \in F_2 : \rho^+(e) \in S, \rho^-(e) \in R \setminus S)}_{(2)} + \underbrace{p(e \in F_3 : \rho^-(e) \in R \setminus S)}_{(3)} \\
 &\quad + p(R^+ \cap S) - p(R^- \setminus S).
 \end{aligned}$$

Sia  $M_S$  il matching associato a  $S$  dalla biezione definita in precedenza. Dal Lemma 38, un arco  $e \in F$  blocca  $M_S$  se e solo se esso appartiene a  $F_0 \cup (1) \cup (2) \cup (3)$ . Ne deduciamo che:

$$\begin{aligned}
 f(M) &= p(M) - p(B_M) \\
 &= p(M_0) + p(R \setminus S) - w(\delta^+(S)) + p(R^+ \cap S) - p(R^- \cap S) - p(F_0) \\
 &= p(M_0) - w(\delta^+(S)) - p(F_0) + \underbrace{p(R^+ \cap S) + p(R \setminus S) - p(R^- \setminus S)}_{(*)},
 \end{aligned}$$

dove la seconda uguaglianza segue dall'argomentazione precedente, con la riscrittura di  $w(\delta^+(S))$  e da  $p(M) = p(M_0) + p(R \setminus S)$ . Osservando che  $(*) = p(R^+ \cap S) + p(R^+ \setminus S) = p(R^+)$  e riordinando gli addendi, si conclude la dimostrazione.  $\square$

Nel caso volessimo trattare il problema R-B UM senza assumere la non negatività di  $p$ , è necessario apportare delle modifiche alla costruzione. Infatti, dovremmo sostituire ognuno dei  $K_1, K_2, K_3$  con due insiemi, in base al segno di  $p(e)$ , ed infine modificare in maniera appropriata l'Algoritmo 6 e il Lemma 39.



# Bibliografia

- [Arcaute and Vassilvitskii, 2009] Arcaute, E. and Vassilvitskii, S. (2009). Social networks and stable matchings in the job market. In *International Workshop on Internet and Network Economics*, pages 220–231. Springer.
- [Askalidis et al., 2013] Askalidis, G., Immorlica, N., Kwanashie, A., Manlove, D. F., and Pountourakis, E. (2013). Socially stable matchings in the hospitals/residents problem. In *Algorithms and Data Structures: 13th International Symposium, WADS 2013, London, ON, Canada, August 12-14, 2013. Proceedings 13*, pages 85–96. Springer.
- [Berman et al., 2004] Berman, P., Karpinski, M., and Scott, A. (2004). Approximation hardness of short symmetric instances of max-3sat. Technical report.
- [Biró et al., 2010] Biró, P., Manlove, D. F., and Mittal, S. (2010). Size versus stability in the marriage problem. *Theoretical Computer Science*, 411(16-18):1828–1841.
- [Cheng and McDermid, 2013] Cheng, C. T. and McDermid, E. (2013). Maximum locally stable matchings. *Algorithms*, 6(3):383–395.
- [Downey and Fellows, 1995] Downey, R. G. and Fellows, M. R. (1995). Fixed-parameter tractability and completeness ii: On completeness for w [1]. *Theoretical Computer Science*, 141(1-2):109–131.
- [Duan and Su, 2012] Duan, R. and Su, H.-H. (2012). A scaling algorithm for maximum weight matching in bipartite graphs. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1413–1424. SIAM.
- [Faenza et al., 2021] Faenza, Y., Mourtos, I., Samaris, M., and Sethuraman, J. (2021). (un) stable matchings with blocking costs. *Operations Research Letters*, 49(5):655–662.
- [Gabow and Tarjan, 1989] Gabow, H. N. and Tarjan, R. E. (1989). Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18(5):1013–1036.
- [Gale and Shapley, 1962] Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15.
- [Gusfield and Irving, 1989] Gusfield, D. and Irving, R. W. (1989). *The stable marriage problem: structure and algorithms*. MIT press.

- [Hamada et al., 2009] Hamada, K., Iwama, K., and Miyazaki, S. (2009). An improved approximation lower bound for finding almost stable maximum matchings. *Information Processing Letters*, 109(18):1036–1040.
- [Hopcroft and Karp, 1973] Hopcroft, J. E. and Karp, R. M. (1973). An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231.
- [Huang and Kavitha, 2013] Huang, C.-C. and Kavitha, T. (2013). Popular matchings in the stable marriage problem. *Information and Computation*, 222:180–194.
- [Irving and Leather, 1986] Irving, R. W. and Leather, P. (1986). The complexity of counting stable marriages. *SIAM Journal on Computing*, 15(3):655–667.
- [Irving et al., 1987] Irving, R. W., Leather, P., and Gusfield, D. (1987). An efficient algorithm for the “optimal” stable marriage. *Journal of the ACM (JACM)*, 34(3):532–543.
- [Király, 2013] Király, Z. (2013). Linear time local approximation algorithm for maximum stable marriage. *Algorithms*, 6(3):471–484.
- [McDermid, 2009] McDermid, E. (2009). A  $3/2$ -approximation algorithm for general stable marriage. In *International Colloquium on Automata, Languages, and Programming*, pages 689–700. Springer.
- [Sotomayor, 1990] Sotomayor, M. A. O. (1990). *Two-sided matching: study in game-theoretic modeling and analysis*. Cambridge University Press.