Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2011-03. Published by the Egerváry Research Group, Pázmány P. sétány 1/C,
H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres. ISSN 1587–4451.

Approximation of Maximum Stable Marriage

Zoltán Király

March 31, 2011

Approximation of Maximum Stable Marriage

Zoltán Király*

Abstract

In 1962 Gale and Shapley gave their famous simple "proposal" algorithm, that always finds a stable matching in bipartite graphs. If ties are allowed in the preference lists, we are usually interested not only in finding some stable matching, but one with maximum size. This problem is APX-hard, and probably cannot be approximated within factor 4/3. We first survey the history of approximation algorithms for this problem. McDermid [M 2009] gave the first 3/2-approximation, and very recently Paluch [P 2010] gave a linear time algorithm with the same ratio. In this paper we give a much simpler algorithm, which is only a slight modification of the historical algorithm of Gale and Shapley. This also gives 3/2-approximation in linear time. The proof of the approximation ratio is also uncomplicated, thus it serves as a good example for teaching purposes.

Keywords: stable marriage, Gale-Shapley algorithm, approximation.

1 Introduction

An instance of the stable marriage problem consists of a set U of men, a set W of women, and a preference list for each person, that is a weak linear order (ties are allowed) on some members of the opposite gender. A pair $(m \in U,\ w \in W)$ is called *acceptable* if m is on the list of w and w is on the list of m. We model acceptable pairs with a bipartite graph $G = (U, W, E)$, (where E is the set of acceptable pairs; we may assume that if w is not on the list of m then m is also missing from the list of w). Note, that total length of the lists is proportional to $|E|$. A *matching* in this graph consists of mutually disjoint acceptable pairs.

We store the weak order of the lists as priorities. For an acceptable pair (m, w), let $\text{pri}(w, m)$ be an integer from 1 up to $|U|$ representing the priority of m for w. We say that $w \in W$ *strictly prefers* $m_1 \in U$ to $m_2 \in U$ if both m_1 and m_2 are acceptable for w, and $\text{pri}(w, m_1) > \text{pri}(w, m_2)$. Ties are represented by equal priorities, e.g., if m_1 and m_2 are tied in w's list, then $\text{pri}(w, m_1) = \text{pri}(w, m_2)$. We will later break up some ties, dividing the men in a tie into two groups, and we will say, that w prefers one group to the other. So there will be a case, that w *prefers* m_1 to m_2, but not strictly

*Department of Computer Science and Communication Networks Laboratory, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary. Research supported by EGRES group (MTA-ELTE), OTKA grants CNK 77780, CK 80124 and TÁMOP grant 4.2.1./B-09/1/KMR-2010-0003.

prefers, and this means, that $\text{pri}(w, m_1) = \text{pri}(w, m_2)$, but m_1 is in the preferred group and m_2 is not. If both of them are in the same group then we say m_1 and m_2 are *alike* for w.

We define $\text{pri}(m, w)$ similarly, of course, $\text{pri}(m, w)$ is not related to $\text{pri}(w, m)$. We represent these priorities in the figures by writing $\text{pri}(m, w)$ and $\text{pri}(w, m)$ close to the corresponding endvertex of edge mw ($\text{pri}(m, w)$ is written next to m, while $\text{pri}(w, m)$ is written next to w). For a man m we will modify his list in either of two ways. Sometimes we delete a woman from the list. Sometimes (at a given point, when his list is empty) we will restore the original list of m. We mean by "*favorite woman*" of m, the mostly preferred woman still on m's list; if there are more alike women on the top of the list, we choose one of them arbitrarily.

Remark. The author apologizes that the text of this paper is not PC. Just like the one of Gale and Shapley [GS 1962], or the phrases used in other mathematical papers on this topic. It would indeed be possible to change the terminology, but we find this approach pretty well-mannered.

Let M be a matching. If m is *matched* in M, or in other words, if m is *engaged* then we denote m's fiancée by $M(m)$. Similarly we use $M(w)$ for the fiancé of an engaged woman w.

Definition 1.1. A pair (m, w) is *blocking* M, if $mw \in E \setminus M$ (they are an acceptable pair and they are not matched) and

- w is either not engaged or w strictly prefers m to her fiancé, **and**
- m is either not engaged or m strictly prefers w to his fiancée.

Definition 1.2. A matching is called *stable* if there is no blocking pair.

It is well-known that a stable matching always exists and can be found in linear time. The celebrated algorithm of Gale and Shapley [GS 1962] is the following.

A man can either be a *lad* or an *old bachelor*. A lad can either be *active* or *engaged*. A woman can either be *maiden* or *engaged*. At the beginning every man is a lad and every woman is a maiden.

Algorithm GS

While there exists an active man m, he proposes to his favorite woman w. If w accepts his proposal, they become engaged. If w refuses him, m will delete w from his list, and will remain active.
When a woman w gets a new proposal from man m, she always accepts this proposal, if she is a maiden. She also accepts this proposal, if she prefers m to her current fiancé. Otherwise she refuses m.
If w accepted m, then she refuses her previous fiancé, if there was one (breaks off her engagement), and becomes engaged to m.
If a man m was engaged to a woman w, and later w refuses him, then m becomes active again, and deletes w from his list.
If the list of m becomes empty, he will turn into an old bachelor and will remain inactive forever.

After Algorithm GS finishes, the engaged pairs make up the output matching M (we may imagine that this time all the engaged pairs get married).

Theorem 1.3 (Gale and Shapley [GS 1962]). *Algorithm GS always ends in a stable matching M. This algorithm runs in $O(|E|)$ time.*

An interesting problem, motivated by applications, is to find a stable matching of maximum size. This problem is known to be NP-hard for even very restricted cases [IMMM 1999, MIIMM 2002]. Moreover, it is APX-hard [HIIMMS 2003] and supposing P≠NP, it cannot be approximated within a factor of strictly less than 21/19, even if ties occur only in the preference lists on one side only, furthermore, if every list is either totally ordered or consists of a single tied pair [HIMY 2007]. Moreover, refining the ideas of [HIMY 2007], Yanagisawa [Y 2007] proved, that an approximation within a factor of $4/3-\varepsilon$ implies $(2-\varepsilon')$-approximation of vertex cover, and this result also applies to the case when each tie has length two. If, moreover, ties occur only in the preference lists on one side only, he proved, that an approximation within a factor of $5/4-\varepsilon$ has the same implication. We note that interestingly the minimization version (where we are looking for a stable matching of minimum size) is also APX-hard [HIIMMS 2003].

As the applications of this problem are important (see e.g., in [IM 2007, IM 2008], where detailed lists of known and possible applications are given, that motivate investigating refined approximations), researchers started to develop good approximation algorithms in the past four years. We say that an algorithm is r-approximating, if it gives a stable matching M with size $|M| \geq (1/r) \cdot |M_{\text{opt}}|$, where M_{opt} is a stable matching of maximum size. Observe, that after a run of GS no unmarried man and unmarried woman can form an acceptable pair. Consequently Algorithm GS gives a 2-approximation, and for complete bipartite graphs (every woman-man pair is acceptable), it gives an optimum. The first non-trivial approximation algorithm was given by Halldórsson et al. [HIMY 2007], where they give a 13/7-approximation if all ties are of length two. The breakthrough was achieved by Iwama, Miyazaki and Yamauchi [IMY 2007] who gave a 15/8-approximation (for any length of ties). This was later improved by Irving and Manlove [IM 2007] to a 5/3-approximation for the special case where ties are allowed on one side only, and moreover only at the ends of the lists. Their algorithm also applies to the Hospitals/Residents problem (see later) if residents have strictly ordered lists. If, moreover, ties are of size 2, Halldórsson et al. [HIMY 2007] gave an 8/5-approximation and in [HIMY 2004] they described a randomized algorithm for this special case with expected ratio of 10/7.

We proposed a simple linear time 5/3-approximating algorithm (called GSA2) for the general case of this problem in [K2 2008, K3 2008, K 2009], that we sketch in Section 4. At the same time, an even simpler 3/2-approximation (called GSA1) was given for the special case where ties are allowed on one side only. For this algorithm, the proof was also very short, see Section 2. This is also valid for the practically important "Hospitals/Residents" problem, where the lists of the Residents are strict, see Section 3. During the talks given at the MATCH-UP workshop in Reykjavík and at ESA in Karlsruhe, we posed several questions, conjectures and open problems. Many of them were answered in the meanwhile.

The conjecture stating that the performance ratio proved for GSA2 is sharp proved to be true by Yanagisawa [Y 2008], who gave a simple example where GSA2 really gives a matching of size exactly $\frac{3}{5} \cdot M_{\text{opt}}$.

Irwing and Manlove [IM 2009] implemented a basic version of our algorithm for the one-sided-ties Hospitals/Residents problem and gave a detailed comparison with their best heuristic. They tested carefully the algorithms with real-life and artificial data. They concluded that for the most cases their best heuristic executed the best, but, on the average, our algorithm also gave a stable assignment of size at least 99.41% of their best one. We do not know too many other examples, where an algorithm with a guaranteed approximation ratio is so close in the practice to the best known heuristic.

We also conjectured, that for the one-sided-ties case, if someone gives $(3/2-\varepsilon)$-approximation, it implies something "surprising" (for example $(3-\varepsilon')$-approximation for vertex cover in 3-uniform hypergraphs). This was – in some sense – disproved by Iwama, Miyazaki, and Yanagisawa [IMY 2010]. They gave a $25/17 \approx 1.47$-approximation for this special case. They solved the relaxed version of an appropriate ILP formulation, and used the fractional optimum to guide the tie-breaking process. Besides this, their algorithm is similar to our one, but the analysis is much harder. Of course, as they have to solve an LP, it gives non-linear running time. But we consider it important, that they could break the 3/2-barrier (the same barrier related to Christofides' algorithm is untouched despite the notable effort of many researchers).

We had another conjecture, stating that a simple modification/repetition of GSA2 gives 3/2-approximation for the general problem. It was (partially) answered by McDermid [M 2009], who gave the first 3/2-approximation for the general case. He used our algorithm, but not with simple repetitions. Actually he used novel and rather complicated techniques, and his algorithm needs $O(n\sqrt{n}|E|)$ time (where $n = |U| + |W|$).

Recently Paluch [P 2010] gave a 3/2-approximation algorithm for the general case with linear running time. This algorithm is still quite complicated, uses many concepts, and the analysis is also lengthy.

In Section 5 we give a simple linear time 3/2-approximation, with a much simpler proof. We lean on two important ideas of Paluch, but our algorithm is a slightly modified version of GSA1 given in the next section.

2 Men have strictly ordered lists

In this section we suppose that the lists of men are strictly ordered. A man can be a lad, a *bachelor* or an old bachelor. If there are two men, m_1 and m_2 with the same priority on a woman w's list, and m_1 is a lad but m_2 is a bachelor, then w **prefers bachelor** m_2 to **lad** m_1. In the description of the algorithm, differences from Algorithm GS are set in boldface.

Algorithm GSA1

While there exists an active man m, he proposes to his favorite woman w. If w accepts his proposal, they become engaged. If w refuses him, m will delete w from his list, and will remain active.
When a woman w gets a new proposal from man m, she always accepts this proposal, if she is a maiden. She also accepts this new proposal, if she prefers m to her current fiancé. Otherwise she refuses m.
If w accepted m, then she refuses her previous fiancé, if there was one (breaks off her engagement), and becomes engaged to m.
If m was engaged to a woman w and later w refuses him, then m becomes active again, and deletes w from his list.
If the list of m **becomes empty for the first time, he turns into a bachelor, his original list is recovered, and he reactivates himself. If the list of** m **becomes empty for the second time**, he will turn into an old bachelor and will remain inactive forever.

This simple algorithm runs in $O(|E|)$ time, as there are at most $2|E|$ proposals altogether. It is easy to see that Algorithm GSA1 gives a stable matching M.

Theorem 2.1 (2008)**.** *If men have strictly ordered preference lists, M is the output of Algorithm GSA1 and M_{opt} is a maximum size stable matching then $|M_{\text{opt}}| \leq \frac{3}{2} \cdot |M|$.*

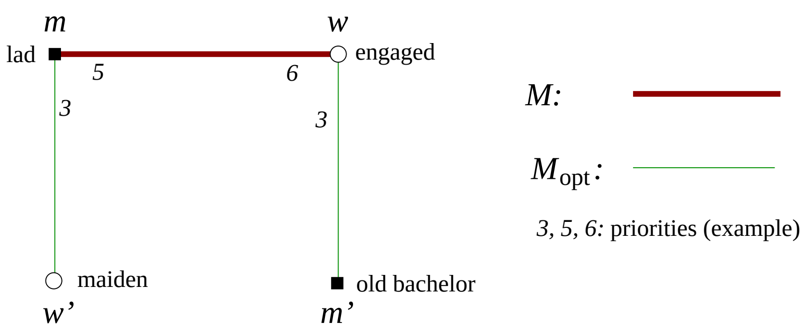

Figure 1: A short augmenting path

Proof. Take the union of M and M_{opt}. We consider common edges as a two-cycle. Each component of $M \cup M_{\text{opt}}$ is either an alternating cycle (of even length) or an alternating path. An alternating path component is called augmenting path if both end-edges are in M_{opt}. An augmenting path is called short, if it consists of 3 edges (see Figure 1). It is enough to prove that in each component there are at most 3/2 times as many M_{opt}-edges as M-edges. This is clearly true for each component except for a short augmenting path.

We claim that a short augmenting path cannot exist. Suppose that $M(m) = w$, $M_{\text{opt}}(m) = w' \neq w$, $M_{\text{opt}}(w) = m' \neq m$ and that m' and w' are single in M. Observe first that w' is a maiden, thus she never got a proposal during Algorithm GSA1. Consequently m is a lad, who prefers w to w'. As the algorithm finished, m' is an old bachelor, so he proposed to w also as a bachelor (see Figure 1), but w preferred m to m'. Consequently w strictly prefers m to m'. However, in this case edge mw blocks M_{opt}, a contradiction. □

3 Hospitals/Residents with strictly ordered residents' lists

In the *Hospitals/Residents* problem, the roles of women are played by hospitals and the roles of men are played by residents. Moreover, each hospital w has a positive integer capacity $c(w)$, the number of free positions. Instead of matchings, we consider *assignments*, that is a subgraph F of G, such that all residents have degree at most one in F, and each hospital w has degree at most $c(w)$ in F, i.e., $d_F(w) \leq c(w)$. For a resident m who is assigned, $F(m)$ denotes the corresponding hospital. For a hospital w, $F(w)$ denotes the set of residents assigned to it. We say that hospital w is *full* if $|F(w)| = c(w)$ and otherwise *under-subscribed*. Here a pair (m, w) is *blocking*, if $mw \in E \setminus F$ (they are an acceptable pair and they are not assigned to each other) and

- m is either unassigned or $\text{pri}(m, w) > \text{pri}(m, F(m))$, and
- w is either under-subscribed or $\text{pri}(w, m) > \text{pri}(w, m')$ for at least one resident $m' \in F(w)$.

An assignment is *stable* if there is no blocking pair. It is well known that Algorithm GS with a slight modification always gives a stable assignment in linear time. If hospital w gets a new proposal from resident m then it accepts him, either if w is under-subscribed or if it prefers m to its less preferred resident. When hospital w is full and accepts, it rejects an arbitrary assigned less preferred resident. This algorithm clearly gives a stable assignment, and it is easy to see that it can be implemented to run in $O(|E|)$ time.

As before, we are interested in giving a maximum size assignment, i.e., a stable assignment F with maximum number of edges (that is a maximum number of assigned residents). We consider here the Hospitals/Residents problem with the restriction that residents have strict orders on acceptable hospitals. Note, that for real-life applications of this scheme, this assumption is realistic. Algorithm GSA1 is also easy to modify for this setup, the only difference is that if a resident's list gets empty, he becomes an honored resident, his list is recovered, and he starts again. A hospital always prefers an honored resident to a non-honored one, if they are alike in the original preference list. Call this algorithm HRGSA1.

Theorem 3.1 (2008)**.** *If residents have strictly ordered preference lists, F is the output of Algorithm HRGSA1 and F_{opt} is any maximum size stable assignment then*

$$|F_{\text{opt}}| \leq \frac{3}{2} \cdot |F|.$$

We have an example that shows an input, where a possible run of this algorithm really gives only this approximation ratio. However, with a different type of restriction we are able to prove a stronger theorem. For a hospital w let $\tau(w)$ denote the length of the longest tie for w, and let $\lambda := \max_{w \in W} \tau(w)/c(w)$.

Theorem 3.2 (2008)**.** *Algorithm HRGSA1 gives approximation ratio not worse than*

$$\frac{4}{3} + \frac{\lambda}{6}$$

Corollary 3.3. *If for each hospital, the length of any tie is not more than half of the hospital's capacity, then the approximation ratio of our algorithm is at most* $\frac{17}{12}$*. If every tie has length at most three, and every hospital has capacity at least* 100*, then the approximation ratio is better than* 1.339*.*

4 General stable marriage

Now we consider the general maximum stable marriage problem. Here we run Algorithm GSA1, and start a second phase after it finishes. In the second phase women propose to men, and men decide whether to refuse or accept. We need some new terminology (these phrases are used in this section only). At the beginning of the second phase every maiden becomes an active mature woman, engaged women are called immature and remain inactive. When a immature woman is refused for the first time, we say that she is on the shelf. If every accessible man refused a woman on the shelf, she becomes to a mature woman. When every accessible man refused a mature woman, she becomes a spinster. If there are some women that are alike for man m, we say that m prefers mature women to women on the shelf, and he prefers women on the shelf to immature women.

Algorithm GSA2

Run GSA1, and then change the roles of women and men. Start the second phase, maidens become active mature women, and in this phase women propose to men.
While there exists an active woman w, she proposes to his favorite man m. If m accepts her proposal, they become engaged. If m refuses her, w will delete m from her list, and will remain active.
If a man m gets a new proposal from woman w, he always accepts this proposal, if he is not engaged. He also accepts this new proposal if, he prefers w to her current fiancée. Otherwise he refuses w.
If m accepted w, then he refuses his previous fiancée, if there was one (breaks off his engagement), and becomes engaged to w.
If w was engaged to a man m and later m refuses her, then w becomes active again, deletes m from her list, and if this was the first refusal in her life, then she is on the shelf.
If the list of woman w, who is on the shelf, becomes empty, her original list is recovered, and she reactivates herself as a mature woman. If the list of a mature woman w becomes empty, she will turn into a spinster and will remain inactive forever.

This simple algorithm runs in $O(|E|)$ time, as there are at most $4|E|$ proposals altogether. It is easy to see that Algorithm GSA2 gives a stable matching M.

Theorem 4.1 (2008)**.** *If M is the output of Algorithm GSA2 and M_{opt} is any maximum size stable matching then*

$$|M_{\text{opt}}| \leq \frac{5}{3} \cdot |M|.$$

5 The new algorithm

A man can either be a *lad*, or a *bachelor*, or an *old bachelor*. A lad or a bachelor can either be *active* or *engaged*. If women w_1 and w_2 have the same priority on m's list, and w_1 is maiden but w_2 is engaged, then m **prefers maiden** w_1 to **engaged** w_2. An engaged man is **uncertain**, if his list contains a woman he prefers to his actual fiancée (this can happen, if there were two maidens with the same highest priority in m's list, and m became engaged to one of them).

A woman can either be *maiden* or *engaged*. An engaged woman is **flighty**, if her fiancé is uncertain. If there are two men, m_1 and m_2 with the same priority on a woman w's list, and m_1 is a lad, but m_2 is a bachelor, then w *prefers bachelor* m_2 to *lad* m_1.

At the beginning every man is a lad and every woman is a maiden. In the description of the algorithm, differences from GSA1 are set in boldface.

New Algorithm

While there exists an active man m, he proposes to his favorite woman w. If w accepts his proposal, they become engaged. If w refuses him, m will delete w from his list, and will remain active.
When a woman w gets a new proposal from man m, she always accepts this proposal, if she is a maiden **or a flighty engaged woman**. She also accepts this proposal, if she prefers m to her current fiancé. Otherwise she refuses m.
If w accepted m, then she refuses her previous fiancé, if there was one (breaks off her engagement), and becomes engaged to m.
If m was engaged to a woman w and later w refuses him, then m becomes active again, and deletes w from his list, **except if w is flighty, in this case m keeps w on the list**.
If the list of m becomes empty for the first time, he turns into a bachelor, his original list is recovered, and he reactivates himself. If the list of m becomes empty for the second time, he will turn into an old bachelor and will remain inactive forever.

After the algorithm finishes, the engaged pairs get married and form matching M.

Lemma 5.1. *When a woman gets the first proposal, she becomes engaged, and will never become maiden again. A woman can become flighty only after the first proposal she gets. After the second proposal a woman can never be flighty. If a woman changes her fiancé, then she always prefers the new fiancé to the previous one, except if she* *is flighty, when she may refuse a preferred man, but in this case she remains on the refused man's list. Formally: suppose the second proposal to woman w was made by m, after this proposal, w's fiancé was m_1, and later the fiancés of w were $m_2, \ldots, m_t$ in this order. Then* $\mathrm{pri}(w, m) \le \mathrm{pri}(w, m_1) \le \ldots \le \mathrm{pri}(w, m_t)$.

Proof. The only statement that needs a proof is, that after getting the second proposal a woman w cannot be flighty. However, in this case she got the last proposal as an engaged woman, so either she kept her first fiancé, consequently she was not flighty, or her new fiancé m cannot be uncertain. (If there is a maiden w' on the list of m with the same priority as w, then m would prefer w' to w, so he would propose to w' first.) □

Lemma 5.2. *The matching M given by the new algorithm is stable.*

Proof. For suppose mw is a blocking pair. If w is maiden then she did not get any proposals, so m did not reach her when he processed his list, consequently he is engaged to a preferred woman w' (though it can be the case, that w' is flighty and now m would prefer w to w', but he does not strictly prefer her, so pair mw cannot be blocking).

If m is not married then he is an old bachelor, consequently he proposed to w twice. By the previous lemma the priority of the fiancé of woman w is monotonically increasing after the second proposal she got. So the husband of w cannot be strictly less preferred than m.

Finally suppose both m and w are married, the wife of m is w', and the husband of w is m'. As pair mw is blocking, m strictly prefers w to w', so m also proposed to w and she refused him. If at the time of this refusal w was not flighty, then she got finally a husband not worse than m. If she was flighty that time, then she remained on the list of m, so m proposed to her again before w'. In all cases we came to a contradiction. □

Lemma 5.3. *There is no short augmenting path.*

Proof. Suppose $m'wmw'$ is a short augmenting path (see Figure 1). The algorithm finished, so m' is an old bachelor, but m is a lad, because w' remained a maiden. As M_{opt} is a stable matching, edge mw does not block it, so either w does not strictly prefer m to m', or m does not strictly prefer w to w'. In the second case – as w' was always a maiden – after m proposed to w and got engaged to her, he became uncertain, thus w became flighty. It is impossible that later w refuses m, because after this m would propose to the preferred maiden w' before he proposes again to w. So w remained flighty at the end, meaning that she got only one proposal, but this is impossible, because m' proposed to her twice.

Assume w does not strictly prefer m to m'. Observe, that when m becomes a bachelor, he has no maidens on his recovered list, consequently an uncertain bachelor cannot exist. Take the moment, when m' proposed to w as a bachelor. If w refused m' at this time, then she was not flighty, because a flighty woman never refuses a proposal. Thus by Lemma 5.1, husband m of w is not less preferred, but this is a contradiction, as a woman prefers bachelors to lads. □

These lemmas together give:

Theorem 5.4. *The new algorithm always gives a stable matching in linear time for the general problem, and is* 3/2*-approximating.*

Remark. *Using these ideas one can easily develop a linear time* 3/2*-approximating Hospital-proposal algorithm for the Hospitals/Residents problem, where Residents have strictly ordered lists. This algorithm may have some importance in the applications, because the arising assignment is probably better for the Hospitals than the one given by in Section 3.*

References

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

[HIIMMS 2003] M. M. Halldórsson, R. W. Irving, K. Iwama, D. F. Manlove, S. Miyazaki, Y. Morita, S. Scott, *Approximability results for stable marriage problems with ties Theor. Comput. Sci.* ***306*** *(2003) pp. 431–447.*

[HIMY 2004] M. M. Halldórsson, K. Iwama, S. Miyazaki, H. Yanagisawa, *Randomized approximation of the stable marriage problem Theor. Comput. Sci.* ***325*** *(2004) pp. 439–465.*

[HIMY 2007] M. M. Halldórsson, K. Iwama, S. Miyazaki, H. Yanagisawa, *Improved approximation results for the stable marriage problem ACM Trans. Algorithms* ***3*** *No. 3, (2007) Article 30.*

[IM 2007] R. W. Irving, D. F. Manlove, *Approximation algorithms for hard variants of the stable marriage and hospitals/residents problems Journal of Combinatorial Optimization (2007) DOI: 10.1007/s10878-007-9133-x*

[IM 2009] R. W. Irving, D. F. Manlove, *Finding Large Stable Matchings Journal of Experimental Algorithmics* ***14*** *(2009) DOI: 10.1145/1498698.1537595*

[IMMM 1999] K. Iwama, D. F. Manlove, S. Miyazaki, Y. Morita, *Stable marriage with incomplete lists and ties Proceedings of the 26th International Colloquium on Automata, Languages and Programming* ***LNCS 1644*** *(1999) pp. 443–452.*

[IMY 2007] K. Iwama, S. Miyazaki, N. Yamauchi, *A* 1.875*-approximation algorithm for the stable marriage problem SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (2007) pp. 288–297.*

[IM 2008] K. Iwama, S. Miyazaki, *A survey of the stable marriage problem and its variants. Proceedings of the International Conference on Informatics Education and Research for Knowledge-Circulating Society, IEEE Computer Society, Washington (2008) pp. 131–136.*

[IMY 2010] K. Iwama, S. Miyazaki, H. Yanagisawa, *A 25/17-Approximation Algorithm for the Stable Marriage Problem with One-Sided Ties ESA '10, LNCS* **6347** *(2010) pp. 135–146, DOI: 10.1007/978-3-642-15781-3_12*

[K1 2008] Z. Király, *Better and simpler approximation algorithms for the stable marriage problem Egres Technical Report TR-2008-04, www.cs.elte.hu/egres/*

[K2 2008] Z. Király, *Better and simpler approximation algorithms for the stable marriage problem Proceedings of MATCH-UP 2008: Matching Under Preferences – Algorithms and Complexity, Satellite workshop of ICALP, July 6, 2008, Reykjavík, Iceland pp. 36–45.*

[K3 2008] Z. Király, *Better and Simpler Approximation Algorithms for the Stable Marriage Problem ESA 2008, Lecture Notes in Computer Science* **5193** *(2008) pp. 623–634.*

[K 2009] Z. Király, *Better and Simpler Approximation Algorithms for the Stable Marriage Problem Algorithmica* **60** *(1), DOI: 10.1007/s00453-009-9371-7, (2009 (online), 2011), pp. 3–20.*

[MIIMM 2002] D. F. Manlove, R. W. Irving, K. Iwama, S. Miyazaki, Y. Morita, *Hard variants of stable marriage Theor. Comput. Sci.* **276** *(2002) pp. 261–279.*

[M 2009] E. J. McDermid *A $\frac{3}{2}$-approximation algorithm for general stable marriage Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece* ***LNCS 555*** *(2009) pp. 689–700.*

[P 2010] K. Paluch *Faster and simpler approximation of stable matchings arXiv:0911.5660v2 Submitted to Operations Research Letters.*

[Y 2007] H. Yanagisawa, *Approximation Algorithms for Stable Marriage Problems PhD Thesis www.lab2.kuis.kyoto-u.ac.jp/~yanagis/thesis_yanagis.pdf (2007)*

[Y 2008] H. Yanagisawa, *Personal communication (2008)*