Algorithmica (2018) 80:410–413
https://doi.org/10.1007/s00453-017-0281-9

ERRATUM

Erratum to: Greedy Matching: Guarantees and Limitations

Bert Besser[1] · Matthias Poloczek[2]

Published online: 10 March 2017

Abstract In "Greedy Matching: Guarantees and Limitations" we erroneously claimed in Theorem 5 that no fully randomized priority algorithm for the maximum matching problem can achieve an expected approximation ratio better than $\frac{5}{6}$. This bound and the provided argument hold for *degree-based* randomized priority algorithms. For *fully* randomized priority algorithms we show a $(1-c)$-hardness bound for a small constant c. Thus, the central conclusion that these myopic algorithms cannot guarantee a maximum matching remains valid.

Erratum to: Algorithmica (2017) 77:201–234 DOI 10.1007/s00453-015-0062-2

1 Background

We study greedy algorithms for the maximum matching problem. Let $G = (V, E)$ be an unweighted and undirected graph. A matching $M \subseteq E$ is a selection of edges

B. Besser: Partially supported by DFG SCHN 503/6-1. M. Poloczek: Supported by the Alexander von Humboldt Foundation within the Feodor Lynen program, and in part by NSF Grant CCF-1115256.

The online version of the original article can be found under doi:10.1007/s00453-015-0062-2.

✉ Matthias Poloczek
poloczek@cornell.edu

Bert Besser
besser@thi.cs.uni-frankfurt.de

[1] Institut für Informatik, Goethe-Universität Frankfurt am Main, Frankfurt am Main, Germany

[2] School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

of G such that no two edges in M share a node. If M has largest cardinality among all matchings in G, then M is called a *maximum* matching. An algorithm is called an α-approximation algorithm for the maximum (cardinality) matching problem, if for any graph it finds in polynomial time a matching whose size is at least α times the size of a maximum matching. A randomized α-approximation algorithm is a polynomial time algorithm that obtains a matching whose *expected* cardinality is at least α times the size of an optimal solution. To explore the limitations of "greedy-like" algorithms, we studied *priority algorithms* in [2], a class of algorithms introduced by Borodin et al. [4].

2 Inapproximability Results for Randomized Priority Algorithms

Angelopoulos and Borodin [1] introduced *fully randomized priority algorithms*: these algorithms proceed like adaptive priority algorithms, but may utilize randomness in determining an ordering of data items and in making decisions.

The class of fully randomized algorithms is quite comprehensive, as it contains for instance the algorithms Greedy, MinGreedy, MRG, Ranking, and the KarpSipser algorithm (see also Sect. 1 of our article [2] for an overview). For an introduction to priority algorithms and to the *adaptive priority game* we refer to Sect. 3 in [2].

MinGreedy is an example of a degree-based randomized priority algorithm, following the definition of Borodin et al. [3]: a priority algorithm is *degree-based* if only degrees of vertices are used in defining the ordering of data items. That is, a degree-based priority algorithm submits an ordering of possible degrees in each round, and receives a node of the first degree that exists in the graph. If the algorithm is *randomized* and there are multiple nodes of the first degree in the graph, then the adversary selects one uniformly at random and gives the corresponding data item to the algorithm. For a deterministic algorithm the adversary may break ties arbitrarily.

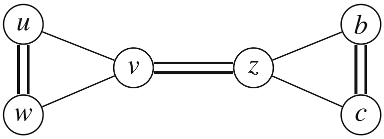
Observe that we have some flexibility in defining whether a requested degree corresponds to a node's degree in the input graph, or is reduced to account for already matched neighbors. Our construction applies for both formulations: to establish the bound we analyze only the first round, when reduced degrees equal original degrees in the input graph.

First we study degree-based randomized priority algorithms, then we consider fully randomized priority algorithms.

Theorem 1 *No* degree-based randomized *priority algorithm, whether greedy or not, can achieve an expected approximation ratio better than* $\frac{5}{6}$ *for the vertex model.*

Proof We apply Yao's Minimax Principle [5]. Thus, we have to construct a hard distribution over input instances, for which we analyze the best deterministic algorithm (that knows the distribution). The distribution corresponds to all the permutations of node labels of the graph depicted in Fig. 1. The graph has a unique maximum matching. The adversary begins by announcing the number of nodes, the number of edges, and the degree sequence. Note that this is not required by the priority framework and further strengthens the bound. We will consider only mistakes made in the first round and assume that the algorithm proceeds optimally afterwards.

Fig. 1 A hard input graph for randomized priority algorithms



First note that if the algorithm decides to isolate the node received in the first round, then it cannot obtain a matching with three edges (which is maximum) but only two edges. Thus, we may assume that the first node will be matched. Furthermore, if the first node is not matched to its optimal neighbor, then again the algorithm obtains two edges and achieves an approximation ratio of $\frac{2}{3}$. On the other hand, if the first node is matched optimally, a maximum matching is obtained by our above assumption.

In the first round, the algorithm receives a data item for a node u of degree two or three, whichever occurs first in the ordering submitted by the algorithm. Observe that each neighbor of u is the optimal mate with same probability because the labeling of the nodes was chosen uniformly at random. Thus, the best strategy for the algorithm is to request no degree three node prior to a degree two node, since the probability of matching the first node optimally decreases with its degree. The bound follows since a node of degree two is matched optimally with probability $\frac{1}{2}$. □

Next we show the bound on fully randomized priority algorithms.

Theorem 2 *No* fully randomized *priority algorithm, whether greedy or not, can achieve an expected approximation ratio better than* $\frac{134}{135}$ *for the vertex model.*

Proof Again we apply Yao's Minimax Principle and utilize the uniform distribution over all graphs corresponding to permutations of node labels in Fig. 1. Before the game starts, the adversary announces the number of nodes, the number of edges, and the degree sequence, thereby reducing the set of possible data items. For the same reason as in the proof of Theorem 1, we may assume that the algorithm matches the node given in the first data item that it receives.

In the first round, recall that the algorithm submits the ordering π of data items without looking at the randomly chosen graph G. We argue that the following event occurs with probability $p > 0$: graph G contains the first data item π_1 in π, *and* the algorithm matches the node corresponding to π_1 to a non-optimal neighbor.

If this event occurs, then the algorithm can only pick one more edge in total, whereas a maximum matching contains three edges. Thus, the expected approximation ratio ρ is at most $p \cdot \frac{2}{3} + (1 - p) \cdot \frac{3}{3} < 1$, as desired.

Note that the probability of a non-optimal matching in the first round is larger than p, since data item π_1 might not belong to G and following data items also have a non-zero probability for a non-optimal matching.

To show that $p > 0$ holds, we distinguish whether π_1 corresponds to a node u of degree two or of degree three. In each case, we count the permutations of node labels that favor the event. Suppose that the algorithm picks v as u's mate. Since π_1 is the first data item of ordering submitted in the first round, node u's neighbors are indistinguishable for the algorithm.

- First assume that we have $\pi_1 = \langle u; v, w \rangle$.
  There are four nodes of degree two that might be labeled u. For each choice, u is matched non-optimally if and only if v has degree three, in which case w is the other node of degree two in the triangle containing u. Now observe that there are 3! ways to label the remaining three nodes.
  Hence the probability for a non-optimal matching is $\frac{4 \cdot 3!}{6!} = \frac{1}{30}$.
- Now assume that $\pi_1 = \langle u; v, w, z \rangle$ holds.
  There are two nodes of degree three which might be labeled u. In order for edge (u, v) to be a non-optimal choice, neighbor v has to be one of two degree-2 nodes in the respective triangle of u. Moreover, for each choice of v there are two ways to label the remaining neighbors of u with w and z, and two ways to label the remaining two nodes in the other triangle.
  Hence the probability for a non-optimal matching is $\frac{2^4}{6!} = \frac{1}{45}$.

The probability p is bounded by the minimum over both cases, i.e. we have $p \geq \frac{1}{45}$. Therefore $\rho \leq \frac{134}{135}$ holds. □

Acknowledgements The authors would like to thank Nicolas Peña and Allan Borodin for pointing them to their mistake.

References

1. Angelopoulos, S., Borodin, A.: Randomized priority algorithms. Theor. Comput. Sci. **411**(26–28), 2542–2558 (2010)
2. Besser, B., Poloczek, M.: Greedy matching: guarantees and limitations. Algorithmica **77**(1), 201–234 (2017)
3. Borodin, A., Boyar, J., Larsen, K.S., Mirmohammadi, N.: Priority algorithms for graph optimization problems. Theor. Comput. Sci. **411**(1), 239–258 (2010)
4. Borodin, A., Nielsen, M.N., Rackoff, C.: (Incremental) priority algorithms. Algorithmica **37**(4), 295–326 (2003)
5. Yao, A.C.C.: Lower bounds by probabilistic arguments. In: Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS), pp. 420–428. IEEE (1983)