The Price of Matching with Metric Preferences

Yuval Emek[1], Tobias Langner[2], and Roger Wattenhofer[2]

[1] Technion, Israel
[2] ETH Zürich, Switzerland

Abstract. We consider a version of the Gale-Shapley stable matching setting, where each pair of nodes is associated with a (symmetric) matching cost and the preferences are determined with respect to these costs. This stable matching version is analyzed through the Price of Anarchy (PoA) and Price of Stability (PoS) lens under the objective of minimizing the total cost of matched nodes (for both the marriage and roommates variants). A simple example demonstrates that in the general case, the PoA and PoS are unbounded, hence we restrict our attention to metric costs. We use the notion of α-stability, where a pair of unmatched nodes defect only if both improve their costs by a factor greater than $\alpha \geq 1$. Our main result is an asymptotically tight trade-off, showing that with respect to α-stable matchings, the Price of Stability is $\Theta\big(n^{\log(1+\frac{1}{2\alpha})}\big)$. The proof is constructive: we present a simple algorithm that outputs an α-stable matching satisfying this bound.

1 Introduction

The aim of this paper is to connect two classic approaches towards *matching*. The first approach tackles matching from a (global) optimization angle à la Edmonds [15, 16]: given $2n$ nodes with pairwise costs $c(x, y) = c(y, x) \in \mathbb{R}_{>0}$, the goal is to construct a perfect matching that minimizes the total cost. The second approach tackles matching from the (local) selfish angle à la Gale and Shapley [18]: each node is equipped with a preference list ranking its potential matches and a matching is *stable* if no two unmatched nodes prefer each other over their current matches.

We consider a restricted case of the stable matching realm, where the nodes preferences are determined based on the aforementioned pairwise costs $c(\cdot, \cdot)$ so that node x prefers node y over node y' if and only if $c(x, y) < c(x, y')$, and focus on the following question: How does the requirement to output a (locally) stable matching affect its (global) total cost? In attempt to provide a quantitative answer to this question, we shall look at matching instances through the *Price of Stability (PoS)* lens that compares the min-cost stable matching to the unrestricted optimum, measuring the ratio of their respective costs. In fact, to provide a deeper understanding of the delicate balance between the global matching cost and its local stability, we generalize the problem by using the notion of an *α-stable* matching for $\alpha \geq 1$, in which no pair of unmatched nodes can defect and thus improve their costs by a factor (strictly) greater than α.

Unfortunately, in general, the Price of Stability may be unbounded, as the following simple example shows: Let G be a complete graph on four nodes u_1, u_2, v_1, v_2 with edge costs $c(u_1, v_1) = c(u_2, v_2) = 1$, $c(u_1, u_2) = \varepsilon$ for some small $\varepsilon > 0$, and $c(v_1, v_2) = c(u_1, v_2) = c(u_2, v_1) = C$ for some large C. Then, the optimal perfect matching matches u_i to v_i for $i = 1, 2$ with a cost of 2, whereas an α-stable matching for any reasonable value of α must match u_1 to u_2, and hence also v_1 to v_2 which incurs an arbitrarily large cost.

Fortunately, real-world matching instances often exhibit *metric* costs, i.e., costs that satisfy the triangle inequality (or its bipartite counterpart). Metric costs are intuitive for matching instances in which the costs are determined by distances, but we argue that they are present also in matching instances with more complex cost functions, e.g., online dating platforms — refer to the full version of this paper for a comprehensive explanation that also addresses the role that PoS plays in these matching scenarios.

The main result of this paper is an asymptotically tight tradeoff between the parameter α and the PoS considering α-stable matchings: the PoS is roughly $n^{0.58}$ when $\alpha = 1$ and it decreases exponentially as α increases. Since this tradeoff is realized by a simple poly-time algorithm, the designers of a matching system can now efficiently tune the parameter α to balance between the stability and the total cost of their system's output.

1.1 Related Work

Studying the impact of selfish players has been a major theoretical computer science success story in the last decade (see, e.g., the 2012 *Gödel Prize* [28,31,38]). In particular, much effort has been invested in quantifying how the efficiency of a system degrades due to selfishness of its players. The most notable notions in this context are the *Price of Anarchy (PoA)* [28, 32] and the *Price of Stability (PoS)* [6, 39], comparing the best possible outcome to the outcome of the worst (PoA) or best (PoS) solution with selfish players. Since their introduction, the Price of Anarchy and the Price of Stability have been extensively analyzed in diverse settings such as selfish routing [6, 9, 12, 13, 37, 38, 40], network formation games [3,4,7,11,41], job scheduling [10,14,27,28], and resource allocation [25,36]. While selfish players are traditionally modeled using the *Nash equilibrium* solution concept, where no player can benefit from a unilateral deviation, in matching settings unilateral deviations are not natural. Instead, we want that no *two* unmatched players prefer each other over their current matching partners. This solution concept is generally known as the Gale-Shapley *stable matching* [18].

For the most part, the stable matching realm has been subdivided into two versions: the *marriage* (bipartite) version, where the players are partitioned into *men* and *women* and each man (resp., woman) is equipped with a list of preferences over the set of women (resp., men); and the *roommates* (all-pairs) version, where each player is equipped with a list of preferences over all other players. Gale and Shapley showed that in the bipartite version, a stable matching always exists, and in fact, can be computed by a simple poly-time algorithm. In contrast, the all-pairs version does not necessarily have a solution. Both versions of the stable matching problem and their manifold variants (strictly/weakly ordered preferences, (in-)complete preference lists, (a-)symmetric preferences) admit an abundance of literature; see, e.g., the books of Knuth [26], Gusfield and Irving [19], Roth and Sotomayor [35], and Manlove [29]. The notion of stability studied in this paper has been coined as *weak stability* by Irving [22].

Sometimes, the players' preferences are associated with real costs so that each preference list is sorted in order of non-decreasing costs. This setting gives rise to the *minimum-cost stable matching* problem, where the goal is to construct a stable matching that minimizes the total cost of matched partners. Irving et al. [23] designed a poly-time algorithm for the bipartite (marriage) version of a special case of this problem, referred to as the *egalitarian stable matching* problem, where a cost of j is associated with each player for matching his/her jth preferred partner. Roth et al. [34] gave an LP-based solution to the problem. Irving's work was generalized by Feder [17] who presented a poly-time algorithm for the bipartite version of the general minimum-cost stable matching problem. Moreover, Feder also established the NP-hardness of the all-pairs (roommates) version and showed that it admits a 2-approximation algorithm.

The players' preferences in general stable matching scenarios exhibit no intrinsic correlations. Several approaches have been taken towards introducing consistency in the preference lists [21, 24, 26, 30]. Most relevant to the current paper is the approach of Arkin et al. [8] who studied the *geometric* stable roommate problem, where the players are identified with points in a Euclidean space and the preferences are given by the sorted distances to the other points. They showed that in the geometric setting, a stable matching always exists and that it is unique if the players' preferences exhibit no ties. These results easily generalize to arbitrary metric spaces. Arkin et al. also introduced the notion of an α*-stable* matching, which is central to the current paper.

There is an extensive literature on matching instances whose preferences are determined by the numerical attributes of the edges, interpreted as gains that should be maximized, rather than costs that should be minimized (cf. *correlated two-sided markets*) [1, 2, 5, 20]. Closely related to the goal of the current paper, Anshelevich et al. [5] establish tight tradeoffs between the matching stability parameter α and the PoA and PoS in the bipartite case under this gain maximization variant. In fact, the simple iterative algorithm presented in Sec. 4.1 is equivalent to the algorithm used in the proof of Theorem 2 in [5], but as it turns out, analyzing the quality of the resulting (α-stable) matching under the cost minimization variant studied in the current paper is much more demanding.

Reingold and Tarjan [33] proved that the approximation ratio of some greedy algorithm for minimum-cost perfect matching in metric graphs is $\Theta(n^{\log(3/2)})$ where $\log(3/2) \approx 0.58$.[3] It turns out that this result is equivalent to establishing the same bound for the PoA of minimum-cost perfect matching in such graphs. In the full version of this paper, we give a simpler proof for the PoA-result and extend their result to obtain a lower bound for the PoS for all $\alpha \geq 1$.

[3] In this paper, $\log x$ denotes the logarithm of x to the base of 2.

2 Setting and Preliminaries

Consider a graph G with vertex set $V(G)$ and edge set $E(G)$. Each edge $e \in E(G)$ is assigned a positive real *cost* $c(e)$. Unless stated otherwise, our graphs have $2n$ vertices, $n \in \mathbb{Z}_{>0}$, and are either complete ($|E(G)| = \binom{2n}{2}$) or complete bipartite ($V(G) = U_1 \cup U_2$, $|U_1| = |U_2| = n$ and $|E(G)| = n^2$). We say that the complete graph G is *metric* if $c(x,y) \leq c(x,z)+c(z,y)$ for every $x, y, z \in V(G)$; we say that the complete bipartite graph G is *metric* if $c(x,y) \leq c(x,z) + c(z,z') + c(z',y)$ for every $x, y, z, z' \in V(G)$, where x, z' and y, z are on opposite sides of G. For an arbitrary graph G, the *distance* $\text{dist}_G(x,y)$ of two vertices x and y of G is defined as the weighted length of the shortest path between x and y in G.

A *matching* is a subset $M \subseteq E(G)$ of the edges such that every vertex in $V(G)$ is incident to at most one edge in M. A matching is called *perfect* if every vertex in $V(G)$ is incident to exactly one edge in M, which implies that $|M| = n$ as $|V(G)| = 2n$. For a perfect matching M and a vertex $x \in V(G)$, we denote by $M(x)$ the unique vertex $y \in V(G)$ such that $(x,y) \in M$. Unless stated otherwise, all matchings mentioned hereafter are assumed to be perfect. (Perfect matchings clearly exist in a complete graph with an even number of vertices and in a complete balanced bipartite graph.) Given an edge subset $F \subseteq E(G)$, we define the *cost* of F as the total cost of all edges in F, denoted by $c(F) = \sum_{e \in F} c(e)$; in particular, the cost of a matching is the sum of its edge costs.

Definition (α-Stable Matching). *Consider some (perfect) matching $M \subseteq E(G)$ and some real number $\alpha \geq 1$. An edge $(u,v) \notin M$ is called* α-unstable *(a.k.a.* α-blocking*) with respect to M if $\alpha \cdot c(u,v) < \min\{c(u, M(u)), c(v, M(v))\}$. Otherwise, the edge is called* α-stable. *A matching M is called* α-stable *if it does not admit any α-unstable edge. We will omit α and call edges as well as matchings just* stable *or* unstable *whenever α is clear from the context or the argumentation holds for every choice of α.*

Let M^* denote a certain (perfect) matching M that minimizes $c(M)$. For simplicity, in what follows, we restrict our attention to complete (rather than complete bipartite) metric graphs, although all our results hold also for the complete bipartite case (following essentially the same lines of arguments).

Definition (α-Price of Stability). *The* α-Price of Stability *of G, denoted by $PoS_\alpha(G)$, is defined as $PoS_\alpha(G) = \min\{c(M)/c(M^*) : M \text{ is } \alpha\text{-stable matching}\}$. Furthermore, $PoS_\alpha(2n) = \sup\{PoS_\alpha(G) : G \text{ is metric}, |V(G)| = 2n\}$. Unless stated otherwise, when the parameter α is omitted, we refer to the case $\alpha = 1$.*

Definition (Price of Anarchy). *The* Price of Anarchy *of a graph G, denoted by $PoA(G)$, is defined as $PoA(G) = \max\{c(M)/c(M^*) : M \text{ is stable matching}\}$. Furthermore, $PoA(2n) = \sup\{PoA(G) : G \text{ is metric}, |V(G)| = 2n\}$.*

Note that since any stable matching by definition is also α-stable for any $\alpha \geq 1$, the Price of Anarchy does not improve by considering α-stability and hence its definition does not include the parameter α.

3 Price of Anarchy

The following theorem was implicitly proven by Reingold and Tarjan [33] in 1981. They showed that for minimum-cost perfect matching in metric graphs, the approximation ratio of the algorithm that picks edges by ascending costs is $\Theta(n^{\log(3/2)})$. Since the matching returned by this greedy algorithm is stable and since every stable matching can be obtained from the algorithm by an appropriate tie-breaking policy, it follows that the PoA of minimum-cost perfect matching in such graphs is also $\Theta(n^{\log(3/2)})$. A simpler and more intuitive proof for Reingold and Tarjan's 30 years old result is given in the full version of this paper.

Theorem 1. *The PoA of minimum-cost perfect matching in metric graphs with* $2n$ *vertices is* $\Theta(n^{\log(3/2)})$.

4 Price of Stability

The upper bound established on the PoA in Sec. 3 clearly holds for the PoS too. In the full version of this paper, we show that the proof technique for the $\Omega(n^{\log(3/2)})$-lower bound of Sec. 3 can be easily adapted to establish the same lower bound for the PoS as well. In fact, we generalize this result, showing that $\mathrm{PoS}_\alpha(2n) = \Omega\left(n^{\log(1+1/(2\alpha))}\right)$ for every $\alpha \geq 1$. Consequently, we turn our attention to bounding $\mathrm{PoS}_\alpha(2n)$ from above, establishing the following asymptotically tight upper bound.

Theorem 2. *The* PoS_α *of minimum-cost perfect matching in metric graphs with* $2n$ *vertices is at most* $3 \cdot n^{\log(1+1/(2\alpha))}$.

The proof of Theorem 2 is constructive, relying on a simple algorithm presented in Sec. 4.1. Sec. 4.2 provides the analysis of this algorithm, showing that the returned matching indeed satisfies the bound. Full proofs missing from this section can be found in the full version of this paper.

4.1 An Algorithm for α-Stable Matchings

The following algorithm Stab transforms a minimum-cost matching M^* in a metric graph into an α-stable matching M.

Algorithm Stab: Start with the minimum-cost matching $M \leftarrow M^*$ and iterate over all edges of G by non-decreasing order of costs. If the edge (u, v) currently considered is α-unstable in the current matching M, replace the edges $(u, M(u))$ and $(v, M(v))$ in M by (u, v) and $(M(u), M(v))$ (this operation is called a *flip* of the edge (u, v)) and continue with the next edge. After having iterated over all edges, return M.

We assume that edge cost ties are resolved in an arbitrary but consistent manner. In the following, we denote by M_i the matching calculated by the above algorithm at the end of iteration i. Moreover, $M_0 = M^*$ is the initial minimum-cost matching and M_S the final matching returned by Stab.

Lemma 3. *For any unstable edge b created by the flip of an edge e, we have* $c(b) > c(e)$.

Corollary 4 follows by induction on i. Lemma 5 then follows by a straightforward analysis of the algorithm's run-time.

Corollary 4. *Let* e_i *be the edge considered in iteration* i*. Then for any unstable edge* b *in* M_i *it holds that either* $c(e_i) < c(b)$ *or* b *will be considered in a later iteration* $j > i$.

Lemma 5. *Algorithm* Stab *transforms a minimum-cost matching into a valid* α*-stable matching in time* $\mathcal{O}(n^2 \log n)$.

4.2 Cost Analysis

Our goal in this section is to show that when Stab is invoked with parameter α for any $\alpha \geq 1$, it returns an α-stable matching M_S satisfying $c(M_S) = c(M^*) \cdot \mathcal{O}(n^{\log(1+1/(2\alpha))})$. Since this section makes heavy use of rooted binary trees and their properties, we require a few definitions. In a *full binary tree*, each inner node has exactly two children. The *depth* $d(v)$ of a node v in a tree T is the length of the unique path from the root of T to v and the *height* $h(T)$ of a tree T is defined as the maximal depth of any node in T. The *height* $h(v)$ of a node v of T is defined to be the height of its subtree. The *leaf set* $\mathcal{L}(T)$ or $\mathcal{L}(F)$ of a tree T or a collection F of trees is the set of all leaves in T or F, resp. The *leaf set* $\mathcal{L}(v)$ of a node v in a tree is $\mathcal{L}(T_v)$ where T_v is the subtree rooted at v. Finally, two nodes with the same parent are called *sibling nodes*. We begin with Lemma 6 stating an important property of the edges that are flipped by Stab.

Lemma 6. *If an edge* e *is flipped in iteration* i*, then* $e \in M_j$ *for all* $j \geq i$ *and, in particular,* $e \in M_S$.

Consider an iteration of Stab where edge (u, v) is flipped because it was unstable at the beginning of the iteration. Then the two edges $(u, M(u))$ and $(v, M(v))$ are replaced by (u, v) and $(M(u), M(v))$. Since the edge (u, v) is selected irrevocably according to Lemma 6, the edges $(u, M(u))$ and $(v, M(v))$ can never be part of M again. The only edge, of the four edges involved, that may be changed again, is the edge $(M(u), M(v))$. Thus, we refer to $(M(u), M(v))$ as an *active* edge. We also refer to all edges in M_0 as active. Using the notion of active edges, we shall now model the changes that Stab applies to the matching during its execution through a logical helper structure called the *flip forest*. To avoid confusion between the basic elements of our graphs and the basic elements of the flip forest, we refer to the former as vertices/edges and to the latter as nodes/links.

Definition (Flip Forest). *The* flip forest $F = (U, K)$ *for a certain execution of* Stab *is a collection of disjoint rooted trees and has node set* U *and link set* K*. For each edge* $e \in V \times V$ *that has been active at some stage during the execution,* *there exists a node $u_e \in U$. This correspondence is denoted by $u_e \sim e$. For each flip of an edge (u, v) in G, resulting in the removal of the edges $(u, M(u))$ and $(v, M(v))$ from M, K contains a link connecting the node $y \sim (u, M(u))$ to its parent $x \sim (M(u), M(v))$ and a link connecting the node $z \sim (v, M(v))$ to its parent $x \sim (M(u), M(v))$. (Observe that, by definition, all three edges $(u, M(u))$, $(v, M(v))$, and $(M(u), M(v))$ are active.) Refer to Fig. 1 for an illustration.*[4]

The definition of a flip forest ensures that for each flip of the algorithm, we obtain a binary *flip tree segment*. When we transcribe each flip operation of the complete execution of Stab into a flip tree segment as explained above, we end up with a collection of full binary trees — the *flip forest*. This is because the parent node of a tree segment may appear as a child node of the tree segment corresponding to a later iteration of the algorithm since its corresponding edge is still active and therefore may participate in another flip. Each such tree is called a *flip tree* hereafter. Figure 2 illustrates a sample execution of Stab.

Observe that all leaves (including isolated nodes) in the flip forest correspond to edges in the minimum-cost matching $M_0 = M^*$. The edges in the matching M_S are implicitly represented by the flip forest: An edge that gets flipped — and is therefore irrevocably selected into M_S — has no corresponding node in F, but we may associate it with the node corresponding to the active edge resulting from the flip. On top of these edges, M_S contains the edges corresponding to the roots of the trees in the flip forest.

We now define a function $\psi : U \mapsto \mathbb{R}$ that maps a real *weight* to each node in the flip forest F as follows. For each leaf ℓ of a flip tree in F, we set $\psi(\ell) \coloneqq c(e)$, where $\ell \sim e$ and we recall that an edge corresponding to a leaf node in F is part of M^*. The function ψ is extended to an inner node x of a flip tree with child nodes y and z by the recursion

$$\psi(x) \coloneqq \psi(y) + \psi(z) + (1/\alpha) \cdot \min\{\psi(y), \psi(z)\} \ . \tag{1}$$

For ease of notation, we call the child with smaller (resp., larger) weight as well as the link leading to its parent *light* (resp., *heavy*); ties are resolved arbitrarily. We denote the light child of a node x as x_L and the heavy child as x_H. Then we can rewrite Eq. (1) as $\psi(x) \coloneqq \psi(x_H) + (1 + 1/\alpha) \cdot \psi(x_L)$.

Lemma 7. *Let x be a node in F and e an edge in G with $x \sim e$. Then $c(e) \leq \psi(x)$.*

At this stage, we would like to relate the weight $\psi(r_T)$ of the roots r_T in F to the cost of the stable matching M_S returned by Stab. To that end, we observe that M_S consists of the edges corresponding to the roots in F and to the edges that have been flipped along the course of the execution; let R and D denote the set of the former and latter edges, respectively. Observe that

$$c(M_S) = \sum_{e \in R} c(e) + \sum_{e \in D} c(e) \ .$$

[4] All figures are deferred to the full version of this paper.

Consider the flip of the edge (u, v) resulting in the insertion of the edge $(M(u), M(v)) \sim x$ to M and the removal of the edges $(u, M(u)) \sim x_L$ and $(v, M(v)) \sim x_H$ from M. Since $\psi(x) = \psi(x_H) + (1 + 1/\alpha) \cdot \psi(x_L)$, we have $\psi(x) - (\psi(x_L) + \psi(x_H)) = \psi(x_L)/\alpha$. Lemma 7 then implies that $\psi(x) - (\psi(x_L) + \psi(x_H)) \geq c(u, M(u))/\alpha$, and since edge (u, v) was flipped, we have $\psi(x) - (\psi(x_L) + \psi(x_H)) \geq c(u, v)$. Therefore,

$$\begin{aligned}
\sum_{e \in D} c(e) &\leq \sum_{\text{internal } x \in U} (\psi(x) - (\psi(x_L) + \psi(x_H))) \\
&= \sum_{\text{flip trees T}} \Big(\psi(r_T) \ - \sum_{\ell \in \mathcal{L}(T)} \psi(\ell) \Big) \\
&= \sum_{\text{flip trees T}} \psi(r_T) \ - \sum_{\ell \in \mathcal{L}(F)} \psi(\ell) \ ,
\end{aligned}$$

where the second equation holds by a telescoping argument. Note further that $\sum_{e \in R} c(e) \leq \sum_{\text{flip trees} T} \psi(r_T)$ and thus

$$c(M_S) \leq 2 \sum_{\text{flip trees T}} \psi(r_T) \ - \sum_{\ell \in \mathcal{L}(F)} \psi(\ell) \ .$$

Since $c(M^*) = \sum_{\ell \in \mathcal{L}(F)} \psi(\ell)$, Corollary 8 follows.

Corollary 8. *The matching M_S returned by* Stab *satisfies*

$$c(M_S) \leq 2 \sum_{\text{flip trees } T} \psi(r_T) - c(M^*) \ .$$

We will now have a closer look at the properties of our flip trees and their weights. It will be convenient to ignore the relation of the flip trees to the Stab algorithm at this stage; in other words, we consider an abstract full binary tree T with a *leaf weight function* $w : \mathcal{L}(T) \to \mathbb{R}_{\geq 0}$. For any leaf ℓ of T, we set $\psi(\ell) = w(\ell)$ and determine the weight $\psi(x)$ of each inner node x in T following the recursion given by Eq. (1). Note that we allow our tree T to have zero-weight leaves now (this can only make our analysis more general).

Definition (Complete Binary Tree). *A full binary tree T is called* complete *if all leaves are at depth $h(T)$ or $h(T) - 1$. Given some positive integer n that will typically be the number of leaves in some tree, let $h(n) = \lceil \log n \rceil$ and $k(n) = 2^{h(n)} - n$. Note that $0 \leq k(n) < 2^{h(n)-1}$.*

Observe that for a complete full binary T with n leaves, $h(n)$ is the height $h(T)$ of T while $k(n)$ equals the number of missing leaves at the maximum depth $h(T)$.

Definition (ψ-Balanced Binary Tree). *A full binary tree T is called ψ-*balanced *if for any two sibling nodes x, y in T, we have $\psi(x) = \psi(y)$.*

Consider a full binary tree T. Let $\Lambda(T)$ denote the sum of the weights of the leaves of T, i.e., $\Lambda(T) = \sum_{\ell \in \mathcal{L}(T)} w(\ell) = \sum_{\ell \in \mathcal{L}(T)} \psi(\ell)$, and let $\Psi(T) = \psi(r_T)$ (recall that r_T denotes the root of T). The following observation is established by induction on the node depth.

Observation 9. *For any node v of a ψ-balanced full binary tree T, we have $\psi(v) = (2 + 1/\alpha)^{-d(v)} \cdot \Psi(T)$.*

Definition (Effect of a Flip Tree). *The* effect $\eta(T)$ *of a full binary tree T is defined as*

$$\eta(T) \;=\; \begin{cases} \Psi(T)/\Lambda(T) & \text{if } \Lambda(T) > 0 \\ 1 & \text{if } \Lambda(T) = 0 \end{cases} .$$

An n-leaf full binary tree T is said to be effective *if it maximizes $\eta(T)$, i.e., if there does not exist any n-leaf full binary tree T' such that $\eta(T') > \eta(T)$.*

Intuitively speaking, if we think of T as a flip tree, then its effect is a measure for the factor by which the flips represented by T increase the cost of M^* when applied to it. But, once again, we do not restrict our attention to flip trees at this stage. The effect of a full binary tree is essentially determined by its topology and by the assignment of weights to its leaves. It is important to point out that the effect of a flip tree is invariant to scaling its leaf weights (see full version of this paper). Our upper bound is established by showing that the effect of an effective n-leaf full binary tree is $\mathcal{O}\big(n^{\log(1+1/(2\alpha))}\big)$. We begin by developing a better understanding of the topology of effective ψ-balanced full binary trees.

Lemma 10. *An effective n-leaf ψ-balanced full binary tree must be complete.*

Proof (sketch). Aiming for a contradiction, suppose that T is not complete. Let z be an internal node at depth d with leaf children x, y (whose depth is $d + 1$) and let z' be a leaf at depth $d' < d$. Let T' be the full binary tree obtained from T by deleting x and y and inserting two new leaves x', y' as children of z'. Let w and w' be the leaf weigh functions of T and T', respectively, defined by requiring that T and T' are ψ-balanced and scaled so that $\Psi(T) = \Psi(T') = 1$; this is well defined since by Observation 9, the ψ-values of all nodes in T and T' (and in particular, the leaf weight functions w and w') are fully determined by their topology and the values of $\Psi(T)$ and $\Psi(T')$ (in a top-down fashion).

We establish the proof by arguing that $\Lambda(T') < \Lambda(T)$ which implies $\eta(T') > \eta(T)$, in contradiction to T being effective. To that end, notice that the construction of T' implies $\Lambda(T') = \Lambda(T) + w'(x') + w'(y') + w'(z) - (w(x) + w(y) + w(z'))$. The assertion follows from Observation 9 by a direct calculation. □

Next, we develop a closed-form expression for the effect of complete ψ-balanced full binary trees. We define the function $\varphi : \mathbb{Z}_{>0} \mapsto \mathbb{R}$ as

$$\varphi(n) := \frac{(2 + 1/\alpha)^{h(n)}}{2^{h(n)} + k(n)/\alpha}$$

and recall that $h(n) = \lceil \log n \rceil$ and $k(n) = 2^{h(n)} - n$. Lemma 11 follows from Observation 9 by direct calculation and Lemma 12 follows from φ's definition.

Lemma 11. *The effect of an n-leaf complete ψ-balanced full binary tree T is $\eta(T) = \varphi(n)$.*

Lemma 12. *The function $\varphi(n)$ is strictly increasing.*

Now, we can show that it is sufficient to consider complete ψ-balanced full binary trees.

Lemma 13. *An effective n-leaf full binary tree must be ψ-balanced.*

Proof. We prove the statement by induction on the number of leaves n. The base case of a tree having a single leaf (which is also the root) holds vacuously; the base case of a tree having two leaves is trivial. Assume that the assertion holds for trees with fewer than n leaves and let T be an effective n-leaf full binary tree. Let T_ℓ and T_r be the left and right subtrees of T and let z be the number of leaves in T_ℓ where $1 \leq z \leq n-1$.

Observe that both T_ℓ and T_r have to be effective as otherwise, $\eta(T)$ could be increased. More precisely, if $T_i \in \{T_\ell, T_r\}$ is not effective, then there exists a full binary tree T_i' with the same number of leaves as T_i (either z or $n-z$) such that $\eta(T_i') > \eta(T_i)$; by replacing T_i with T_i' in T and scaling $\Lambda(T_i')$ so that $\Lambda(T_i') = \Lambda(T_i)$, we increase $\Psi(T)$ without affecting $\Lambda(T)$, thus increasing $\eta(T)$, in contradiction to T being effective. By the inductive hypothesis, both T_ℓ and T_r are ψ-balanced, hence Lemma 10 guarantees that both are complete. This allows us to use Lemma 11 to determine the effects of T_ℓ and T_r as $\varphi(z)$ and $\varphi(n-z)$, respectively.

Assume without loss of generality that the leaf weights are scaled such that $\Lambda(T) = \Lambda(T_\ell) + \Lambda(T_r) = 1$ and set $\Lambda(T_\ell) = x$, $\Lambda(T_r) = 1 - x$, for some $0 \leq x \leq 1$. We consider a set of $n-1$ functions $f_z : [0,1] \mapsto \mathbb{R}_{>0}$ (parametrized by $1 \leq z \leq n-1$) with

$$f_z(x) = \begin{cases} \varphi(z) \cdot x + (1 + 1/\alpha)\varphi(n-z) \cdot (1-x) & \text{if } \varphi(z)x \geq \varphi(n-z)(1-x) \\ (1 + 1/\alpha)\varphi(z) \cdot x + \varphi(n-z) \cdot (1-x) & \text{if } \varphi(z)x \leq \varphi(n-z)(1-x) \end{cases}$$

that, by Lemma 11, determine the effect of T given that T_ℓ has $1 \leq z \leq n-1$ leaves and $\Lambda(T_\ell) = x \in [0,1]$. Observe that each f_z is a piecewise linear continuous function, linear in the intervals $[0, b_z]$ and $[b_z, 1]$, where b_z is the *break point* of f_z satisfying $\varphi(z)b_z = \varphi(n-z)(1-b_z)$. Hence, f_z must attain its maximum either at a boundary point 0 or 1, or at the break point b_z, where the latter case corresponds to a ψ-balanced tree.

Consider the function $f(x) = \max_z f_z(x)$ whose maximum corresponds to the effect of an effective n-leaf full binary tree and let $\hat{x} \in \operatorname{argmax}_{x \in [0,1]} f(x)$. We argue that $\hat{x}$ can be neither 0 nor 1. Indeed, if $\hat{x} = 0$, then $\Lambda(T) = \Lambda(T_r)$ and $\Psi(T) = \Psi(T_r)$, hence $\eta(T) = \eta(T_r)$ for the corresponding tree T. But since T_r has fewer leaves than T and is complete and ψ-balanced, Lemmas 11 and 12 dictate that its effect — and thus also the effect of T — must be smaller than the effect of an n-leaf complete ψ-balanced full binary tree, a contradiction to the choice of $\hat{x}$ maximizing $f(x)$. An analogous argument excludes $\hat{x} = 1$. It follows that the maximum of $f(x)$ must be attained at a point $0 < \hat{x} < 1$, which, by the definition of f, is the break point b_z of some function f_z and thus realized by a ψ-balanced tree. □

Combining Lemmas 10, 11, and 13 and recalling that $h = h(n) = \lceil \log n \rceil \leq \log n + 1$ and $k = k(n) \geq 0$, we conclude that the effect of an n-leaf full binary tree is at most

$$\frac{(2+1/\alpha)^h}{2^h + k/\alpha} \;\leq\; \frac{(2+1/\alpha)^h}{2^h} \;\leq\; (1+1/(2\alpha))^{\log n+1} \;\leq\; 3/2 \cdot n^{\log(1+1/(2\alpha))} \;.$$

Returning to the definition of the flip forest F, we recall that there exists one leaf in F for each of the n edges in the minimum-cost matching M^* and therefore each flip tree has at most n leaves. Furthermore, since

$$c(M^*) = \sum_{\text{flip trees } T} \sum_{\ell \in \mathcal{L}(T)} \psi(\ell) = \sum_{\text{flip trees } T} \Lambda(T) \;,$$

we can employ Corollary 8 to derive

$$\frac{c(M_S)}{c(M^*)} \;\leq\; 2 \cdot \frac{\sum_{\text{flip trees } T} \Psi(T)}{\sum_{\text{flip trees } T} \Lambda(T)} \;\leq\; 2 \cdot \max_{\text{flip trees } T} \eta(T) \;\leq\; 3 \cdot n^{\log(1+1/(2\alpha))} \,,$$

thus establishing Theorem 2.

References

1. Abraham, D., Levavi, A., Manlove, D., O'Malley, G.: The stable roommates problem with globally-ranked pairs. In: WINE'07
2. Ackermann, H., Goldberg, P.W., Mirrokni, V.S., Röglin, H., Vöcking, B.: Uncoordinated two-sided matching markets. SICOMP'11
3. Albers, S., Eilts, S., Even-Dar, E., Mansour, Y., Roditty, L.: On nash equilibria for a network creation game. In: SODA'06
4. Andelman, N., Feldman, M., Mansour, Y.: Strong price of anarchy. In: SODA'07
5. Anshelevich, E., Das, S., Naamad, Y.: Anarchy, stability, and utopia: creating better matchings. AAMAS'13
6. Anshelevich, E., Dasgupta, A., Kleinberg, J.M., Tardos, É., Wexler, T., Roughgarden, T.: The price of stability for network design with fair cost allocation. SICOMP'08
7. Anshelevich, E., Dasgupta, A., Tardos, E., Wexler, T.: Near-optimal network design with selfish agents. In: STOC'03
8. Arkin, E.M., Bae, S.W., Efrat, A., Okamoto, K., Mitchell, J.S.B., Polishchuk, V.: Geometric stable roommates. IPL'09
9. Awerbuch, B., Azar, Y., Epstein, A.: The price of routing unsplittable flow. In: STOC'05
10. Awerbuch, B., Azar, Y., Richter, Y., Tsur, D.: Tradeoffs in worst-case equilibria. TCS'06
11. Chen, H.L., Roughgarden, T.: Network design with weighted players. In: SPAA'06
12. Christodoulou, G., Koutsoupias, E.: On the price of anarchy and stability of correlated equilibria of linear congestion games. In: ESA'05

13. Christodoulou, G., Koutsoupias, E.: The price of anarchy of finite congestion games. In: STOC'05
14. Czumaj, A., Vöcking, B.: Tight bounds for worst-case equilibria. In: SODA'02
15. Edmonds, J.: Paths, trees, and flowers. Canadian J. of Math. 1965
16. Edmonds, J.: Maximum matching and a polyhedron with $0,1$ vertices. J. of Research of the National Bureau of Standards 1965
17. Feder, T.: A new fixed point approach for stable networks and stable marriages. JCSS'92
18. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly 1962
19. Gusfield, D., Irving, R.W.: The stable marriage problem: structure and algorithms. MIT Press, Cambridge, MA, USA (1989)
20. Hoefer, M., Wagner, L.: Designing profit shares in matching and coalition formation games. In: WINE'13
21. Huang, C.C.: Two's company, three's a crowd: Stable family and threesome roommates problems. In: ESA'07.
22. Irving, R.W.: Stable marriage and indifference. DAM'94
23. Irving, R.W., Leather, P., Gusfield, D.: An efficient algorithm for the "optimal" stable marriage. JACM'87
24. Irving, R.W., Manlove, D.F., Scott, S.: The stable marriage problem with master preference lists. DAM'08
25. Johari, R., Tsitsiklis, J.N.: Efficiency loss in a network resource allocation game. MOR'04
26. Knuth, D.E.: Marriages stables et leurs relations avec d'autres problèmes combinatoires. Les Presses de l'Université de Montréal (1976)
27. Koutsoupias, E., Mavronicolas, M., Spirakis, P.G.: Approximate equilibria and ball fusion. TOCS'03
28. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. Computer Science Review 2009
29. Manlove, D.F.: Algorithmics of matching under preferences. World Scientific (2014)
30. Ng, C., Hirschberg, D.S.: Three-dimensional stable matching problems. SIDMA'91
31. Nisan, N., Ronen, A.: Algorithmic mechanism design. GEB'01
32. Papadimitriou, C.: Algorithms, games, and the internet. In: STOC'01
33. Reingold, E.M., Tarjan, R.E.: On a greedy heuristic for complete matching. SICOMP'81
34. Roth, A.E., Rothblum, U.G., Vande Vate, J.H.: Stable matchings, optimal assignments, and linear programming. MOR'93
35. Roth, A.E., Sotomayor, M.A.O.: Two-sided matching: a study in game-theoretic modeling and analysis. Cambridge University Press (1990)
36. Roughgarden, T.: Potential functions and the inefficiency of equilibria. In: ICM'06
37. Roughgarden, T.: The price of anarchy is independent of the network topology. JCSS'03
38. Roughgarden, T., Tardos, E.: How bad is selfish routing? JACM'02
39. Schulz, A.S., Stier Moses, N.E.: On the performance of user equilibria in traffic networks. In: SODA'03
40. Suri, S., Tóth, C.D., Zhou, Y.: Selfish load balancing and atomic congestion games. Algorithmica 2007
41. Vetta, A.: Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In: FOCS'02

Appendix

A Online Dating

Consider a (heterosexual) dating platform with the goal to help a set W of women and a set M of men finding suitable partners and let us assume that $|W| = |M|$. When signing up for the platform, a woman $w \in W$ (resp., a man $m \in M$) has to complete a questionnaire about a set of her (his) own characteristics including absolute numerical properties such as age, height, or weight; relative numerical properties such as sportiness, empathy, or ambition; and combinations thereof that capture more complex aspects such as movie taste. This questionnaire generates a point $v(w)$ ($v(m)$) in the real vector space V_W (V_M), where each characteristic is represented by one or more dimensions. Then, she (he) completes a different questionnaire describing the characteristics of her ideal male partner w^* (his ideal female partner m^*), which in turn generates a point $v(w^*)$ in V_M ($v(m^*)$ in V_W). Notice that V_M and V_W may correspond to different sets of characteristics (after all, men and women might be interested in different qualities of the respective opposite sex).

We determine the valuation of woman w for man m by $\|v(m) - v(w^*)\|_M$, where $\|\cdot\|_M$ is some norm on V_M. (The same goes with the opposite sexes with respect to some norm $\|\cdot\|_W$ on V_W.) Presuming that one would rather not be in a relationship with a partner that is not attracted to oneself, it is natural to measure the cost of a match between a woman $w \in W$ and a man $m \in M$ in terms of

$$c(w, m) = \max\{\|v(m) - v(w^*)\|_M,\ \|v(w) - v(m^*)\|_W\}$$

with the reasoning that "a chain (a relationship in this case) is only as strong as its weakest link". Observe that by definition, the symmetric cost function $c(\cdot, \cdot)$ obeys the bipartite counterpart of the triangle inequality. Refer to Figure 3 for an example with two-dimensional vector spaces V_W and V_M.

The popularity of a dating platform depends significantly on the trust that its users have in the platform finding them a suitable partner. This trust can be boosted by providing rigid guarantees for the matching established by the platform. Two such natural guarantees are that the total matching cost is minimized, which means that on average, the partners in a matched pair have optimal valuations for each other; and that the matching is stable, which means that no unmatched pair has an incentive to deviate from the matching recommended by the platform. However, it turns out that these two guarantees cannot coexist, hence we allow for an approximation of the minimum cost (perfect) matching and relax the notion of stability. The PoS then tells us how well the dating platform can do in terms of the total matching cost compared to a benchmark which is not subject to the (relaxed) stability constraint.

The problem described above corresponds to matchings in complete bipartite graphs (the marriage version). If, instead, we consider a same-sex dating platform, we end up with points in a single vector space and a cost function defined over all user pairs which corresponds to matchings in complete graphs (the roommates version).

B Price of Anarchy

In this section, we present a simpler and more intuitive proof for Reingold and Tarjan's 30-year-old result cast in Theorem 1, which relies on a series of elementary reductions, essentially showing that PoA($2n$) is realized by weighted line graphs, i.e., metric graphs that can be embedded isometrically into the real line. Following that, we introduce a family of weighted line graphs with PoA of $\Theta(n^{\log(3/2)})$ and show that no other weighted line graph admits higher PoA.

Definition (Matching Configuration). *A* matching configuration *(MC)* $\xi = (G, M^*, M)$ *consists of a metric graph G, a minimum-cost matching M^*, and a stable matching M on G. The* ratio *of ξ is defined as $\rho(\xi) := c(M)/c(M^*)$.*

Observe that the definition of a MC ξ induces a collection $\mathcal{A}(\xi)$ of alternating cycles in the symmetric difference $M \oplus M^*$, where an alternating cycle is a cycle whose edges are alternatingly from M and M^*. The cycles in $\mathcal{A}(\xi)$ are referred to hereafter as the alternating cycles *exhibited* by ξ. We say that ξ is *spanned* by the cycles in $\mathcal{A}(\xi)$ if each vertex of G belongs to an alternating cycle in $\mathcal{A}(\xi)$. Clearly, graphs with two vertices admit a single (perfect) matching, hence PoA(2) = 1, so in what follows, it suffices to consider MCs on $2n$ vertices for $n > 1$. The following lemma states that it also suffices to consider MCs spanned by a single alternating cycle.

Lemma 14. *For every MC $\xi = (G, M^*, M)$ on $2n$ vertices, there exists a MC $\hat{\xi}$ on $2n'$ vertices, $1 < n' \leq n$, spanned by a single alternating cycle such that $\rho(\hat{\xi}) \geq \rho(\xi)$.*

Proof. Since $\mathcal{A}(\xi) = \emptyset$ implies $\rho(\xi) = 1$, we may assume hereafter that $|\mathcal{A}(\xi)| \geq 1$. So let A be an alternating cycle in $\mathcal{A}(\xi)$ maximizing the ratio $c(M_A)/c(M^*_A)$, where M_A and M^*_A are the matchings M^* and M, resp., restricted to the edges of A. Let G_A be the subgraph of G induced by $V(A)$ and take $\hat{\xi} = (G_A, M^*_A, M_A)$. Observe that $\hat{\xi}$ is a valid MC, since M^*_A and M_A are still a minimum-cost matching and a stable matching, resp., in G_A. By the choice of A, it follows that $\rho(\hat{\xi}) \geq \rho(\xi)$. □

In the following, we will say that the edge costs in a graph $G = (V, E)$ *agree* with the distances in a subgraph $G' = (V, E')$ on the same vertices, if and only if for any edge (x, y) in G we have $c(x, y) = \text{dist}_{G'}(x, y)$.

Definition (Weighted Cycle MC). *A MC $\xi = (G, M^*, M)$ is said to be a* weighted cycle MC *if ξ is spanned by a single alternating cycle A and the edge costs in G agree with the distances in the subgraph of G induced by the edges in $E(A)$.*

Our next lemma states that it suffices to bound the PoA in weighted cycle MCs.

Lemma 15. *For every MC $\xi = (G, M^*, M)$ on $2n$ vertices that is spanned by a single alternating cycle, there exists a weighted cycle MC $\hat{\xi}$ on $2n$ vertices such that $\rho(\hat{\xi}) \geq \rho(\xi)$.*

Proof. Let A be the single alternating cycle spanning ξ. If ξ is not a weighted cycle MC, then G must admit a *shortcut* — an edge $(x, y) \in E(G) \setminus E(A)$ satisfying $c(x, y) < \mathrm{dist}_A(x, y)$, where $\mathrm{dist}_A(x, y)$ denotes the distance between x and y in the (weighted) cycle A. Let (x, y) be a shortcut minimizing $c(x, y)$ and let $z \in V(G) \setminus \{x, y\}$ be the vertex minimizing $c(x, z) + c(z, y)$. Observe that $c(x, y)$ must be strictly smaller than $c(x, z) + c(z, y)$ as (x, y) is a shortcut of G and G does not admit any shorter shortcut. We argue that the cost of (x, y) can be increased to $c(x, z) + c(z, y)$ without violating the validity of ξ as a MC. As there are only finitely many shortcuts, the assertion follows since repeating this step (finitely many times) removes all the shortcuts of G. To that end, note that after increasing $c(x, y)$ to $c(x, z) + c(z, y)$, M^* remains a minimum-cost matching of G (we only increased the cost of some edge not in M^*) and M remains a stable matching of G (we only increased the cost of some edge not in M). So, all we have to show is that G remains metric, which follows from the choice of z. □

Definition (Weighted Line MC). *We say that a $(2n)$-vertex metric graph G is a* weighted line graph *if it can be isometrically embedded into the real line. As such, it is convenient to identify the vertices of G with the reals $x_1 < \cdots < x_{2n}$ so that $c(x_i, x_j) = x_j - x_i$ for every $1 \leq i < j \leq 2n$. In some cases, it will also be convenient to define a weighted line graph by setting all the differences $x_{i+1} - x_i$ without explicitly specifying the x_is themselves. A* weighted line MC *$\xi = (G, M^*, M)$ is a MC on $2n$ vertices satisfying:*

(1) G is a weighted line graph;
(2) $M^* = \{(x_{2i-1}, x_{2i}) \mid 1 \leq i \leq n\}$; and
(3) $M = \{(x_{2i}, x_{2i+1}) \mid 1 \leq i < n\} \cup \{(x_1, x_{2n})\}$.

Observe that ξ is spanned by a single alternating cycle $A = (x_1, \ldots, x_{2n}, x_1)$.

Note that requirement (2) in the definition is not really necessary: the requirement that G is a weighted line graph already implies that $\{(x_{2i-1}, x_{2i}) \mid 1 \leq i \leq n\}$ is the unique minimum-cost matching of G as every other matching M' contains some edge (x_i, x_j) such that $|j - i| > 1$; it is easy to show that such an edge must belong to an improving alternating cycle, hence M' cannot be optimal. Given a $(2n)$-vertex weighted line graph G, we shall subsequently denote this unique minimum-cost stable matching by $M^*(G)$ and the matching $\{(x_{2i}, x_{2i+1}) \mid 1 \leq i < n\} \cup \{(x_1, x_{2n})\}$ by $M(G)$. By definition, $\xi = (G, M^*(G), M(G))$ is a valid (weighted line) MC if and only if $M(G)$ is stable. Note also that a weighted line MC is a refinement of a weighted cycle MC, with the additional requirement that the cost of the longest edge in the unique alternating cycle A equals the total cost of all other edges of A. Building on this fact, the next lemma states that it suffices to consider weighted line MCs.

Lemma 16. *For every weighted cycle MC $\xi = (G, M^*, M)$ on $2n$ vertices, there exists a weighted line MC $\hat{\xi}$ on $2n$ vertices such that $\rho(\hat{\xi}) \geq \rho(\xi)$.*

Proof. Let A be the single alternating cycle spanning ξ and let e be an edge in M that maximizes $c(e)$. Let $W_{-e} = \sum_{e' \in E(A) \setminus \{e\}} c(e')$. Clearly, $c(e) \leq W_{-e}$, as otherwise, G is not metric. We argue that if $c(e) < W_{-e}$, then the cost of e can be increased to W_{-e} (while also adapting the cost of all edges in G whose cost is affected by e, bearing in mind that the edge costs in G have to agree with the distances in the subgraph induced by $E(A)$) without violating the validity of ξ as a MC; the assertion follows because this step turns ξ into a weighted line MC. To that end, note that after increasing $c(e)$ to W_{-e}, G remains metric (ξ is a weighted cycle MC) and M^* remains a minimum-cost matching (we only increased the cost of some edges not in M^*). So, all we have to show is that M remains stable, which follows from the choice of e. □

Once we restrict our attention to weighted line configurations, we can augment G with new vertices without significantly affecting the ratio of the MC.

Lemma 17. *For every weighted line MC $\xi = (G, M^*, M)$ on $2n$ vertices and for any $\varepsilon > 0$, there exists a weighted line MC $\hat{\xi}$ on $2(n+1)$ vertices such that $\rho(\hat{\xi}) \geq \rho(\xi) - \varepsilon$.*

Proof. Recall that the vertices of G are identified with the reals $x_1 < \ldots < x_{2n}$. Let $\hat{G}$ be the weighted line graph obtained from G by augmenting $V(G)$ with two new vertices identified with the reals $y = x_{2n} + \delta$ and $y' = y + \delta'$ for some $\delta' > \delta > 0$. The assertion follows since by taking a sufficiently small δ', we guarantee both that $M(\hat{G})$ is stable in $\hat{G}$ and that $c(M(\hat{G}))/c(M^*(\hat{G})) \geq \rho(\xi) - \varepsilon$. □

We now turn to present a family of metric graphs referred to as *Reingold-Tarjan graphs*, acknowledging Reingold and Tarjan's paper [33], where these graphs were first introduced.

Consider some integer $k > 0$. The k^{th} Reingold-Tarjan graph H^k is a weighted line graph whose 2^k vertices are identified with the reals $x_1^k < \cdots < x_{2^k}^k$. It is defined recursively: For $k = 1$, we set $x_2^1 - x_1^1 = 1$. Assume that H^k is already defined and let $D^k = x_{2^k}^k - x_1^k$ be its *diameter*. Then, H^{k+1} is defined by placing two disjoint instances of H^k on the real line with an S^{k+1} *spacing* between them, i.e., $x_{2^k+1}^{k+1} - x_{2^k}^{k+1} = S^{k+1}$, yielding $D^{k+1} = 2 \cdot D^k + S^{k+1}$. In the current[5] construction, we set $S^k = D^{k-1}$, thus the diameter of H^k satisfies $D^k = 3^{k-1}$. Refer to Fig. 4 for an illustration of the parametrized Reingold-Tarjan graph H_α^4 that will be used later. The original graph is obtained by setting the two parameters α and ϵ to 1 and 0, respectively.

Recall that $M^*(H^k)$ matches x_{2i-1}^k with x_{2i}^k for every $1 \leq i \leq 2^{k-1}$; since all these edges have cost 1, it follows that $c(M^*(H^k)) = 2^{k-1}$. Furthermore, we

[5] A parametrized variant of the Reingold-Tarjan graphs is presented in Appendix C, where we use a different value for S^k.

argue by induction on k that the matching

$$M(H^k) = \{(x^k_{2i}, x^k_{2i+1}) \mid 1 \leq i < 2^{k-1}\} \cup \{(x^k_1, x^k_{2^k})\}$$

is stable and that its cost is

$$c(M(H^k)) = D^k + (D^k - c(M^*)) = 2 \cdot 3^{k-1} - 2^{k-1} \ .$$

Therefore, $\xi^k_{RT} = (H^k, M^*(H^k), M(H^k))$, referred to as the k^{th} *Reingold-Tarjan MC* hereafter, is a valid weighted line MC with ratio

$$\rho(\xi^k_{RT}) \;=\; \frac{c(M(H^k))}{c(M^*(H^k))} \;=\; \frac{2 \cdot 3^{k-1} - 2^{k-1}}{2^{k-1}} \;=\; \Theta\big((3/2)^{k-1}\big) \;=\; \Theta\big(n^{\log(3/2)}\big) \ ,$$

where the last equation follows by setting $2n = 2^k$. Using Lemma 17 to extend the Reingold-Tarjan graphs for $2n \neq 2^k$, we immediately conclude that $\mathrm{PoA}(2n) = \Omega(n^{\log(3/2)})$, establishing the lower bound part of Theorem 1. The upper bound part of the theorem is established by combining Lemmas 14, 15, 16, and 17 with the following lemma.

Lemma 18. *The k^{th} Reingold-Tarjan MC ξ^k_{RT} satisfies the inequality $\rho(\xi^k_{RT}) \geq \rho(\xi)$ for any weighted line MC ξ on 2^k vertices.*

Proof. By induction on k. The assertion holds trivially for $k = 1$, so assume that it holds for k and consider an arbitrary weighted line MC $\xi = (G, M^*(G), M(G))$ on 2^{k+1} vertices identified with the reals $x_1 < \cdots < x_{2^{k+1}}$. Let L and R be the subgraphs of G induced by the vertices $x_1, \dots, x_{2^k}$ and $x_{2^k+1}, \dots, x_{2^{k+1}}$, resp. Let $e = (x_{2^k}, x_{2^k+1})$ and let $D_L = x_{2^k} - x_1$ and $D_R = x_{2^{k+1}} - x_{2^k+1}$. We refer to the vertices x_1 and x_{2^k} (resp., x_{2^k+1} and $x_{2^{k+1}}$) as the *external* vertices of L (resp., R) and to the vertices $x_2, \dots, x_{2^k-1}$ (resp., $x_{2^k+2}, \dots, x_{2^{k+1}-1}$) as the *internal* vertices of L (resp., R). Observe that $e \in M(G)$ and since $M(G)$ is a stable matching of G, we must have $x_{2^k+1} - x_{2^k} = c(e) \leq \min\{D_L, D_R\}$ as otherwise, at least one of the edges (x_1, x_{2^k}) or $(x_{2^k+1}, x_{2^{k+1}})$ is unstable. Figure 5 illustrates the various notions.

We say that a 2^k-vertex weighted line graph is *consistent* with H^k if it can be obtained from H^k by scaling the edge costs. Fixing the external vertices of L and R, we argue that the internal vertices of L and R can be repositioned so that L and R, resp., become consistent with H^k without violating the validity of ξ as a weighted line MC and without decreasing the ratio $\rho(\xi)$. We shall establish this fact for L; the proof for R is analogous. Note first that since $M(H^k)$ is stable in H^k and since $c(e) \leq D_L$, it follows that by repositioning the internal vertices of L so that L becomes consistent with H^k, we do not violate the stability of $M(G)$. Second, by the inductive hypothesis, repositioning the internal vertices of L so that L becomes consistent with H^k maximizes $c(M(L))/c(M^*(L))$, thus $\rho(\xi)$ cannot decrease after this repositioning step, which establishes the argument. So, assume hereafter that both L and R are consistent with H^k.

Assume without loss of generality that $D_L \geq D_R$, so $c(e) = x_{2^k+1} - x_{2^k}$ is at most D_R. In fact, since R is consistent with H^k, it follows that we can increase the difference $x_{2^k+1} - x_{2^k}$ until it is equal to D_R, keeping the difference $x_{i+1} - x_i$ unchanged for all other is, without violating the validity of ξ as a weighted line MC and without decreasing the ratio $\rho(\xi)$. So, assume hereafter that $D_L \geq c(e) = D_R$. Now, we argue that we can scale down the differences $x_{i+1} - x_i$ for every $1 \leq i < 2^k$, keeping $x_{i+1} - x_i$ unchanged for all other is, until we obtain $D_L = c(e) = D_R$, without decreasing the ratio $\rho(\xi)$. This completes the proof since $D_L = c(e) = D_R$ implies that $G = H^{k+1}$.

Let $\ell = c(M(L)) - D_L$, $\ell^* = c(M^*(L))$, $r = c(M(R)) - D_R$, and $r^* = c(M^*(R))$; notice that $\ell + \ell^* = D_L$ and $r + r^* = D_R$. Since $c(e) = D_R$, we can express $\rho(\xi)$ as

$$\rho(\xi) = \frac{c(M(G))}{c(M^*(G))} = \frac{2\ell + \ell^* + 2(r + r^*) + 2r + r^*}{\ell^* + r^*} = \frac{2\ell + \ell^* + 4r + 3r^*}{\ell^* + r^*} \ .$$

Recalling that $D_L \geq D_R$, we express D_L as $D_L = (1 + \lambda)D_R$ for some $\lambda \geq 0$, and so $\ell = (1 + \lambda)r$ and $\ell^* = (1 + \lambda)r^*$. Thus,

$$\rho(\xi) = \frac{2(1 + \lambda)r + (1 + \lambda)r^* + 4r + 3r^*}{(1 + \lambda)r^* + r^*} = \frac{(6 + 2\lambda)r + (4 + \lambda)r^*}{(2 + \lambda)r^*} \ .$$

Assuming that the edge costs in G (as a whole) are scaled so that $R = H^k$ (rather than merely being consistent with H^k), and recalling the properties of ξ^k_{RT}, we get

$$\rho(\xi) = \frac{(6 + 2\lambda)(3^{k-1} - 2^{k-1}) + (4 + \lambda)2^{k-1}}{(2 + \lambda)2^{k-1}} = \left(\frac{6 + 2\lambda}{2 + \lambda}\right) \cdot (3/2)^{k-1} - 1 \ .$$

The lemma follows since the function $f(\lambda) = \frac{6+2\lambda}{2+\lambda}$ is monotonically decreasing for $\lambda \geq 0$, meaning that it assumes its maximum for $\lambda = 0$ which implies that $D_L = D_R$ has to hold. □

C Lower Bound on PoS_α

Our goal in this section is to prove Theorem 19 and thereby establish a lower bound on PoS_α of minimum-cost perfect matching in metric graphs with $2n$ vertices.

Theorem 19. *PoS_α of minimum-cost perfect matching in metric graphs with $2n$ vertices is $\Omega(n^{\log(1+1/(2\alpha))})$.*

The graph construction that lies at the heart of this lower bound, denoted H^k_α, is a parametrized variant of the Reingold-Tarjan graph H^k presented in Appendix B and depicted in Fig. 4 for arbitrary values of α. Specifically, the 2-vertex graph H^1_α is identical to H^1; and the 2^{k+1}-vertex graph H^{k+1}_α is constructed recursively by placing two disjoint instances of H^k_α, each of diameter D^k_α, on the real line, only that this time, the spacing between them is set to $S^{k+1}_\alpha = (1/\alpha - \varepsilon)D^k_\alpha$, for some sufficiently small $\varepsilon > 0$ that will be determined later on. This implies that $D_\alpha^k = (2 + 1/\alpha - \varepsilon)^{k-1}$ and $S_\alpha^{k+1} = (1/\alpha - \varepsilon)(2 + 1/\alpha - \varepsilon)^{k-1}$.

Now let M be an α-stable matching in H_α^k. We argue that M has to contain each edge $e = (x, y)$ with $c(e) = 1/\alpha - \varepsilon$. Indeed, if $e \notin M$, then e is α-unstable in M since $c(e) < \alpha \cdot \min\{c(x, x'), c(y, y')\}$ for all other vertices x', y'. Given that all vertices with distance $1/\alpha - \varepsilon$ are therefore already matched, we can apply the same argument for each edge connecting two adjacent vertices with edge cost $(1/\alpha - \varepsilon)(2 + 1/\alpha - \varepsilon)$ and thereby conclude that these edges have to be in M as well. By repeating this argument, we end up with the unique α-stable matching M that has to contain the edge $(x_1^k, x_{2^k}^k)$ whose cost is D_α^k and and all other edges whose cost differs from 1. Thus, $c(M) \geq D_\alpha^k = (2 + 1/\alpha - \varepsilon)^{k-1}$.

On the other hand, the cost of the minimum-cost matching M^* is not larger than that of the matching using all cost 1 edges, thus we can bound the cost of M^* as $c(M^*) \leq 2^{k-1}$. Together, we conclude that

$$\begin{aligned} \mathrm{PoS}_\alpha(H_\alpha^k) \; &\geq \; \frac{c(M)}{c(M^*)} \; \geq \; \frac{(2 + 1/\alpha - \varepsilon)^{k-1}}{2^{k-1}} \\ &= \; \Omega\left(1 + \frac{1}{2\alpha}\right)^{k-1} \; = \; \Omega\left(n^{\log\left(1 + \frac{1}{2\alpha}\right)}\right) \; , \end{aligned} \tag{2}$$

where the last two equalities hold by taking a sufficiently small ε and by recalling that H_α^k has $2n = 2^k$ vertices, resp.

D Additional Proofs from Section 4

Proof (Lemma 3). We consider a flip of the edge $e = (u, v)$ and denote by $e' = (M(u), M(v))$ the second new edge joining M as a result of the flip. The two edges that are removed by the flip are denoted by f and g. See Fig. 6 for an illustration of the situation.

When an edge e is flipped, there are essentially two different cases for an unstable edge to be created. The unstable edge contains either one vertex of e or one vertex of e'. No other vertices are involved in the flip and thus every *new* unstable edge has to contain at least one of the four vertices. We assume without loss of generality that a vertex of the edge g is incident to the unstable edge created by the flip.

Let us first consider the case where a vertex of e is incident to the new unstable edge. This case is denoted as the edge b_1 in Fig. 6. We assume that b_1 is stable before the flip and unstable thereafter. For b_1 to be unstable after the flip, we must have $\alpha \cdot c(b_1) < c(e)$ and $\alpha \cdot c(b_1) < c(c)$. But as e is unstable before the flip, we have $\alpha \cdot c(e) < c(g)$ and thus we get $\alpha \cdot c(b_1) < c(e) < c(g)/\alpha \leq c(g)$. This means that b_1 was already unstable before the flip, which is a contradiction to the assumption. Hence, no vertex of e can be part of the new unstable edge.

Let us now consider the case, where a vertex from e' is part of the new unstable edge (b_2 in Fig. 6). Since b_2 is stable before the flip and unstable after it, we must have $c(g) \leq \alpha \cdot c(b_2) < c(e')$. But as e is unstable before the flip, we have $\alpha \cdot c(e) < c(g)$, and thus we get $c(e) < c(g)/\alpha \le c(b_2)$ which completes the proof. □

Proof (Lemma 5). The running time of the algorithm is dominated by the step of sorting the edges in G according to their cost. This takes $\mathcal{O}(n^2 \log n)$ steps. The second phase — the actual algorithm — runs in $\mathcal{O}(n^2)$ steps since it iterates once over all edges in $V \times V$ and each iteration takes $\mathcal{O}(1)$ time.

The correctness of the algorithm is established by Corollary 4 since it states that in the last iteration, all unstable edges have strictly larger cost than the edge currently considered or will be considered later. Since this edge is already the one with the largest cost and all edges have been considered, there cannot be any unstable edges in the final matching M_S. □

Proof (Lemma 6). Let us assume for the sake of contradiction that $e = (u, v)$ was flipped in iteration i of the algorithm and further that $(u, v) \notin M_j$ for some $j > i$. According to the algorithm, we have $(u, v) \in M_i$. Since $(u, v) \notin M_j$, there has to exist an iteration k with $i < k \le j$ where (u, v) is removed from M_{k-1} such that $(u, v) \notin M_k$. For this to happen, either edge (u, u') or (v, v') for some vertex u' or v' must be flipped in iteration k because it was unstable in M_{k-1}. Without loss of generality, we assume that (u, u') is unstable in M_{k-1} and flipped in iteration $k > i$. Thus, we have $c(u, u') \le \alpha \cdot c(u, u') < c(u, v)$. But this means that STAB would have considered the edge (u, u') before considering the edge (u, v), a contradiction to the assumption. □

Proof (Lemma 7). We prove the statement by induction over the height of x in its flip tree. The assertion holds for every leaf $x \sim e$ in the flip forest as $\psi(x) = c(e)$ by definition. Assume that the statement holds for the two children x_L and x_H of a node x that represents a flip of the edge (u, v). Then $x \sim (M(u), M(v)) = e$ and we assume without loss of generality that $x_H \sim (u, M(u)) = e_u$ and $x_L \sim (v, M(v)) = e_v$. Thus, by the inductive hypothesis, $c(e_u) \le \psi(x_H)$ and $c(e_v) \le \psi(x_L)$. This flip tree segment represents the replacement of the edges e_u and e_v by e and (u, v), which happened because the edge (u, v) was unstable with respect to M, that is, $\alpha \cdot c(u, v) < \min\{c(e_v), c(e_u)\}$. Since G is metric, we can bound $c(e)$ as follows.

$$\begin{aligned} c(e) &\le c(e_u) + c(e_v) + c(u, v) \\ &< c(e_u) + (1 + 1/\alpha) \cdot c(e_v) \\ &\le \psi(x_H) + (1 + 1/\alpha) \cdot \psi(x_L) \qquad \text{(inductive hypothesis)} \\ &= \psi(x) \end{aligned}$$
□

Proof (Lemma 10). Aiming for a contradiction, suppose that T is not complete. Let z be an internal node at depth d with leaf children x, y (whose depth is $d+1$) and let z' be a leaf at depth $d' < d$. Let T' be the full binary tree obtained from T by deleting x and y and inserting two new leaves x', y' as children of z'. Let w and w' be the leaf weigh functions of T and T', respectively, defined by requiring that T and T' are ψ-balanced and scaled so that $\Psi(T) = \Psi(T') = 1$;

this is well defined since by Observation 9, the ψ-values of all nodes in T and T' (and in particular, the leaf weight functions w and w') are fully determined by their topology and the values of $\Psi(T)$ and $\Psi(T')$ (in a top-down fashion).

We establish the proof by arguing that $\Lambda(T') < \Lambda(T)$ which implies $\eta(T') > \eta(T)$, in contradiction to T being effective. To that end, notice that the construction of T' implies that

$$\Lambda(T') = \Lambda(T) + w'(x') + w'(y') + w'(z) - (w(x) + w(y) + w(z')) ,$$

so it suffices to prove that $w'(z) - w(x) - w(y) < w(z') - w'(x') - w'(y')$. Employing Observation 9, we calculate

$$w'(z) = (2 + 1/\alpha)^{-d} \qquad w(x) = w(y) = (2 + 1/\alpha)^{-(d+1)}$$
$$w(z') = (2 + 1/\alpha)^{-d'} \qquad w'(x') = w'(y') = (2 + 1/\alpha)^{-(d'+1)} ,$$

so the proof reduces to showing that

$$(2 + 1/\alpha)^{-d} (1 - 2/(2 + 1/\alpha)) < (2 + 1/\alpha)^{-d'} (1 - 2/(2 + 1/\alpha))$$

which holds since $d > d'$. □

Proof (Lemma 11). Again we assume without loss of generality that the weights of the leaves are scaled so that $\Psi(T) = 1$. By definition, T has $2^h - 2k$ leaves at depth h and k leaves at depth $h - 1$. Employing Observation 9, we conclude that

$$\begin{aligned} \Lambda(T) &= (2^h - 2k) \cdot (2 + 1/\alpha)^{-h} + k \cdot (2 + 1/\alpha)^{-(h-1)} \\ &= (2 + 1/\alpha)^{-h} \cdot (2^h - 2k + k \cdot (2 + 1/\alpha)) \\ &= (2 + 1/\alpha)^{-h} \cdot (2^h + k/\alpha) \ . \end{aligned}$$

Since $\Psi(T) = 1$, we have $\eta(T) = 1/\Lambda(T)$ which completes the proof. □

Proof (Lemma 12). We show that for all $n \in \mathbb{Z}_{>0}$, it holds that $\varphi(n+1) > \varphi(n)$ and distinguish two cases. First, we consider the case that $n \neq 2^i$ for all $i \in \mathbb{Z}_{>0}$. Observe that $h(n+1) = h(n)$ and $k(n+1) < k(n)$ and therefore $\varphi(n+1) > \varphi(n)$. Now, we examine the case that $n = 2^i$ for some $i \in \mathbb{Z}_{>0}$. We have $h(n+1) = i+1$ and $h(n) = i$ as well as $k(n + 1) = 2^i - 1$ and $k(n) = 0$. Plugging these values into φ, we obtain $\varphi(n + 1) > \varphi(n)$ and the proof is complete. □

D.1 The Effect of a Flip Tree is Scaling-Invariant

Our goal in this section is to show that the effect of a full binary tree is invariant under scaling its leaf weights.

We define the *light depth* $\lambda(x)$ of a node x in a flip tree T as the number of light links on the direct path from x to the root r_T of T. Lemma 20 relates the weight of an inner node of a flip tree to the weights of the leaves of its subtree.

Lemma 20. *Every node x in a flip tree satisfies*

$$\psi(x) \;=\; \sum_{\ell \in \mathcal{L}(x)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x)} \cdot \psi(\ell) \;\; .$$

Proof. We prove the statement by induction over the height of x in its flip tree. The statement holds for a leaf node x since then we have $\mathcal{L}(x) = \{x\}$ and $\lambda(x) - \lambda(x) = 0$. Assume that the statement holds for both children x_H and x_L of a node x. By definition, we have

$$\begin{aligned}
\psi(x) \;&=\; \psi(x_H) + (1+1/\alpha) \cdot \psi(x_L) \\
&=\; \sum_{\ell \in \mathcal{L}(x_H)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x_H)} \cdot \psi(\ell) \\
&\quad\; + (1+1/\alpha) \cdot \sum_{\ell \in \mathcal{L}(x_L)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x_L)} \cdot \psi(\ell) \\
&=\; \sum_{\ell \in \mathcal{L}(x_H)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x)} \cdot \psi(\ell) \\
&\quad\; + (1+1/\alpha) \cdot \sum_{\ell \in \mathcal{L}(x_L)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x)-1} \cdot \psi(\ell) \\
&=\; \sum_{\ell \in \mathcal{L}(x_H)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x)} \cdot \psi(\ell) + \sum_{\ell \in \mathcal{L}(x_L)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x)} \cdot \psi(\ell) \\
&=\; \sum_{\ell \in \mathcal{L}(x)} (1+1/\alpha)^{\lambda(\ell)-\lambda(x)} \cdot \psi(\ell) \;\; ,
\end{aligned}$$

where we used $\lambda(x_L) = \lambda(x) + 1$ and $\lambda(x_H) = \lambda(x)$. □

Corollary 21 is immediate, since $\lambda(r_T) = 0$ for the root r_T of a flip tree T.

Corollary 21. *The root r_T of a flip tree T satisfies*

$$\psi(r_T) \;=\; \sum_{\ell \in \mathcal{L}(T)} (1+1/\alpha)^{\lambda(\ell)} \cdot \psi(\ell) \;\; .$$

As the effect of a flip tree T is defined to be $\Psi(T)/\Lambda(T)$ and by definition, $\Psi(T) = \psi(T)$ as well as $\Lambda(T) = \sum_{\ell \in \mathcal{L}(r_T)} \psi(\ell)$, we observe that scaling the leaf weights of a flip tree does not change its effect and the statement follows.

Illustrative Figures

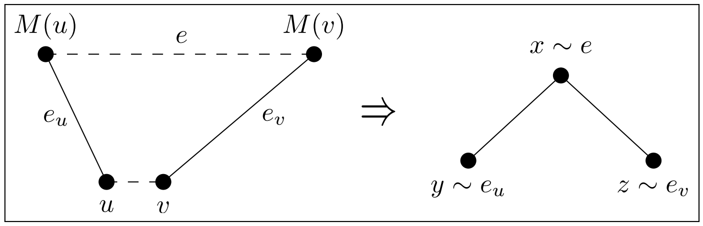
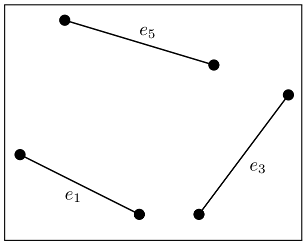
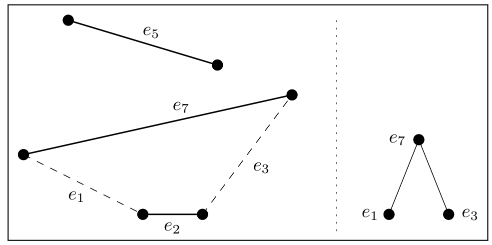


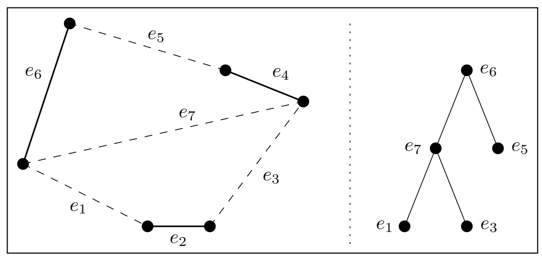
Fig. 1. The left side shows a matching configuration with an unstable edge (u, v), which will be flipped by STAB. This flip is then represented by the flip tree segment on the right, which depicts the replacement of the two active edges $(u, M(u)) \sim y$ and $(v, M(v)) \sim z$ by the active edge $(M(u), M(v)) \sim x$.



(a) The initial matching M_0 is a minimum-cost matching.



(b) Matching M_1 is obtained from M_0 by flipping e_2. The corresponding flip tree is shown on the right.



(c) Matching M_2 is obtained from M_1 by flipping e_4. The corresponding flip tree is shown on the right.

Fig. 2. This figure shows how the initial minimum-cost matching M_0 is transformed by an execution of STAB through the flips of the edges e_2 and e_4 along with the flip forest (here only a single flip tree) corresponding to the execution. Edges in the current matching are drawn with solid lines while edges in matchings of previous iterations are drawn with dashed lines.

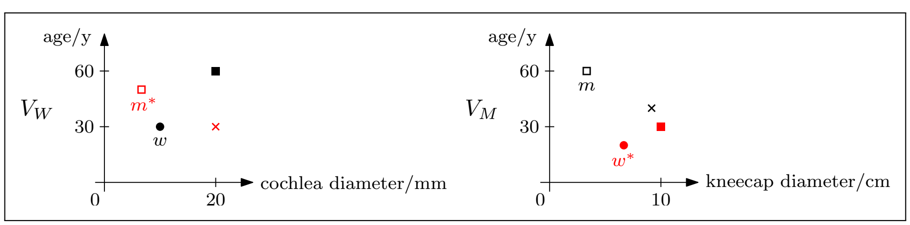


Fig. 3. In a fictitious example where age is relevant for both sexes while men are particularly interested in women's cochleas and women care about men's kneecaps, we depict the points corresponding to the characteristics of four individuals, two women and two men. A black mark indicates the characteristic of a specific user while the corresponding shape in red marks the characteristic of that person's ideal partner.

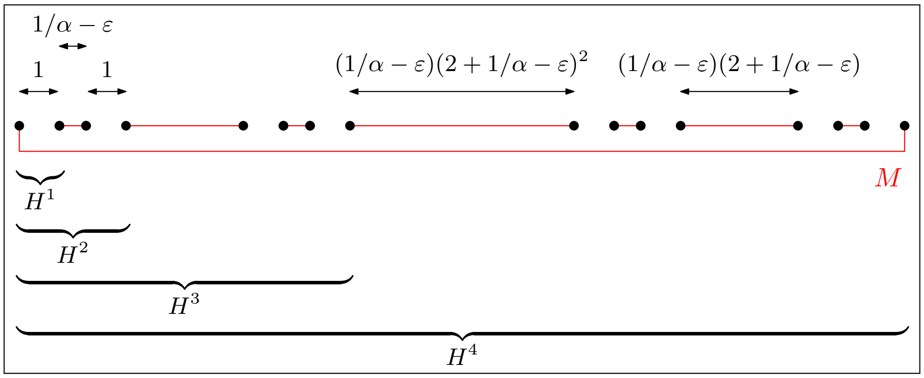


Fig. 4. This parametrized Reingold-Tarjan graph H^4_α with 2^4 vertices has a unique "expensive" α-stable matching M (red edges). Setting the optional parameters α and ε (that are used in the proof of the PoS lower bound) to 1 and 0, resp., yields the original Reingold-Tarjan graph H^4.

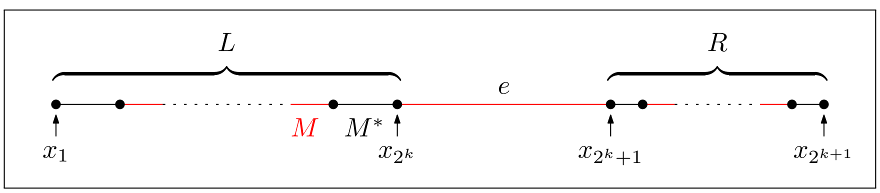


Fig. 5. Any MC ξ on 2^k vertices can be transformed into a Reingold-Tarjan MC without decreasing the ratio $\rho(\xi)$. The black edges are part of the minimum-cost matching M^* while the red edges belong to the stable matching M.

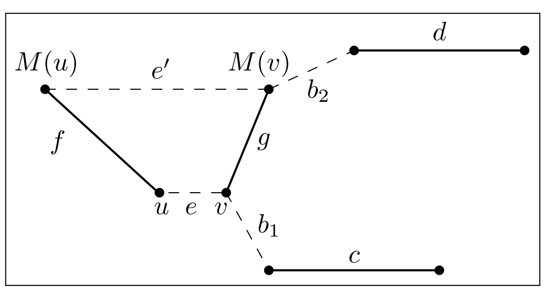


Fig. 6. This figure illustrates the two different cases of Lemma 3.