Università degli Studi Roma Tre
Facoltà di Scienze M.F.N.

Sintesi della Tesi di Laurea in Matematica

di

Letizia Monaldi

Matching su grafi e alcune varianti del problema del matrimonio stabile

Relatore

Prof. Marco Liverani

Il Candidato

Il Relatore

ANNO ACCADEMICO 2003-2004
Maggio 2005

Classificazione AMS: 05D15, 05C85, 05C90, 90C27, 68R10.

Parole chiave: grafi, matching, stable matching, stable marriage problem.

Sintesi

Un *matching* per un grafo G è un sottoinsieme di archi a due a due non incidenti, cioè non aventi estremi in comune. In particolare è un insieme di coppie distinte di nodi in relazione tra loro, relazione rappresentata dall'arco che li congiunge.

In questo lavoro ci occupiamo di matching su grafi bipartiti, cioè su grafi per i quali si può suddividere l'insieme dei nodi in due insiemi disgiunti A e B tali che tutti gli archi del grafo abbiano un estremo in A e in B.

Una proprietà caratteristica di questo tipo di grafi è l'inesistenza di cicli di lunghezza dispari. Infatti la presenza di un ciclo dispari in un grafo bipartito implicherebbe l'esistenza di un arco che congiunge due vertici dello stesso insieme (A o B), viceversa se un grafo non ha cicli dispari si può costruire una partizione dell'insieme dei vertici ponendo da un lato quelli che hanno ordine pari in un cammino del grafo e dall'altro quelli che hanno invece ordine dispari.

Un matching su un grafo bipartito G si dice *completo* da A a B o viceversa se abbina tutti i nodi di A o di B rispettivamente. Se poi $|A| = |B|$ allora un matching completo si dice *perfetto* e G si dice *fattorizzabile*.

Se G è fattorizzabile allora necessariamente $|A| = |B|$ ed ogni $S \subset A$ è tale che $|S| \leq |N(S)|$[1]. Per il Teorema di Hall tale condizione (*marriage condition*), evidentemente necessaria per la fattorizzabilità di G, è anche

[1] $N(S) \subset B$ è l'insieme degli adiacenti ai nodi in S.

sufficiente.

Un matching è *massimo* se ha cardinalità massima tra tutti i matching del grafo, dunque se il grafo è fattorizzabile allora un matching massimo è un matching perfetto; ma se G non è fattorizzabile e dunque non soddisfa la *marriage condition* allora esiste un insieme $S \subset A$, detto *insufficiente* tale che $|S| > |N(S)|$. Si definisce allora *deficiency* di S la quantità $\delta(S) := |S| - |N(S)|$. La deficiency massima tra tutti i sottoinsiemi di A definisce invece la *deficiency* di G ($\delta(G)$) e gli insiemi insufficienti aventi deficiency massima si dicono *tight*. König e Ore nel 1955 dimostrarono che $|A| - \delta(G)$ è la cardinalità massima possibile dei matching su un tale grafo G.

Possiamo considerare i problemi di matching come particolari problemi di ottimizzazione che richiedono la suddivisione in coppie distinte di elementi, legati da qualche relazione, rispettando dei vincoli ben determinati; sono tra i più studiati problemi di ottimizzazione combinatoria ed hanno molteplici applicazioni, sia come modello di problemi reali sia come strumento da utilizzare nella risoluzione di problemi di ottimizzazione più complessi.

In questo lavoro suddividiamo i problemi di matching in tre classi:

1. problema di accoppiamento di massima cardinalità, noto in letteratura come problema del "matching massimo";

2. problema del "matching di peso massimo/minimo";

3. problema del "matching stabile".

Gli algoritmi che risolvono il problema del matching massimo si avvalgono delle proprietà di particolari cammini del grafo: i cammini aumentanti rispetto ad un certo matching. Dato un matching M, un cammino M-*aumentante* è un cammino M-alternante, cioè formato da un'alternanza di archi di M e non, che inizia e termina con nodi non abbinati in M (Figura 1). Se M è un matching e P l'insieme degli archi di un cammino M-aumentante allora

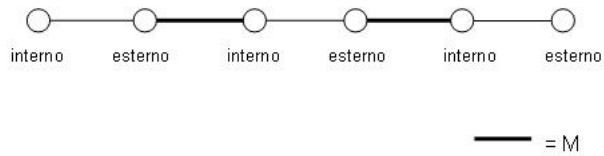

Figura 1: Cammino $M-$aumentante

$M' := P \oplus M$ [2] è un matching di cardinalità $|M| + 1$, ottenuto da M per *augmentation*.

Dunque se esiste un cammino M-aumentante allora M non può essere massimo. Meno evidente è il fatto che se non esistono cammini M-aumentanti allora M è massimo. È ciò che provarono Berge (1957) e Norman Rabin (1959).

Questi risultati suggeriscono il seguente schema algoritmico per la ricerca di un matching massimo:

1. partire da un matching M vuoto o contenente un solo arco;
2. cercare un cammino M-aumentante;
3. se un tale cammino esiste costruire il matching M' per *augmentation* e tornare al passo 2 ponendo $M := M'$;
4. se non esistono cammini aumentanti allora M è massimo.

La fase non banale di una procedura di questo tipo è la modalità di ricerca di un cammino aumentante. L'algoritmo che descriviamo, per realizzare questa ricerca, costruisce un grafo ausiliario orientato $G' = (B, E')$ tale che $(b_i, b_j) \in E'$ se e soltanto se b_i è un nodo di ordine dispari (o esterno) che precede b_j, anch'esso esterno, in un cammino alternante, ed opera

[2]Dati due insiemi A e B si definisce *differenza simmetrica* tra A e B l'insieme $A \oplus B := (A - B) \cup (B - A)$.

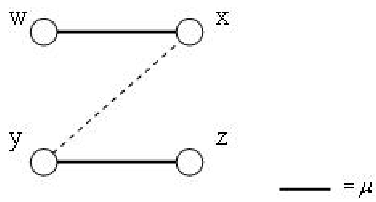

Figura 2: (x, y) è una coppia che blocca la stabilità di μ se x preferisce y a w e y preferisce x a z.

su di esso una visita in ampiezza. Attraverso questa esplorazione ricorsiva si individuano i possibili cammini aumentanti in tempo $O(|E|)$. Il tempo di esecuzione totale necessario per ottenere un matching massimo del grafo bipartito $G = ((A, B), E)$ è dell'ordine $O(\min(|A|, |B|) \cdot |E|)$.

Per quanto riguarda il problema del matching di peso massimo o minimo ne illustriamo la formulazione come problema di programmazione lineare intera. Innanzitutto è il problema in cui associando ad ogni arco del grafo un peso, cioè un numero intero positivo, e definendo il peso del matching come la somma dei pesi degli archi che lo compongono, si ricerca tra tutti i matching del grafo quello che ha un peso massimo o minimo. Questo problema, come in generale tutti i problemi di matching, formulato come problema di programmazione linare intera può sfruttare gli strumenti della programmazione lineare.

Vero e proprio oggetto di questa trattazione è il problema del matching stabile per grafi bipartiti.

Fissando per ogni nodo x del grafo bipartito una relazione d'ordine lineare $(<_x)$ su $N(x)$, che esprima una preferenza tra le possibili coppie di elementi, si può ricercare un matching massimo che sia stabile rispetto a questa relazione. In un matching stabile, seppure un elemento y preferisca essere abbinato a x, x non ricambia questa preferenza, cioè y non può cercare di migliorare la propria situazione destabilizzando il sistema di coppie. Riassumendo, un matching μ è stabile se non esiste una coppia di elementi $(x, y) \notin \mu$ tali che si preferiscano entrambi agli elementi cui sono rispettivamente abbinati in μ, in tal caso si dice che non esiste una *coppia blocco* per μ (Fig. 2).

Gale e Shapley presentarono questo problema nel 1962 definendolo *stable marriage problem* cioè ponendolo in termini di matrimoni stabili tra uomini e donne. Da questo punto di vista gli elementi da abbinare sono uomini e donne, le preferenze, espresse dai singoli componenti di questo gruppo, riflettono la volontà di sposare una piuttosto che un'altra persona e l'abbinamento stabile in questione costituisce un insieme di "matrimoni stabili".

Questa chiave di lettura si è mantenuta nel tempo e tuttora nell'affrontare lo studio di varianti di questo problema si continua a parlare di matrimoni stabili tra uomini e donne per far riferimento a matching stabili su grafi bipartiti.

Gli stessi Gale e Shapley realizzarono una dimostrazione costruttiva dell'esistenza di matching stabili. Consideriamo (M, W) la partizione del grafo bipartito, ed esprimiamo le relazioni di preferenza ponendo $m >_w m'$ se e soltanto se w preferisce m a m' (ossia m precede m' nella lista di preferenza di w); secondo la procedura di Gale e Shapley (GS) inizialmente si tentano le coppie (m, w) per ogni $m \in M$ e w al primo posto nella lista di preferenza di m (diremo che ogni m si "propone" a questi elementi). Gli elementi di W che sono abbinati a più di un elemento "scelgono" quello che preferiscono. Successivamente ogni elemento m non più abbinato, o "single", viene abbinato all'elemento che occupa la seconda posizione nella propria lista di preferenza. Di nuovo, i w abbinati simultaneamente a più elementi, scelgono il loro preferito tra questi, e gli altri ritornano single. Si ripete questo processo finché non vengono abbinati tutti gli elementi di W.

Il matching μ ottenuto è stabile infatti se esistessero $m \in M$ e $w \in W$ tali che m preferisca w a $\mu(m)$ e w preferisca m a $\mu(w)$ allora w dovrebbe essere stato abbinato prima a m e evidentemente deve aver rifiutato m per un elemento che preferisce.[3]

Dunque l'algoritmo GS genera una successione di proposte da parte degli elementi di M e rifiuti da parte di quelli di W.

La scelta dell'insieme che ha il compito di proporre le coppie è arbitraria, possiamo dunque far partire le proposte da W. I matching stabili che si ottengono al termine di queste due possibili procedure non necessariamente coincidono. In particolare se si parte da M allora il matching che si ottiene è tra tutti i matching stabili quello in cui gli $m \in M$ vengono abbinati ai loro preferiti (M-optimal), mentre quelli di W vengono abbinati agli elementi che ritengono i peggiori possibili (W-pessimal). Invertendo i ruoli tra M e W si ottiene il matching W-optimal e M-pessimal. Se tali matching coincidono allora esiste un unico matching stabile. Si verifica inoltre che queste procedure richiedono un tempo di esecuzione dell'ordine $O(n^2)$ $(n = |M| = |W|)$.

L'algoritmo Fondamentale è una formulazione iterativa dell'algoritmo GS (che invece ha una struttura in parallelo), presentata da Knuth nel 1972. L'estensione di questo algoritmo (Fondamentale Esteso) si deduce semplicemente osservando che se un elemento w viene abbinato a m allora si possono eliminare dalla lista di w tutti quegli elementi che seguono m, dal momento che seppure si cerchi di abbinare uno di questi a w è scontato che w lo rifiuti. Quest'ultimo algoritmo ha due interessanti proprietà: un qualsiasi matching perfetto del grafo ridotto ottenuto al termine dell'esecuzione di FE, eliminando gli archi corrisponenti alle coppie rifiutate, è un matching stabile; considerando invece le liste di preferenza ridotte, anch'esse ottenute eliminando le coppie rifiutate durante l'esecuzione, si verifica che gli elementi di W sono abbinati all'elemento in coda a queste liste mentre gli elementi di

[3] Indichiamo con μ sia il matching che la corrispondenza biunivoca $\mu(\cdot)$ da $M \cup W$ in se stesso indotta dal matching. Se $(x, y) \in \mu \Rightarrow \mu(x) = y$ e $\mu(y) = x$; se x non è abbinato in $\mu \Rightarrow \mu(x) = x$.

M a quelli in testa alle proprie liste.

Nella versione classica del problema, si pone $|M| = |W|$ e si considerano liste di preferenza complete e strettamente ordinate. Dunque tutte le coppie di elementi sono possibili coppie di un matching stabile e in ogni lista di preferenza ciascun elemento occupa una posizione distinta.

Dal 1985 a oggi, la ricerca in tale ambito si è orientata verso l'analisi di un modello che non preveda queste restrizioni proprio per ampliare il campo di applicazione dello stesso. L'obiettivo che ci proponiamo è di esplorare alcune varianti del problema evidenziando le principali differenze, e analizzando e implementando algoritmi risolutivi per ciascuna.

In un problema di matching stabile in generale si richiede che la cardinalità del matching sia massima, cioè che abbini il maggior numero di elementi. Nella versione classica del problema, un qualunque matching stabile è perfetto, se invece si ha $|W| < |M|$ allora si ricerca un matching stabile completo da W a M, cioè che abbini tutti gli elementi dell'insieme di cardinalità minore. La stabilità in tal caso deve prevedere che alcuni elementi di M non possono essere abbinati, dunque l'esistenza di un m single e un w che preferisca m all'elemento che gli è stato assegnato, generi instabilità.

L'algoritmo proposto per costruire un matching stabile non presenta sostanziali differenze rispetto ai precedenti. Dunque anche questo problema viene risolto in tempo $O(n^2)$, dove $n = |W|$.

Procediamo analizzando il caso in cui le liste di preferenza possono essere incomplete (Stable Marriage with Incomplete List): non tutti gli elementi sono *accettabili* gli uni per gli altri dunque non tutte le coppie sono *compatibili*, cioè possibili coppie di un matching. Un'istanza di questo tipo viene rappresentata mediante un grafo orientato e non completo.

Un matching stabile per istanze di questo tipo non è necessariamente perfetto ma si dimostra che tutti i matching stabili per una stessa istanza hanno la medesima cardinalità. Da una qualsiasi istanza di SMI si può costruire un'istanza dello stable marriage classico (SM) semplicemente completando in maniera arbitraria le liste incomplete e inserendo due pseudo-elementi m e w per separare gli elementi che effettivamente compaiono nella lista e quelli invece inseriti successivamente. Si fissa poi una lista di preferenza arbitraria per m e per w in modo tale che l'uno sia l'elemento "meno gradito" dall'altro e viceversa. Successivamente dimostriamo che esiste un matching stabile perfetto per l'istanza di SMI se e soltanto se il matching stabile per l'istanza di SM sopra descritta contiene la coppia (m, w). Inoltre se un tale matching stabile esiste allora tutti i possibili matching stabili per questa istanza sono perfetti, in caso contrario non esisterà alcun matching stabile perfetto.

Applicando FE al *grafo di compatibilità*, cioè al grafo bipartito non orientato i cui archi rappresentano esclusivamente coppie compatibili, si ottiene un matching stabile μ non necessariamente perfetto. Si dimostra inoltre che i matching stabili per istanze di SMI hanno tutti la stessa cardinalità ed in particolare se un elemento è abbinato in un matching stabile allora lo è in qualsiasi altro mentre se è single rispetto ad un matching stabile allora rimane single in ogni matching stabile. Questo ci garantisce che il matching stabile μ è massimo.

Successivamente prendiamo in considerazione il caso in cui le liste di preferenza possono essere non strettamente ordinate, cioè due o più elementi possono occupare la stessa posizione (in tal caso si dice che questi elementi formano un *tie* nella lista). Vogliamo dunque esprimere una possibile indifferenza verso due o più elementi. Un'istanza del problema SMT (*Stable Marriage with Tie*), è costituita da un grafo G bipartito non orientato e completo e da una funzione P che associa ad ogni nodo v del grafo una relazione d'ordine parziale (completa) sull'insieme di vertici $N(v) \cup \{v\}$ che individueremo con $\geq_v$.

In condizioni di possibile indifferenza tra gli elementi, si possono verificare esattamente tre casi in cui può essere compromessa la stabilità di un matching μ:

- se uno degli elementi x di una coppia di μ preferisce strettamente un qualche elemento y a $\mu(x)$ e y è indifferente nei confronti di x e $\mu(y)$.

- se uno degli elementi x di una coppia del matching preferisce strettamente qualche y a $\mu(x)$ o è indifferente e a sua volta y preferisce strettamente x a $\mu(y)$ o è indifferente;

- se uno degli elementi v di una coppia del matching preferisce strettamente qualche elemento u a $\mu(v)$ e u preferisce strettamente v a $\mu(u)$;

Nel primo caso parleremo di matching *fortemente* instabile, nel secondo di matching *super* instabile e nel terzo di matching *debolmente* instabile. Dunque possiamo distinguere tre tipi di stabilità.

Si sceglierà la nozione di stabilità adeguata al problema che si vuole risolvere.

Una differenza fondamentale tra i tre tipi di stabilità è che data una istanza di SMI esiste sempre un matching debolmente stabile mentre non necessariamente esistono matching fortemete o super stabili.

Sia $M = \{m_1, m_2\}$ e $W = \{w_1, w_2\}$. L'istanza rappresentata dalle seguenti liste non ammette matching fortemente stabili:

$$\begin{array}{llll} \mathbf{m_1}: & w_1, \quad w_2 & \mathbf{w_1}: & m_2, \quad m_1 \\ \mathbf{m_2}: & (w_1, \quad w_2) & \mathbf{w_2}: & m_2, \quad m_1 \end{array}$$

Le parentesi indicano che m_2 è indifferente nei confronti di w_1 e w_2 quindi tutte le preferenze sono strette tranne quella di m_2.
Il matching $\{(m_1, w_2), (m_2, w_2)\}$ è bloccato da (m_2, w_2) mentre il matching $\{(m_1, w_1), (m_2, w_2)\}$ è bloccato da (m_2, w_1).

Invece una istanza che non ammette matching super stabili si ha ad esempio quando c'è totale indifferenza tra gli elementi.

Per quanto riguarda la ricerca di matching fortemente stabili presentiamo l'algoritmo STRONG che è una estensione degli algoritmi precedenti.

- La sequenza di proposte rimane tale e quale tranne nel momento in cui un $m \in M$ ha due o più elementi in testa alla propria lista, in tal caso verrà abbinato a tutti questi simultaneamente.

- Se un elemento $w \in W$ viene abbinato a m allora tutti gli elementi che, nella lista di w, occupano posizioni strettamente inferiori a quella di m, vengono eliminati dalla lista di w e w viene eliminato dalla loro.

- Non appena una lista diventa vuota concludiamo che non esistono matching fortemente stabili.

- Quando invece vengono abbinati tutti gli elementi si considera il grafo bipartito $\hat{G}$ associato a tali abbinamenti. Se $\hat{G}$ contiene un matching perfetto allora esso costituisce un matching fortemente stabile per l'istanza di SMT (visto che gli archi di $\hat{G}$ rappresentano coppie fortemente stabili). In caso contrario deve esistere almeno un sottoinsieme di M *insufficiente*, cioè un insieme costituito da k elementi di M collettivamente abbinati a meno di k elementi di W, per qualche k. Si cerca allora l'insieme critico C, cioè quell'insieme insufficiente avente *deficiency* massima e non contenente a sua volta insiemi con questa stessa proprietà, perché si verifica che ogni w abbinato ad elementi di C non potrà avere come partner (fortemente) stabile nessuno di questi né qualsiasi altro elemento che occupi la stessa posizione nella lista di w. Tutte le coppie (m, w) con $m \in C$ dovranno essere perciò eliminate assieme alle coppie (m', w) con m' in coda alla lista di w e si riprenderà di nuovo la sequenza di proposte da parte di quegli elementi che, in seguito a questo processo di eliminazione, sono tornati single.

- L'intero processo si ripete finché non si ottiene un matching perfetto nel grafo bipartito degli abbinamenti, che costituirà quindi un matching fortemente stabile, oppure fin quando qualche lista diventa vuota, indicando così l'inesistenza di matching fortemente stabili.

Le operazioni di ricerca di un insieme critico e di un matching massimo sul grafo $\hat{G}$ fanno sì che il tempo di esecuzione totale dell'algoritmo sia dell'ordine di $O(n^4)$.

L'algoritmo SUPER invece, verifica l'esistenza di matching super stabili. Inizialmente procede esattamente come gli algoritmi precedenti dunque con la sequenza di proposte ed eliminazioni: questa sequenza può terminare con una o più liste di M vuote, indicando così l'assenza di matching super stabili, oppure con tutti gli m abbinati ad uno o più elementi di W (che occupano la stessa posizione nella lista di m).

Supponiamo che $w \in W$ sia abbinato a più elementi, allora qualsiasi elemento m tra questi si scelga come partner di w non potrà essere un partner super stabile, infatti w preferisce gli altri almeno quanto m e gli altri lo preferiscono almeno quanto tutti quegli elementi rimasti nelle loro liste. Riassumendo, nessun elemento di W che ha più possibili "partner" può ammettere come partner super stabile ne' questi ne' qualsiasi altro che occupi la loro stessa posizione, perciò tali coppie dovranno essere eliminate e si proseguirà la sequenza di proposte. L'intero processo si ripete finché non produce un abbinamento, che costituirà quindi un matching super stabile, oppure finché la lista di qualche m diventa vuota indicando così l'inesistenza di matching super stabili. Il tempo di esecuzione rimane $O(n^2)$.

Per quanto concerne la stabilità debole si osserva che è possibile passare da una istanza I di SMT ad una istanza I' di SM semplicemente fissando un ordinamento stretto arbitrario degli elementi che occupano una stessa posizione nelle liste dell'istanza I, ossia *separando arbitrariamente i tie* delle liste di I.

Operando una separazione dei tie dell'istanza di pagina 9 otteniamo le liste seguenti:

$$\begin{array}{lll} \mathbf{m_1}: & w_1, & w_2 \\ \mathbf{m_2}: & w_1, & w_2 \end{array} \qquad \begin{array}{lll} \mathbf{w_1}: & m_2, & m_1 \\ \mathbf{w_2}: & m_2, & m_1 \end{array}$$

$$\begin{array}{lll} \mathbf{m_1}: & w_1, & w_2 \\ \mathbf{m_2}: & w_2, & w_1 \end{array} \qquad \begin{array}{lll} \mathbf{w_1}: & m_2, & m_1 \\ \mathbf{w_2}: & m_2, & m_1 \end{array}$$

Un matching stabile per I' è sicuramente un matching debolmente stabile per I. Quindi applicando l'algoritmo FE ad una tale istanza di SM si ottiene un matching debolmente stabile per l'istanza di SMT.

L'ultima delle varianti del problema del matrimonio stabile classico che analizziamo è quella in cui vengono meno entrambe le restrizioni che prevedevano che le liste di preferenza fossero complete e strettamente ordinate.

Una istanza I di SMTI (*Stable Marriage with Ties and Incomplete lists*) è costituita da un grafo G orientato non necessariamente completo e da liste di preferenza incomplete e/o non strettamente ordinate. Come per SMI, è necessario costruire il grafo di compatibilità che conterrà esclusivamente quegli archi di G che connettono elementi compatibili. Per semplicità ometteremo questo passaggio supponendo implicitamente che il grafo G sia già epurato da coppie incompatibili e sia quindi non orientato.

Valgono tre possibili nozioni di stabilità che, rispetto alle precedenti, devono tenere in considerazione il fatto che le liste possono essere incomplete, quindi come in SMI alcuni elementi possono rimanere single.

Anche per le istanze di SMTI non sempre esistono matching super stabili e matching fortemente stabili. Presentiamo allora due algoritmi, SUPER2 e STRONG2, che verificano l'esistenza di matching super stabili e fortemente stabili, rispettivamente, in tempo polinomiale ($O(n^2)$ e $O(n^4)$).

Per queste due nozioni di stabilità il problema rimane dello stesso ordine di difficoltà, la stessa cosa non si può dire per la stabilità debole, infatti la ricerca di matching debolmente stabili di cardinalità massima per istanze di SMTI diventa un problema NP arduo.

Nell'ultima parte di questo lavoro definiamo il problema di decisione WEAK-SMTI che ha per istanze le coppie (I, k) (dove I è una istanza di SMTI e k è un intero) e si chiede se esiste un matching debolmente stabile di cardinalità k. Facciamo riferimento ad un articolo di Manlove e altri, pubblicato nel 1999 [15], per provare la NP-completezza di WEAK-SMTI realizzando una riduzione dal problema EXACT-MAXIMAL-MATCHING. Infine descriviamo una proprietà dei matching debolmente stabili che ci garantisce che operando una separazione arbitraria dei tie e applicando all'istanza così ottenuta l'algoritm FE, si ottiene un matching stabile la cui cardinalità si discosta dal valore ottimo, cioè massimo, per un fattore 2.

In appendice presentiamo i listati di programmi che implementano in linguaggio C diversi algorimi analizzati in questo lavoro.

Bibliografia

[1] L. Lovász, M.D. Plummer, *Matching Theory*, North-Holland Mathematics Studies vol.121, 1986.

[2] R. Diestel, *Graph Theory*, second edition, Springer, 2000.

[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduzione agli algoritmi*, Jackson libri, 1999.

[4] C.H. Papadimitriou, K. Steiglitz, *Combinatorial optimization*, Dover publications, 1998.

[5] D. Gale, L.S. Shapley, *College admissions and the stability of the stable marriage*, Amer. Math. Monthly 69, (1962), 9-15.

[6] D. Gusfield, R. Irving, *The stable marriage problem: structure and algorithms*, MIT Press Series in the Foundations of Computing, MIT Press, Cambridge, Mass., (1989)

[7] D.E. Knuth, *Stable Marriage and its Relation to Other Combinatorial Problems*, CRM Proceedings & Lecture Notes Vol. 10, AMS, (1997). Traduzione inglese di *Mariages Stables*, Les Presses De l'Université de Montréal, Montréal (1976)

[8] D.G. Mc Vitie, L.B. Wilson, *The stable marriage problem*, Comm. of the ACM 14, (1971), 486-490.

[9] D. Kapur, M.S. Krishnamoorthy, *Worst case choice for the stable marriage problem*, Inform. Proc. Letters, 21 (1985) 27-30.

[10] M. Sotomayor, *A non-constructive elementary proof of existance of stable marriages*, Games and Economic Behavior 13, (1996) 135-137.

[11] M. Yannakakis, F. Gavril, *Edge dominating sets in graphs*, SIAM J. Applied Mathematics, 18 n.1, (1980), 364-372.

[12] Cheng Ng, D.S. Hirshberg, *Lower bounds for the stable marriage problem and its variants*, SIAM J. Comput. 19, (1990), 71-77.

[13] R.W. Irving, *Stable marriage and indifference*, Discrete Applied Mathematics 48 (1994), 261-272.

[14] D. Manlove, *Stable marriage with ties and unacceptable partners*, Technical ReportTR-1999-29 University of Glasgow, Department of Computing Science (1999).

[15] K. Iwama, D. Manlove, S. Miyazaki, Y. Morita, *Stable marriage with incomplete lists and ties*, Proc. ICALP '99: the 26th Internat. Colloq. on Automata, Languages and Programming, LNCS vol.1644, Springer, Berlin, (1999), 443-452.

[16] M.R. Garey, D.S. Johnson, *Computer and intractability, a guide to the theory of NP-Completeness*, Freeman, San Francisco, 1979.

[17] F. Manlove, W. Irving, K. Iwama, S. Miyazaki, Y. Morita, *Hard variants of stable marriage*, Theoretical Computer Science 276 (2002) 261-279.

[18] M.M. Halldòrsson, K. Iwama, S. Miyazaki, H. Yanagisawa, *Improved approximation of the stable marriage problem*, Proc. ESA 2003, LNCS 2832 (2003), 266-277.

[19] K. Iwama, S. Miyazaki, K. Okamoto, *A* $(2 - c\frac{log(n)}{n})$*-approximation algorithm for the stable marriage problem*, SWAT 2004, LNCS 3111 (2004), 349-361.