

The Computational Complexity of Weighted Greedy Matching

Argyrios Deligkas

Department of Computer Science
University of Liverpool, UK.
a.deligkas@liverpool.ac.uk

George B. Mertzios

School of Engineering and
Computing Sciences
Durham University, UK.
george.mertzios@durham.ac.uk

Paul G. Spirakis*

Department of Computer Science
University of Liverpool, UK
and
Computer Technology Institute, Greece.
p.spirakis@liverpool.ac.uk

Abstract

Motivated by the fact that in several cases a matching in a graph is stable if and only if it is produced by a greedy algorithm, we study the problem of computing a *maximum weight greedy matching* on weighted graphs, termed GREEDY-MATCHING. In wide contrast to the maximum weight matching problem, for which many efficient algorithms are known, we prove that GREEDYMATCHING is *strongly NP-hard* and *APX-complete*, and thus it does not admit a PTAS unless $P=NP$, even on graphs with maximum degree at most 3 and with at most three different integer edge weights. Furthermore we prove that GREEDYMATCHING is *strongly NP-hard* if the input graph is in addition *bipartite*. Moreover we consider three natural parameters of the problem, for which we establish a *sharp threshold* behavior between NP-hardness and computational tractability. On the positive side, we present a randomized approximation algorithm (RGMA) for GREEDYMATCHING on a special class of weighted graphs, called *bush graphs*. We highlight an unexpected connection between RGMA and the approximation of maximum cardinality matching in unweighted graphs via randomized greedy algorithms. We show that, if the approximation ratio of RGMA is ρ , then for every $\epsilon > 0$ the randomized MRG algorithm of [8] gives a $(\rho - \epsilon)$ -approximation for the maximum cardinality matching. We conjecture that a tight bound for ρ is $\frac{2}{3}$; we prove our conjecture true for four subclasses of bush graphs. Proving a tight bound for the approximation ratio of MRG on unweighted graphs (and thus also proving a tight value for ρ) is a long-standing open problem [26]. This unexpected relation of our RGMA algorithm with the MRG algorithm may provide new insights for solving this problem.

Introduction

Stable matching problems lie in the intersection of AI, economics and social choice theory and have been studied extensively over the years. Originally, *two-sided matchings* were studied [18, 29]. In a two-sided matching problem there is an underlying graph where the vertices correspond to players that want to be matched with a vertex from their neighborhood. Traditionally in two-sided matchings, every player is associated with an *ordering* over its neighbors that

correspond to its preferences. The goal is to decide whether a stable matching exists and, if it does, to find one.

Recently, a line of research has emerged that studies the *quality* of stable matchings [5]. In this setting, each match is associated with a utility and the goal is to find a stable matching where the social welfare in this matching, i.e. the sum of the utilities, is as large as possible. Extreme Programming [12] is a prominent example. In this case, the players are programmers and the utility of a match corresponds to the productivity of the pair of programmers. The goal is to maximize the productivity while keeping the individual programmers happy. Another application where a stable matching with high quality is desired, is kidney exchange. Traditionally only compatibility constraints were imposed for kidney exchange problems [28, 30], but recently other factors which give a score to each possible match were considered too [13, 20]. For this reason, recent AI literature on kidney exchange tried to optimize the overall social welfare of the matching instead of the number of compatible matches [2, 11].

The simplest and probably the most well-studied problem regarding the quality of the stable matching is known as *correlated stable matching* [21]. This model can be represented by an edge-weighted graph, where the players who are matched together obtain the same utility specified by the weight of the edge between them. Apart from the extreme programming and the kidney exchange problems, the correlated stable matching problem finds applications in social networks [21], job markets [7], distributed networks [17, 23], and market sharing [17]. Furthermore, a potential function exists for this model which in turn guarantees the existence of at least one stable matching even in non-bipartite graphs with ties [1, 23]. Thus, an immediate question is to understand the quality of a stable matching compared to the “best” one and to compute a stable matching of high quality. Anshelevich, Das, and Naamad [3] and Anshelevich, Bhardwaj, and Hoefer [4] studied the price of anarchy and stability of this model. Furthermore, in [3] it was observed that a matching is stable in the Nash sense if it is produced by a *greedy* matching algorithm. As it turns out, a correlated matching is stable *if and only if* it can be produced by a greedy algorithm. A greedy algorithm creates a matching by iteratively adding edges to matching that have the maximum available weight. Hence, a natural algorithmic

*The work was supported partially by the EU ERC Project AL-GAME.

question is whether a maximum weight greedy matching can be efficiently computed or approximated. Although greedy algorithms for matching problems have been studied extensively in the past [8, 15, 16, 19, 24, 26], to the best of our knowledge not much is known about the problem of computing a maximum weight greedy matching.

Related work

The scenarios of matching problems where the vertices of the graph correspond to players can vary from matching employees and employers [22], to matching kidney donors and recipients [2, 28]. The authors in [3] provided algorithms that compute *almost* stable matchings. Our work is closely related to [3], although their techniques cannot be applied to our problem since we focus *only* on matchings that are greedy (i.e. stable). Recently, greedy matching algorithms were used in ordinal settings to approximate the cardinal utility of maximum weight matching [6].

Greedy matchings have been studied extensively over the years. The classical result by Korte and Hausmann [19] states that an arbitrary greedy matching is a $\frac{1}{2}$ -approximation of the maximum cardinality matching, i.e. every greedy matching on unweighted graphs picks at least half of maximum number of edges that any matching can pick. For edge-weighted graphs, Avis [10] showed that every algorithm that greedily picks edges with the maximum currently available weight is a $\frac{1}{2}$ -approximation of the maximum weight matching. Hence, every weighted greedy matching is also a $\frac{1}{2}$ -approximation for the maximum weight greedy matching problem. Several authors studied randomized greedy algorithms for the maximum cardinality matching problem. The currently best randomized algorithm, known as MRG [8], picks the next edge to add to the matching by first selecting a random unmatched vertex V of the graph and then a random unmatched neighbor of v . Aronson, Dyer, Frieze and Suen [8] showed that MRG breaks the $\frac{1}{2}$ -barrier and that it achieves a $\frac{1}{2} + 1/400,000$ -approximation guarantee on every graph. Recently, Poloczek and Szegedy [26] provided a different analysis for MRG and showed that it achieves an approximation guarantee of at least $\frac{1}{2} + \frac{1}{256}$. However, as experiments suggest, the approximation guarantee of MRG can be as large as $\frac{2}{3}$ [26].

Our contribution

In this paper we study the computational complexity of computing and approximating a *maximum weight greedy matching* in a given edge-weighted graph, i.e. a greedy matching with the greatest weight among all greedy matchings. This problem is termed GREEDYMATCHING. In wide contrast to the maximum weight matching, for which many efficient algorithms are known (see [14] and the references therein), we prove that GREEDYMATCHING is *strongly NP-hard* by a reduction from a special case of MAX2SAT. Our reduction also implies hardness of approximation; we prove that GREEDYMATCHING is *APX-complete*, and thus it does not admit a PTAS unless $P=NP$. These hardness results hold even for input graphs with maximum degree at most 3 and

with at most three different integer edge weights, namely with weights in the set $\{1, 3, 4\}$. Furthermore, by using a technique of Papadimitriou and Yannakakis [25], we extend the NP-hardness proof to the interesting case where the input graph is in addition *bipartite*. Next, we study the decision variations GREEDYVERTEX and GREEDYEDGE of the problem, where we now ask whether there exists a greedy matching in which a specific vertex u or a specific edge (u, v) is matched. These are both natural questions, as the designer of the stable matching might want to ensure that a specific player or a specific pair of players is matched in the solution. We prove that both GREEDYVERTEX and GREEDYEDGE are also strongly NP-hard.

As GREEDYMATCHING turns out to be computationally hard, it makes sense to investigate how the complexity is affected by appropriately restricting the input. In this line of research we consider three natural parameters of the problem, for which we establish a *sharp threshold* behavior. As the first parameter we consider the number of the weight values of the graph. Note that when there is only one weight value on the edges, GREEDYMATCHING can be reduced to the maximum cardinality problem and thus it can be solved efficiently. We prove that GREEDYMATCHING is NP-complete even on graphs that are bipartite or planar, have maximum vertex degree 4, and there are only two weight values on their edges.

As the second parameter we consider the *minimum ratio* λ_0 of any two *consecutive weights*. Assume that the graph has ℓ different edge weights $w_1 > w_2 > \dots > w_\ell$; we define for every $i \in [\ell - 1]$ the ratio $\lambda_i = \frac{w_i}{w_{i+1}}$ and the minimum ratio $\lambda_0 = \min_{i \in [\ell-1]} \lambda_i$. We prove that, if $\lambda_0 \geq 2$ then GREEDYMATCHING can be solved in polynomial time, while for any constant $\lambda_0 < 2$ GREEDYMATCHING is strongly NP-hard and APX-complete, even on graphs with maximum degree at most 3 and with at most three different edge weights.

The last parameter we consider is the *maximum edge cardinality* μ of the *connected components* of $G(w_i)$, among all different weights w_i , where $G(w_i)$ is the subgraph of G spanned by the edges of weight w_i . Although at first sight this parameter may seem unnatural, it resembles the number of times that the greedy algorithm has to break ties. At the stage where we have to choose among all available edges of weight w_i , it suffices to consider each connected component of the available edges of $G(w_i)$ separately from the other components. In particular, although the weight of the final greedy matching may highly depend on the order of the chosen edges within a connected component, it is independent of the ordering that the various different connected components are processed. Thus μ is a reasonable parameter for GREEDYMATCHING. In the case $\mu = 1$ there exists a unique greedy matching for G which can be clearly computed in polynomial time. We prove that GREEDYMATCHING is strongly NP-hard and APX-complete for $\mu \geq 2$, even on graphs with maximum degree at most 3 and with at most five different edge weights.

On the positive side, we consider a special class of weighted graphs, called *bush graphs*, where all edges of the same weight in G form a star (bush). We present a ran-

domized approximation algorithm (RGMA) for GREEDYMATCHING on bush graphs and we highlight an unexpected connection between RGMA and the randomized MRG algorithm for greedily approximating the maximum cardinality matching on unweighted graphs. In particular we show that, if the approximation ratio of RGMA for GREEDYMATCHING on bush graphs is ρ , then for every $\epsilon > 0$ MRG [8] is a $(\rho - \epsilon)$ -approximation algorithm for the maximum cardinality matching. We conjecture that a tight bound for ρ is $\frac{2}{3}$; among our results we prove our conjecture true for four subclasses of bush graphs. Proving a tight bound for the approximation ratio of MRG on unweighted graphs (and thus also proving a tight value for ρ) is a long-standing open problem [8, 15, 26]. This unexpected relation of our RGMA algorithm with the MRG algorithm may provide new insights for solving this problem.

Preliminaries

Every graph considered in this paper is undirected. For any graph $G = (V, E)$ we use $G + u$ to denote the graph $G' = (V', E')$ where $V' = V \cup \{u\}$ and E' consists of the set E and all the edges the vertex u belongs to. Similarly $G - V'$ denotes the induced graph of G defined by $V \setminus V'$, where $V' \subseteq V$. We study graphs $G = (V, E)$ with positive edge weights, i.e. each edge $e = (u, v) \in E$ has a weight $w(e) = w(u, v) > 0$. The *degree* of a vertex u is the number of its adjacent vertices in G . We use $G(w_i)$ to denote the subgraph of G spanned by the edges of weight w_i . A *matching* $\mathcal{M} \subseteq E$ is a set of edges such that no pair of them are adjacent. The weight of a matching \mathcal{M} is the sum of the weights of the edges in \mathcal{M} , formally $w(\mathcal{M}) = \sum_{e \in \mathcal{M}} w(e)$. A *greedy matching* is a maximal matching constructed by the Greedy Matching Procedure.

Input: Graph $G = (V, E)$, with $w_1 > w_2 > \dots > w_\ell$ edge weight values

Output: Greedy matching \mathcal{M}

1. $\mathcal{M} \leftarrow \emptyset$
2. **for** $i = 1 \dots \ell$ **do**
3. **while** there is an $e \in E$ such that $w(e) = w_i$ **do**
4. Pick an edge $e^* \in E$ with $w(e^*) = w_i$ and add it to \mathcal{M} ;
5. Remove all edges adjacent to e^* from E ;

Algorithm 1: Greedy Matching Procedure

Notice that in Step 4 the edge that is added to the matching \mathcal{M} is not specified explicitly. The rule that specifies which edge is chosen in Step 4 can be deterministic or randomized, resulting in a specific *greedy matching algorithm*. We denote by $\text{OPT}(G)$ the optimum of GREEDYMATCHING with input G , i.e. a *maximum weight greedy matching* of G .

GREEDYMATCHING

INSTANCE: Graph $G = (V, E)$ with positive edge weights.

TASK: Compute a maximum weight greedy matching \mathcal{M} for G .

Furthermore, we study another two related problems, where we ask whether there is a greedy matching that matches a specific vertex or a specific edge.

GREEDYVERTEX

INSTANCE: Graph $G = (V, E)$ with positive edge weights and a vertex $v \in V$.

QUESTION: Is there a greedy matching \mathcal{M} such that $(v, u) \in \mathcal{M}$, for some $u \in V$?

GREEDYEDGE

INSTANCE: Graph $G = (V, E)$ with positive edge weights and an edge $(u, v) \in E$.

QUESTION: Is there a greedy matching \mathcal{M} such that $(v, u) \in \mathcal{M}$?

Hardness of GREEDYMATCHING

In this section we study the complexity of computing a maximum weight greedy matching. We prove that GREEDYMATCHING is strongly NP-hard and APX-complete, even on graphs with maximum degree at most 3 and with at most three different integer weight values. By slightly modifying our reduction, we first prove that GREEDYMATCHING remains strongly NP-hard also when the graph is in addition bipartite, and we then prove that also the two decision problem variations GREEDYVERTEX and GREEDYEDGE are also strongly NP-hard. Our hardness reductions are from the MAX2SAT(3) problem [9, 27], which is the special case of MAX-SAT where in the input CNF formula ϕ every clause has at most 2 literals and every variable appears in at most 3 clauses; we call such a formula ϕ a 2SAT(3) formula.

Note that the decision version of GREEDYMATCHING, where we ask whether there exists a greedy matching with weight at least B , belongs to the class NP. Indeed we are able to verify in polynomial time whether a given matching \mathcal{M} is maximal, greedy and has weight at least B . The maximality and the weight of the matching \mathcal{M} can be computed and checked in linear time. To check whether \mathcal{M} is greedy, we first check whether the largest edge weight in \mathcal{M} equals the largest edge weight in G . In this case we remove from G all vertices incident to the highest weight edges of \mathcal{M} and we apply recursively the same process in the resulting induced subgraph. Then \mathcal{M} is greedy if and only if we end up with a graph with no edges.

Overview of the reduction.

Given a 2SAT(3) formula ϕ with m clauses and n variables x_1, \dots, x_n we construct an undirected graph G with $10n + m$ vertices and $9n + 2m$ edges. Then we prove that there exists a truth assignment that satisfies at least k clauses of ϕ if and only if there exists a greedy matching \mathcal{M} in G with weight at least $14n + k$. Without loss of generality we make the following assumptions on ϕ . Firstly, if a variable occurs only with positive (resp. only with negative) literals, then we trivially set it true (resp. false) and remove the associated clauses. Furthermore, without loss of generality, if a variable x_i appears three times in ϕ , we assume that it appears once as a positive literal x_i and two times as a negative literal \bar{x}_i ; otherwise we rename the negation with a new variable. Similarly, if x_i appears two times in ϕ , then it appears once as a positive literal x_i and once as a negative literal \bar{x}_i .

For each variable x_i we create a subgraph \mathcal{G}_{x_i} and for each clause C_j we create one vertex v_j . The vertices created from the clauses will be called v -vertices. Each subgraph \mathcal{G}_{x_i} is a path with 10 vertices, where three of them are distinguished; the vertices $\alpha_{x_i}, \beta_{x_i}$ and γ_{x_i} . Each distinguished vertex can be connected with at most one v -vertex that represents a clause. Furthermore, every v -vertex is connected with at most two vertices from the subgraphs \mathcal{G}_{x_i} ; one distinguished vertex from each of the subgraphs \mathcal{G}_{x_i} that correspond to the variables of the clause. The edge weights in the subgraphs \mathcal{G}_{x_i} are not smaller than the weights of the edges connecting the v -vertices with the distinguished vertices of the subgraphs \mathcal{G}_{x_i} .

The construction

The gadget \mathcal{G}_{x_i} that we create for variable x_i is illustrated in Figure 1; the distinguished vertices of \mathcal{G}_{x_i} are $\alpha_{x_i}, \beta_{x_i}$ and γ_{x_i} . The vertex α_{x_i} corresponds to the *positive* literal of the variable and vertices β_{x_i} and γ_{x_i} correspond to the *negative* literal \bar{x}_i .

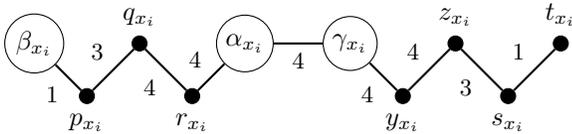


Figure 1: The gadget \mathcal{G}_{x_i} .

The vertex v_j associated to clause C_j , where $j \in [m]$, is made adjacent to the vertices that correspond to the literals associated with that clause. For example, if $C_j = (x_1 \vee \bar{x}_2)$ we will connect the vertex v_j with one of the vertices $\alpha_{x_1}, \beta_{x_1}, \gamma_{x_1}$ and with one of the vertices $\alpha_{x_2}, \beta_{x_2}, \gamma_{x_2}$. In order to make these connections in a consistent way, we first fix an arbitrary ordering over the clauses. If the variable x_i occurs as a positive literal in the clause C_j , then we add the edge (v_j, α_{x_i}) of weight 3. Next, if C_j is the first clause that the variable x_i occurs with a negative literal (in the fixed ordering of the clauses), then we add the edge (v_j, β_{x_i}) of weight 1. Finally, if the clause C_j is the second clause that the variable x_i occurs as a negative literal, then we add the edge (v_j, γ_{x_i}) of weight 3. That is, if a variable x_i appears

only two times in ϕ , then only the two distinguished vertices α_{x_i} and β_{x_i} of \mathcal{G}_{x_i} are adjacent to a v -vertex. This completes the construction of the graph G . Note that, by the construction of G , in any maximum greedy matching of G , there are exactly four alternative ways to match the edges of each of the subgraphs \mathcal{G}_{x_i} , as illustrated in Fig. 2-5.

APX-completeness

In order to prove that GREEDYMATCHING is APX-complete, first we prove in the next lemma that given an assignment that satisfies k clauses we can construct a greedy matching with weight $14n + k$. The intuition for this lemma is as follows. Starting with a given satisfying truth assignment τ for the input formula ϕ , we first construct the matching \mathcal{M}^- in every \mathcal{G}_{x_i} (cf. Figure 2), and thus the β -vertices are initially free to be matched to v -vertices. Then, if a variable x_i is true in τ , we change the matching of \mathcal{G}_{x_i} from \mathcal{M}^- to \mathcal{M}^+ (cf. Figure 4), such that only the α -vertex (and not the β and γ -vertices) of \mathcal{G}_{x_i} is free to be matched to a v -vertex. On the other hand, if the variable x_i is false in τ , then we either keep the matching \mathcal{M}^- in \mathcal{G}_{x_i} , or we replace \mathcal{M}^- with \mathcal{M}^{--} in \mathcal{G}_{x_i} (cf. Figure 3). Note that in \mathcal{M}^- only β_{x_i} is free to be matched, while in \mathcal{M}^{--} both β_{x_i} and γ_{x_i} are free to be matched with a v -vertex; in both cases the α -vertex of \mathcal{G}_{x_i} is “blocked” from being matched to a v -vertex. Then, using the fact that τ satisfies k clauses of ϕ , we can construct a matching of G where k v -vertices are matched and the total weight of this matching is at least $14n + k$.

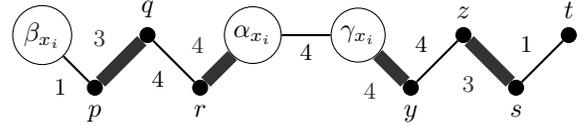


Figure 2: The matching \mathcal{M}^- with weight 14 for the subgraph \mathcal{G}_{x_i} . For simplicity of notation we do not include the subscript x_i in the non-distinguished vertices.

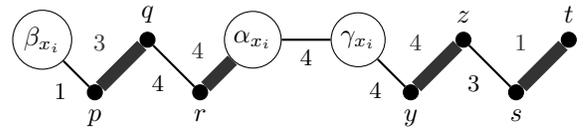


Figure 3: The matching \mathcal{M}^{--} with weight 12 for the subgraph \mathcal{G}_{x_i} .

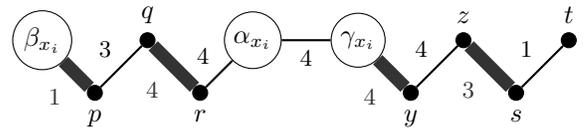


Figure 4: The matching \mathcal{M}^+ with weight 12 for the subgraph \mathcal{G}_{x_i} .

Lemma 1. *If there is an assignment that satisfies at least k clauses then, there is a greedy matching with weight at least $14n + k$.*

Next we prove in Lemma 3 that, if there is a greedy matching with weight $14n + k$, then there is an assignment that satisfies at least k clauses. In order to prove Lemma 3, first we prove in Lemma 2 a crucial property of the constructed graph G , namely that in any greedy matching at most one of the vertices α_{x_i} and γ_{x_i} can be matched with a v -vertex.

Lemma 2. *Let \mathcal{M} be an arbitrary greedy matching of G and let $i \in \{1, 2, \dots, n\}$. Then, in the subgraph \mathcal{G}_{x_i} , at most one of the vertices α_{x_i} and γ_{x_i} can be matched with a v -vertex.*

Proof. The proof is done by contradiction. Assume otherwise that both α_{x_i} and γ_{x_i} are matched with some v -vertices in \mathcal{M} . Note that both these edges that connect the vertices α_{x_i} and γ_{x_i} with the corresponding v -vertices have weight 3. Furthermore, none of the edges (α_{x_i}, r) , (γ_{x_i}, y) , and $(\alpha_{x_i}, \gamma_{x_i})$ belong to \mathcal{M} . Thus, since the weight of the edge $(\alpha_{x_i}, \gamma_{x_i}) \notin \mathcal{M}$ is 4, it follows \mathcal{M} is not greedy, which is a contradiction. That is, if both edges (α_{x_i}, r) and (γ_{x_i}, y) of the subgraph \mathcal{G}_{x_i} are not matched within \mathcal{M} , then $(\alpha_{x_i}, \gamma_{x_i}) \in \mathcal{M}$, as it is illustrated in the “bad” matching \mathcal{M}^b of Fig. 5. \square

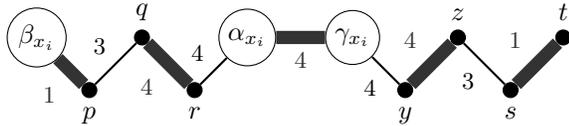


Figure 5: The “bad” matching \mathcal{M}^b for the subgraph \mathcal{G}_{x_i} with weight 14.

We are now ready to prove Lemma 3.

Lemma 3. *If there is a greedy matching with weight at least $14n + k$ in G , then there exists an assignment that satisfies at least k clauses of the formula ϕ .*

In the following theorem we conclude with the main result of this section.

Theorem 1. *GREEDYMATCHING is strongly NP-hard and APX-complete. In particular, unless $P=NP$, GREEDYMATCHING admits no PTAS, even on graphs with maximum degree at most 3 and with at most three different integer weight values.*

GREEDYMATCHING in Bipartite graphs

The graph G that we constructed from ϕ is not necessarily bipartite, as it may contain an odd-length cycle. More specifically, it is possible that the following cycle of length 9 exists:

$$v \rightarrow \beta_{x_i} \rightarrow p \rightarrow q \rightarrow r \rightarrow \alpha_{x_i} \rightarrow v' \rightarrow \gamma_{x_j} \rightarrow \alpha_{x_j} \rightarrow v.$$

However, as we prove in this section, GREEDYMATCHING remains strongly NP-hard also when the graph is in addition bipartite.

Theorem 2. *GREEDYMATCHING is strongly NP-hard, even on bipartite graphs with maximum degree at most 3 and with at most three different integer weight values.*

Hardness of GREEDYVERTEX and GREEDYEDGE

We now prove that the decision problems GREEDYVERTEX and GREEDYEDGE are also strongly NP-hard.

Theorem 3. *The decision problems GREEDYVERTEX and GREEDYEDGE are strongly NP-hard, even on graphs with at most five different edge weights.*

Further parameters of GREEDYMATCHING

In this section we investigate the influence of three natural parameters to the computational complexity of GREEDYMATCHING. As the first parameter we consider the number of different weight values on the edges. As the second parameter we study the minimum ratio λ_0 between two consecutive weight values, and as the last parameter we consider the maximum cardinality μ of the connected components of $G(w_i)$, over all possible weight values w_i . We prove that GREEDYMATCHING has a *sharp threshold* behavior with respect to each of these parameters.

Number of different weight values

Observe that if there is only one weight value on the edges of the graph, then GREEDYMATCHING can be reduced to the maximum cardinality matching problem. We prove that GREEDYMATCHING is NP-complete even when the underlying graph has only two weight values on its edges, is bipartite or planar and the maximum vertex degree is four. Thus, we completely characterize the complexity of GREEDYMATCHING with respect to the number of different weight values.

Theorem 4. *GREEDYMATCHING is NP-hard even in bipartite or planar graphs with at most two different weight values and maximum vertex degree four.*

Note that Theorem 4 does not supersede Theorems 1 and 2. Theorem 1 establishes APX-completeness for the problem. Furthermore, in Theorem 2 the graph has maximum vertex degree at most three, while in Theorem 4 the graph has maximum degree at most four.

Minimum ratio of consecutive weights

Here we consider the parameter $\lambda_0 = \min_i \lambda_i$, where $\lambda_i = \frac{w_i}{w_{i+1}} > 1$ is the ratio between the i th pair of consecutive edge weights. First we prove that, if $\lambda_0 \geq 2$, then there exists at least one maximum weight matching of G that is an optimum solution for GREEDYMATCHING on G , obtaining the next theorem.

Theorem 5. *GREEDYMATCHING can be computed in polynomial time if $\lambda_0 \geq 2$.*

Recall that in the proof of Theorem 1 the weight values 1, 3, and 4 were used, thus the GREEDYMATCHING is hard for $\lambda_0 \leq 4/3$. In the next theorem we amplify this result by showing that GREEDYMATCHING is NP-hard for any constant $\lambda_0 < 2$. That is, complexity of GREEDYMATCHING has a threshold behavior at the parameter value $\lambda_0 = 2$.

Theorem 6. *GREEDYMATCHING is strongly NP-hard and APX-complete for any constant $\lambda_0 < 2$, even on graphs with maximum degree at most 3 and with at most three different integer weight values.*

Maximum edge cardinality of a connected component in $G(w_i)$

Another parameter that we can consider is the maximum edge cardinality μ of the connected components of $G(w_i)$, among all different weights w_i . Since $\mu = 1$ implies that there is a unique greedy matching for G which can be clearly computed in polynomial time, we consider the case $\mu \geq 2$. In the original construction in every gadget \mathcal{G}_{x_i} there is a path with five edges where each edge has weight 4. Thus $\mu = 5$ in the graph G . However, by slightly modifying the previous construction we get the following theorem.

Theorem 7. GREEDYMATCHING is strongly NP-hard and APX-complete for $\mu \geq 2$, even on graphs with maximum degree at most 3 and with at most five different integer weight values.

A randomized approximation algorithm

In this section we provide a randomized approximation algorithm (RGMA) for GREEDYMATCHING with approximation ratio $\frac{2}{3}$ on four special classes of graphs. Furthermore we highlight an unexpected relation between RGMA and the randomized MRG algorithm for greedily approximating the maximum cardinality matching, the exact approximation ratio of which is a long-standing open problem [8, 15, 26]. Before we present our randomized algorithm RGMA, we first introduce the following class of weighted graphs, called *bush graphs*.

Definition 1 (Bush graph). An edge-weighted graph $G = (V, E)$ with ℓ edge weight values $w_1 > w_2 > \dots > w_\ell$ is a bush graph if, for every $i \in \{1, 2, \dots, \ell\}$, the edges of $G(w_i)$ form a star, which we call the i -th bush of G .

Input: Bush Graph G with edge weight values $w_1 > \dots > w_\ell$.

Output: A greedy matching \mathcal{M}_{RG} .

$\mathcal{M}_{RG} \leftarrow \emptyset$

for $i = 1 \dots \ell$ **do**

if $G_i \neq \emptyset$

 Select uniformly at random an edge $e_i \in G_i$ and add e_i to \mathcal{M}_{RG}

 Remove from G the endpoints of e_i and all edges of G_i

end

Algorithm 2: RGMA algorithm

Bush graphs and maximum cardinality matching

In this section we present the connection of the problem GREEDYMATCHING on (weighted) bush graphs to the problem of approximating the maximum cardinality matching in unweighted graphs via randomized greedy algorithms, cf. Theorem 8. Notice that we cannot directly apply the RGMA algorithm on unweighted graphs, since the algorithm has to consider the different bushes in a specific total order which is imposed by the order of the weights. Thus, in order to approximate a maximum cardinality matching in a given

unweighted graph G using the RGMA algorithm, we first appropriately convert G to a (weighted) bush graph G^* using the next Bush Decomposition algorithm, and then we apply RGMA on G^* .

Input: Unweighted graph $G = (V, E)$ and $\epsilon = \frac{1}{|V|^3}$.

Output: A (weighted) bush graph G^* .

Set $k \leftarrow 0$;

while $E \neq \emptyset$ **do**

 Choose a random vertex $u \in V$;

 For every $v' \in S := \{v' \in V : (u, v') \in E\}$ set $w(u, v') = 1 - k \cdot \epsilon$;

 Remove the edges of S from E ;

$k \leftarrow k + 1$;

end

Algorithm 3: Bush decomposition

Any unweighted graph $G = (V, E)$ can be considered as a weighted graph with edge weights $w(u, v) = 1$ for every edge $(u, v) \in E$, and thus in this case $\text{OPT}(G)$ coincides with the maximum cardinality matching in G . In the next lemma we relate $w(\text{OPT}(G^*))$ with $w(\text{OPT}(G))$.

Lemma 4. $w(\text{OPT}(G)) \geq w(\text{OPT}(G^*)) \geq w(\text{OPT}(G)) - \frac{1}{n}$.

With Lemma 4 in hand the next theorem follows:

Theorem 8. Let ρ be the approximation guarantee of RGMA algorithm on every bush graph. Then, for every $\epsilon < 1$, RGMA computes a $(\rho - \epsilon)$ -approximation of the maximum cardinality matching for unweighted graphs.

We conjecture that a tight bound for ρ is $\frac{2}{3}$ and we prove our conjecture in Theorem 9 for four subclasses of bush graphs (cf. the Definitions 2 and 3).

Definition 2. Let $G = (V, E)$ be an arbitrary edge-weighted tree which is a bush graph. Let x_1, \dots, x_k be the bush centers of G with decreasing weight and let x_i be a child of x_{i-1} , for every $2 \leq i \leq k$. If every bush of G has at least two leaves and at most one of these leaves is the center of another bush, then G is called a skewed bush tree.

Definition 3. Let $G = (V, E)$ be an unweighted tree and let r be a distinguished root of G . Let G^* be the (edge-weighted) bush graph that is produced by the Bush Decomposition, where we always choose the next bush center u in a breadth-first search fashion starting at the root r . If all leaves of G^* (and of G) have the same distance from r and if every bush has at least two edges, then G^* is called a balanced bush tree.

Theorem 9. RGMA achieves a $\frac{2}{3}$ approximation in the following four subclasses of bush graphs: bush graphs with two weight values, bush graphs with at most two edges per bush, skewed bush trees, and balanced bush trees.

Conclusions

Several interesting open questions stem from our paper. Probably the most important one is to derive tight approximation guarantees ρ for the maximum weight greedy matching problem, even for bush graphs. We conjecture that $\rho =$

$\frac{2}{3}$; an affirmative answer to our conjecture would imply that the algorithm MRG for maximum cardinality matching in unweighted graphs has an approximation ratio of almost $\frac{2}{3}$, thus solving a longstanding open problem [8, 15, 26].

References

- [1] David J. Abraham, Ariel Levavi, David Manlove, and Gregg O'Malley. The stable roommates problem with globally ranked pairs. *Internet Mathematics*, 5(4):493–515, 2008.
- [2] D.J. Abraham, A. Blum, and T. Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In *EC*, pages 295–304, 2007.
- [3] E. Anshelevich, S. Das, and Y. Naamad. Anarchy, stability, and utopia: creating better matchings. *Autonomous Agents and Multi-Agent Systems*, 26(1):120–140, 2013.
- [4] Elliot Anshelevich, Onkar Bhardwaj, and Martin Hoefer. Friendship and stable matching. In *ESA 2013*, pages 49–60, 2013.
- [5] Elliot Anshelevich and Sanmay Das. Matching, cardinal utility, and social welfare. *SIGecom Exchanges*, 9(1):4, 2010.
- [6] Elliot Anshelevich and Shreyas Sekar. Blind, greedy, and random: Algorithms for matching and clustering using only ordinal information. In *AAAI*, pages 390–396, 2016.
- [7] Esteban Arcaute and Sergei Vassilvitskii. Social networks and stable matchings in the job market. In *WINE*, pages 220–231, 2009.
- [8] J. Aronson, M. Dyer, A. Frieze, and S. Suen. Randomized greedy matching II. *Random Structures & Algorithms*, 6(1):55–73, 1995.
- [9] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag, 1999.
- [10] D. Avis. A survey of heuristics for the weighted matching problem. *Networks*, 13(4):475–493, 1983.
- [11] Pranjal Awasthi and Tuomas Sandholm. Online stochastic optimization in the large: Application to kidney exchange. In *IJCAI*, pages 405–411, 2009.
- [12] Milind Dawande, Subodha Kumar, Vijay Mookerjee, and Chelliah Sriskandarajah. Maximum commonality problems: Applications and analysis. *Management Science*, 54(1):194–207, 2008.
- [13] Segev DL, Gentry SE, Warren DS, Reeb B, and Montgomery RA. Kidney paired donation and optimizing the use of live donor organs. *JAMA*, 293(15):1883–1890, 2005.
- [14] R. Duan and S. Pettie. Linear-time approximation for maximum weight matching. *Journal of the ACM*, 61(1):1–23, 2014.
- [15] M. Dyer and A. Frieze. Randomized greedy matching. *Random Structures & Algorithms*, 2(1):29–45, 1991.
- [16] M. Dyer, A. Frieze, and B. Pittel. The average performance of the greedy matching algorithm. *Annals of Applied Probability*, 3(2):526–552, 05 1993.
- [17] Michel X. Goemans, Erran L. Li, Vahab S. Mirrokni, and Marina Thottan. Market sharing games applied to content distribution in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1020–1033, 2006.
- [18] Dan Gusfield and Robert W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA, USA, 1989.
- [19] D. Hausmann and B. Korte. K-greedy algorithms for independence systems. *Zeitschrift für Operations Research*, 22(1):219–228, 1978.
- [20] Philip J. Held, Barry D. Kahan, Lawrence G. Hunsicker, David Liska, Robert A. Wolfe, Friedrich K. Port, Daniel S. Gaylin, Jose R. Garcia, Lawrence Agodoa, and Henry Krakauer. The impact of hla mismatches on the survival of first cadaveric kidney transplants. *New England Journal of Medicine*, 331(12):765–770, 1994.
- [21] Martin Hoefer. Local matching dynamics in social networks. In *ICALP*, pages 113–124, 2011.
- [22] B. Jovanovic. Job matching and the theory of turnover. *Journal of Political Economy*, 87(5):972–990, 1979.
- [23] Fabien Mathieu. Self-stabilization in preference-based systems. *Peer-to-Peer Networking and Applications*, 1(2):104–121, 2008.
- [24] Z. Miller and D. Pritikin. On randomized greedy matchings. *Random Structures & Algorithms*, 10(3):353–383, 1997.
- [25] C.H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.
- [26] M. Poloczek and M. Szegedy. Randomized greedy algorithms for the maximum matching problem with new analysis. In *FOCS*, pages 708–717, 2012.
- [27] Venkatesh Raman, B. Ravikumar, and S. Srinivasa Rao. A simplified NP-complete MAXSAT problem. *Information Processing Letters*, 65(1):1–6, 1998.
- [28] A.E. Roth, T. Sönmez, and M.U. Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.
- [29] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-sided matching : a study in game-theoretic modeling and analysis*. Econometric society monographs. Cambridge university press, 1992, Cambridge, Mass., 1990.
- [30] Alvin E. Roth, Tayfun Snmez, and M. Utku nver. A kidney exchange clearinghouse in new england. *American Economic Review*, 95(2), 2005.