Approximation Bounds For Minimum Degree Matching

Bert Besser*

Institut für Informatik, Goethe-Universität Frankfurt am Main, Germany

Abstract. We consider the MinGreedy strategy for Maximum Cardinality Matching. MinGreedy repeatedly selects an edge incident with a node of minimum degree. For graphs of degree at most three we show that the worst case approximation ratio of MinGreedy is $\frac{2}{3}$; this is optimal among adaptive priority algorithms in the vertex model. Furthermore, for graphs of degree at most $\Delta$ we show that MinGreedy computes matchings of size at least $\frac{\Delta-1/2}{2\Delta-2}$ times optimal and no adaptive priority algorithm can guarantee matchings larger than $\frac{\Delta-1}{2\Delta-3}$ times optimal. We conjecture optimality of MinGreedy and prove it for $\Delta$-regular graphs. Even when considering expected approximation ratios of randomized greedy strategies, no better worst case bounds are known for graphs of small degrees.

Keywords: matching, greedy, approximation, priority algorithm.

1 Introduction

In the Maximum Cardinality Matching Problem a node disjoint subset of edges of maximum size is to be determined. Matching problems have many applications, e.g. image feature matching in computer vision or protein structure comparison in computational biology.

A maximum matching can be found in polynomial time, e.g. by the algorithm of Micali and Vazirani [MV80] running in time $O(\sqrt{|V|} \cdot |E| \cdot \alpha(|E|, |V|))$ where $\alpha(|E|, |V|)$ is the inverse Ackermann function [Vaz12]. The algorithm of Mucha and Sankowski [MS04] runs asymptotically faster on dense graphs, its runtime is $O(|V|^\omega)$ where $\omega < 2.38$ is the exponent needed to perform matrix multiplication.

However, there are much faster greedy algorithms that in practice compute very large matchings, even near optimal ones. Very good approximate solutions may already be satisfactory in some applications. If maximum matchings are needed, one can save lots of runtime when feeding large greedy matchings into optimal algorithms which iteratively improve an initial solution.

MinGreedy. The (randomized) MinGreedy algorithm computes a matching $M$ by repeatedly picking edges incident to nodes of currently minimum degree, see Figure 1. MinGreedy can be implemented in linear time $O(|V| + |E|)$.

The experimentally observed approximation performance is quite impressive. Tinhofer [Tin84] observed that on random graphs of varying density MinGreedy performed superior to Greedy (which randomly selects an edge) and

* Partially supported by DFG SCHN 503/6-1.
\[ M = \emptyset \]
repeat until all edges removed from input graph:
  select (random) node \( u \) of minimum non-zero degree
  select (random) neighbor \( v \) of \( u \)
  pick edge \( \{u, v\} \), i.e., set \( M = M \cup \{\{u, v\}\} \)
  remove all edges incident with \( u \) and \( v \) from input graph
return \( M \)

**Fig. 1.** The (randomized) **MinGreedy** algorithm

The **MinGreedy** algorithm, abbreviated MRG in literature, ignores node degrees and repeatedly selects a node and then a neighbor uniformly at random. The expected approximation ratio was shown to be at least \( \frac{1}{2} + \frac{1}{100,000} \) by Aronson et al. [ADFS95].

The random edge algorithm **Greedy** repeatedly selects an edge uniformly at random. For graphs with degrees bounded by \( \Delta \) an expected lower bound on the approximation ratio of \( \frac{\Delta}{2\Delta - 1} \) was shown by Dyer and Frieze [DF91] and later improved by Miller and Pritikin [MP97] to \( \frac{1}{2}(\sqrt{(\Delta - 1)^2 + 1} - \Delta + 2) \). If Greedy prefers edges of degree-1 nodes, the Karp-Sipser algorithm is obtained, which is asymptotically optimal w.h.p. on large sparse random graphs [KS81].

The **Shuffle** algorithm, proposed by Goel and Tripathi [GT12], is an adaptation of the **Ranking** algorithm of Karp et al. [KVV90] to non-bipartite graphs. Shuffle selects a random permutation \( \pi \) of the nodes and repeatedly matches the, according to \( \pi \), first non-isolated node to its first unmatched neighbor. Chan et al. [CCWZ14] showed that Shuffle achieves an approximation ratio of at least \( 2 \cdot (5 - \sqrt{7})/9 \approx 0.523 \).

**Inapproximability.** To show performance bounds for classes of greedy algorithms, Borodin et al. [BNR02] proposed the model of **adaptive priority algorithms**. The model formalizes the greedy nature of an algorithm: while gathering knowledge about the input, irrevocable decisions have to be made to construct a solution.

Davis and Impagliazzo [DI04] introduced the **vertex model** to study adaptive priority algorithms for graph problems. Adaptive priority algorithms in the ver-
tex model, which we call \( \mathcal{APV} \)-algorithms, include e.g. \textsc{MinGreedy} and MRG but implement much more powerful node and edge selection routines. In particular, in each step a node \( v \) and an incident edge is not picked arbitrarily but based on all knowledge already gathered about \( v \) and its neighbors, e.g. is a neighbor matched or unmatched, what is the degree of a neighbor, what are the neighbors of a neighbor, etc.

Despite the apparent strength of \( \mathcal{APV} \)-algorithms Poloczek \cite{Poloczek2012} constructed rather simple graphs (degrees are at most three) with worst case approximation ratio at most \( \frac{4}{3} \). For graphs with arbitrarily large degree Besser and Poloczek (see Poloczek’s PhD thesis \cite{Poloczek2012}) showed that no \( \mathcal{APV} \)-algorithm achieves worst case approximation ratio better than \( \frac{1}{2} + o(1) \). Poloczek also studied randomized priority algorithms and showed an inapproximability bound of \( \frac{5}{6} \) \cite{Poloczek2012}.

\textbf{Contributions.} From now on we reserve the name \textsc{MinGreedy} for the deterministic version of \textsc{MinGreedy} in which a node of minimum degree and an incident edge is picked by a worst case adversary.

We show that \textsc{MinGreedy} approximates an optimal matching within a factor of \( \frac{2}{3} \) for graphs of degree at most three and within a factor of at most \( \frac{\Delta - 1}{\Delta - 3} \) for \( \Delta \)-regular graphs.

We also show that this worst case approximation performance is optimal for (deterministic) \( \mathcal{APV} \)-algorithms in both cases. In particular, we improve the construction of Besser and Poloczek given in \cite{Poloczek2012} and present hard input instances of degree at most \( \Delta \) for which an \( \mathcal{APV} \)-algorithm computes a matching of size at most \( \frac{\Delta - 1}{2\Delta - 3} \) times optimal.

If degrees are bounded by \( \Delta \), we show that \textsc{MinGreedy} achieves an approximation ratio of at least \( \frac{\Delta - 1}{2\Delta - 2} \). We conjecture optimality of \textsc{MinGreedy} among \( \mathcal{APV} \)-algorithms.

Our worst case performance guarantees are stronger than the best known worst case bounds on the expected performance of MRG (\( \frac{1}{2} + \frac{1}{2000000} \)) and \textsc{Shufle} (\( \approx 0.523 \)), if \( \Delta \) is small, and of \textsc{Greedy} (\( \approx 0.618 \) for \( \Delta = 3 \)), for all \( \Delta \).

\textbf{Techniques.} For our performance guarantees for \textsc{MinGreedy} we study the matching graph composed of the edges of a matching \( M \), computed by \textsc{MinGreedy}, and of a maximum matching \( M^* \). The connected components are alternating paths and cycles. Only paths of length three have poor “local” approximation ratio (of \( M \)-edges to \( M^* \)-edges). To obtain a global performance guarantee, we balance local approximation ratios by transferring “\( M \)-funds” from rich to poor components using edges of the input graph.

Incorporating the properties of \textsc{MinGreedy} within an amortized analysis is our technical contribution.

\textbf{Overview.} In Section \ref{sec:charging} we present the charging scheme used to prove the performance guarantees for \textsc{MinGreedy}. In Section \ref{sec:23} we show our \( \frac{2}{3} \) bound for graphs of degree at most three respectively the \( \frac{\Delta - 1}{\Delta - 3} \) bound for \( \Delta \)-regular graphs. For graphs of degree at most \( \Delta \) we present in Section \ref{sec:general} our performance guarantee of \( \frac{\Delta - 1/2}{2\Delta - 2} \). Our inapproximability results for \( \mathcal{APV} \)-algorithms are given in Section \ref{sec:inapproximability}.

3
2 The Charging Scheme

The Matching Graph. Let $G = (V, E)$ be a connected graph, $M^*$ a maximum matching in $G$ and $M$ a matching computed by MINGREEDY when applied to input $G$. To analyze the worst case approximation ratio of MINGREEDY we investigate the graph $H = (V, M \cup M^*)$.

The connected components of $H$ are paths and cycles composed of edges of $M$ and $M^*$. For example, $H$ contains so-called $(M\text{-})augmenting paths$: an augmenting path $X$ has $m_X$ edges of $M$ and $m_X^* = m_X + 1$ edges of $M^*$ and starts and ends with an $M^*$-edge:

Other connected components of $H$ are edges of $M \cap M^*$, which we call singletons. For a singleton $X$ we have $m_X = m_X^* = 1$. We may focus on these two component types:

**Proposition 1.** There is a maximum matching $M^*$ in $G$ such that each connected component of $H$ is a singleton or an augmenting path.

**Proof.** Let $M'$ be a maximum matching in $G = (V, E)$ and let MINGREEDY compute the matching $M$. We show how to transform $M'$ into $M^*$. The connected components of the graph $(V, M \cup M')$ are singletons and alternating paths and cycles, where a path starts and ends with an $M$-edge or $M'$-edge and a cycle does not have path endpoints. To prove the statement we eliminate paths starting and ending both with an $M$-edge, paths starting and ending with different types of edges, and cycles. There is no path $X$ of the first type, since if there was, then we could replace the $m_X'$ many $M'$-edges of $X$ with the $m_X' + 1$ many $M$-edges of $X$ to obtain a matching larger than $M'$. In a component $X$ of the latter types we replace the $M'$-edges of $X$ with the $M$-edges of $X$: component $X$ is replaced by $m_X$ many singletons.

**Local Approximation Ratios.** To bound the approximation ratio

$$\alpha = \frac{|M|}{|M^*|}$$

of MINGREEDY, our approach is to bound local approximation ratios

$$\alpha_X = \frac{m_X}{m_X^*},$$

of components $X$ of $H$. Observe that a $\frac{1}{2}$-path $X$

$$\text{--} \text{--} \# \text{--} \text{--} \text{--} \text{--} \text{--} \text{--},$$
i.e., an augmenting path with \( m_X = 1 \) edge of \( M \) and \( m_X^* = 2 \) edges of \( M^* \), has local approximation ratio \( \alpha_X = \frac{1}{2} \), while all other components have local approximation ratios at least \( \frac{2}{3} \). In particular, a singleton \( X \) has optimal local approximation ratio \( \alpha_X = 1 \). We have to balance local approximation ratios. For any component \( X \), we say that \( X \) has \( M \)-funds \( m_X \) and we introduce a change \( t_X \) to the \( M \)-funds of \( X \) such that the changed local approximation ratio of \( X \) is lower bounded by
\[
\alpha_X = \frac{m_X + t_X}{m_X^*} \geq \beta
\]
for appropriately chosen \( \beta \). If \( \sum_X t_X = 0 \) holds, then the total \( M \)-funds \( \sum_X m_X + t_X = |M| = \sum_X m_X = |M| \) are unchanged and hence MinGreedy achieves approximation ratio at least
\[
\alpha = |M|/|M^*| = \left( \sum_X m_X + t_X \right) / |M^*| \geq \left( \sum_X \beta m_X^* \right) / |M^*| = \beta
\]

**Transferring \( M \)-Funds.** The idea is to transfer \( M \)-funds from rich components to poor components: as an example, we could transfer \( M \)-funds from a singleton to a \( \frac{1}{2} \)-path. But which components should be involved in a transfer? Observe that an augmenting path endpoint \( w \) is detrimental to its augmenting path \( X \), since \( w \) decreases the local approximation ratio of \( X \). So a transfer should push \( M \)-funds towards poor \( w \) from a rich \( M \)-covered node. Since the \( G \)-neighbors of \( w \) are \( M \)-covered (otherwise \( M \) would not be maximal), our approach is to move \( M \)-funds to \( w \) from \( G \)-neighbors of \( w \) which belong to other components: \( M \)-funds are moved over certain edges in
\[
C := E \setminus (M \cup M^*)
\]
which connect the components of \( H \), see Figure 2 for an illustration. We verify

that a poor component, in particular a \( \frac{1}{2} \)-path, is able to receive \( M \)-funds:

**Proposition 2.** An augmenting path endpoint \( w \) is incident with \( C \)-edge, since the degree of \( w \) in \( G \) is at least
\[
d_G(w) \geq 2.
\]

**Proof.** When MinGreedy picks the first \( M \)-edge of the augmenting path of \( w \), a node of degree at least two is selected (with an incident \( M \)-edge and \( M^* \)-edge). So \( w \) also has degree at least two and is incident not only with its \( M^* \)-edge. \( \square \)
When transferring \( M \)-funds over all \( C \)-edges we face the danger of augmenting path endpoints having large degree and drawing very large amounts of \( M \)-funds: rich components might become poor and now have too small local approximation ratio. The following definition prevents high degree path endpoints from wasting \( M \)-funds.

**Definition 1.** Let \( w \) be an augmenting path endpoint. Edge \( \{v, w\} \in C \) is a transfer if in the step of \textsc{MinGreedy} matching \( v \) the degree of \( w \) drops to at most \( d(w) \leq \Delta - 2 \).

We frequently denote a transfer \( \{v, w\} \) as \( (v, w) \) to stress its direction. In order to refer to transfers from/to a given component \( X \), we also call \( (v, w) \) a debit to \( v \) and a credit to \( w \). I.e., debits to a component are transfers directed from its \( M \)-covered nodes, credits to an augmenting path are transfers directed to its path endpoints.

**Bounding Local Approximation Ratios.** Let \( X \) be a component of \( H \) and denote the numbers of debits and credits to nodes of \( X \) by \( d_X \) respectively \( c_X \). We call \( d_X - c_X \) the total debits to \( X \). Given an upper bound on the debits to \( X \) and a lower bound on the credits to \( X \), we obtain total debits to \( X \) of at most

\[
d_X - c_X \leq T_X.
\]

Whenever we move \( M \)-funds over an edge \( \{v, w\} \in C \), we transfer an amount \( \theta \) of \( M \)-funds to the augmenting path endpoint \( w \). Hence the local approximation ratio of \( X \) is at least

\[
\alpha_X = \frac{m_X - \theta d_X + \theta c_X}{m_X} = \frac{m_X - \theta(d_X - c_X)}{m_X} \geq \frac{m_X - \theta T_X}{m_X}.
\]

In the analysis we find \( T_X \) and \( \theta \) such that \( \alpha_X \geq \beta \) for all components. Hence \textsc{MinGreedy} computes matchings of size at least \( \beta \) times optimal.

### 3 Maximum Degree Three and \( \Delta \)-Regular Graphs

**Theorem 1.** If \( \Delta = 3 \) or \( G \) is a \( \Delta \)-regular graph, then \textsc{MinGreedy} computes a matching of size at least \( \frac{\Delta - 3}{\Delta - 2} \) times optimal.

To prove the theorem we lower bound local approximation ratios of all components by \( \frac{\Delta - 3}{\Delta - 2} \). Therefore, we claim credit and debit bounds for all components, then show how to choose \( \theta \) and prove the claimed bounds in Section 3.1.

Recall that there are two types of components in \( H \), namely singletons and augmenting paths. Let \( X \) be a singleton. Since \( X \) has only \( M \)-covered nodes, by definition \( X \) does not get credits, i.e., \( c_X = 0 \). Since a node \( v \) of \( X \) has degree at most \( \Delta \), there are at most \( \Delta - 1 \) debits to \( v \). We demand that at least two debits are "missing", i.e., that \( X \) has at most \( d_X \leq 2(\Delta - 1) - 2 \) debits. Hence total debits to \( X \) and the local approximation ratio of \( X \) are bounded as follows:
\[ d_X - c_X = d_X \leq 2(\Delta - 1) - 2 \quad (1) \]
\[ \alpha_X = \frac{1 - \theta(d_X - c_X)}{1} \geq 1 - 2\theta(\Delta - 2) . \quad (1') \]

Note that a \( \frac{\Delta - 1}{2\Delta - 3} \) performance guarantee holds if we choose \( \theta \leq \frac{1}{2(2\Delta - 3)} \).

Let \( X \) be an augmenting path. By Proposition 2, a path endpoint \( w \) of \( X \) is able to receive credits. We show that \( w \) receives as many as \( c_w \geq \Delta - 2 \) credits. Since \( X \) has two path endpoints we get

\[ c_X \geq 2(\Delta - 2) . \quad (3) \]

An \( M \)-covered node \( v \) of \( X \) has at most \( \Delta - 2 \) debits, since the degree of \( v \) is at most \( \Delta \) and \( v \) is also incident to its \( M \)-edge and \( M^* \)-edge, for which we do not move \( M \)-funds. So \( d_X \leq 2m_X(\Delta - 2) \) holds. Analyzing MinGreedy, we give stronger total debit bounds of at most

\[ d_X - c_X \leq 2m_X(\Delta - 2) - 2(\Delta - 2) - 2 \quad (4) \]

Note that in particular two missing debits are enough to satisfy (4). As a consequence, it holds that

\[ \alpha_X = \frac{m_X - \theta(d_X - c_X)}{m^*_X} \geq \frac{m_X - 2\theta m_X(\Delta - 2) + 2\theta(\Delta - 1)}{m_X + 1} \]
\[ = 1 - 2\theta(\Delta - 2) + \frac{2\theta(2\Delta - 3) - 1}{m_X + 1} , \]

which, if we choose \( \theta = \frac{1}{2(2\Delta - 3)} \), is equivalent to

\[ \alpha_X \geq 1 - 2\theta(\Delta - 2) . \quad (4') \]

Combining (1) and (4) we obtain our claimed \( \frac{\Delta - 1}{2\Delta - 3} \) performance guarantee for MinGreedy, since local approximation ratios of singletons \( X \) as well as augmenting paths \( X \) are lower bounded by

\[ \alpha_X \geq 1 - 2\theta(\Delta - 2) = 1 - \frac{2(\Delta - 2)}{2(2\Delta - 3)} = \frac{\Delta - 1}{2\Delta - 3} . \]
3.1 Credit and Debit Bounds

Before we verify the total debit bounds claimed in (1) and (4), we prepare the analysis with a simple but important observation.

**Observation 1.** A transfer \((v, w)\) is removed from \(G\) in the step of \textsc{MinGreedy} that matches \(v\).

*Proof.* The augmenting path endpoint \(w\) is never matched. \(\square\)

**Credits.** Let \(X\) be an augmenting path and \(w\) a path endpoint of \(X\). In Lemma 1 we prove the claimed credit bounds (2) and (3) for \(w\) respectively \(X\). Intuitively, the higher the degree of \(w\) is, the more credits receives \(w\), since in many steps matching transfer neighbors of \(w\) the degree of \(w\) drops to \(d(w) \leq \Delta - 2\).

**Lemma 1.** Let degrees be bounded by \(\Delta\). The number of credits \(c_w\) to an augmenting path endpoint \(w\) is at least \((d_G(w) - 1, \Delta - 2) \geq 1\).

*Proof.* By Proposition 2 we have \(d_G(w) \geq 2\), hence \(\min \{d_G(w) - 1, \Delta - 2\} \geq 1\).

If \(d_G(w) \leq \Delta - 1\) holds, then any time the degree of \(w\) drops, it drops to at most \(\Delta - 2\). Therefore all \(C\)-edges incident with \(w\) are credits.

If \(d_G(w) = \Delta\) holds, then after \(d(w)\) drops to at most \(d(w) = \Delta - 1\), all \(C\)-edges of \(w\) removed later are credits: at most one \(C\)-edge of \(w\) is not a credit. \(\square\)

**Debits.** To verify the total debit bounds (1) and (4), we also have to show that a component \(X\) has missing debits, i.e., that \(M\)-covered nodes of \(X\) do not have maximum numbers of debits, be it due to low degree in \(G\) or due to incident \(C\)-edges which are not transfers. As discussed, two missing debits are sufficient. The following steps of \textsc{MinGreedy} are crucial in the analysis of missing debits: a step of \textsc{MinGreedy} creates a component \(X\) if the first \(M\)-edge of \(X\) is picked.

**Lemma 2.** Let degrees be bounded by \(\Delta\). Assume that a component \(X\) is created when a node \(u\) of degree \(d(u)\) is selected. Two debits to \(X\) are missing if
(a) \(d(u) \leq \Delta - 2\) or
(b) \(d(u) = \Delta\) or
(c) \(\Delta \geq 4\) or
(d) \(X\) is a singleton.

*Proof.* Assume that at most one debit to a node of \(X\) is missing. We show a contradiction. Let \(X\) be created in step \(s\) and \(u\) be matched to \(v\). By assumption, at least one of \(u, v\) has a debit.

We prove (a). If \(d(u) \leq \Delta - 2\) holds at step \(s\), then by Observation 1, node \(u\) has two missing debits. A contradiction.

For (b) to (d) we may focus on \(d(u) \geq \Delta - 1\) by (a). Here is a sketch of the argument. Consider a debit \((x, w)\) to \(x \in \{u, v\}\). By Observation 1, step \(s\) removes \((x, w)\) from \(G\) and, by definition of transfers, after step \(s\) the degree of \(w\) is \(d(w) \leq \Delta - 2\). In particular, after step \(s\) the degree of \(w\) is smaller than the degree of \(u\) before step \(s\). Here comes the crucial claim:
at step $s + 1$, there is an augmenting path endpoint $w$ which is not isolated and has minimum degree $d(w)$. Since $w$ is an augmenting path endpoint, node $w$ is not selected by MinGreedy in step $s + 1$. Hence we obtain a contradiction if at step $s + 1$ all minimum degree nodes are augmenting path endpoints. So assume that a node $y \neq w$ is selected in step $s + 1$ with minimum degree $d(y) = d(w) \leq \Delta - 2$. Since the degree of $y$ was at least $\Delta - 1$ before step $s$, in step $s$ edges of $C$ connecting $y$ with $u$ or $v$ are removed. But $u, v, y$ are $M$-covered, hence the removed edges are not debits to $u, v$. We obtain a contradiction if we find at least two missing debits.

We prove (b). Assume that $d(u) = \Delta$ holds at step $s$. Since in step $s$ at most two edges incident with $w$ are removed and $(x, w)$ is a transfer, we have $d(w) = \Delta - 2$ at step $s + 1$. In particular, our claim holds: node $w$ is not isolated, since $d(w) = \Delta - 2 \geq 1$, and $w$ has minimum degree. So let $y \neq w$ be the node selected in step $s + 1$. The degree of $y$ also drops from $\Delta$ to $\Delta - 2$ in step $s$ when incident edges $\{u, y\}, \{v, y\}$ are removed. If $y$ is not a node of $X$, then both $\{u, y\}, \{v, y\}$ are $C$-edges but they are not debits, since $u, v, y$ are $M$-covered, see Figure 3. Hence both $u, v$ have a missing debit and we have found a contradiction. If $y$ is a node of $X$, then at least one of $\{u, y\}, \{v, y\}$, say $\{u, y\}$, is a $C$-edge. Since both $u, y$ are $M$-covered, edge $\{u, y\}$ is not a transfer, see Figure 4. Hence both $u, y$ have a missing debit, a contradiction.

![Fig. 3. An example where node $y$ is not a node of $X$.](image1)

![Fig. 4. An example where node $y$ is a node of $X$.](image2)

We prove (c). Since (a) and (b) apply to $\Delta \geq 4$, it suffices to prove the case that $d(u) = \Delta - 1$ holds at step $s$. Recall that by Observation 1 node $u$ has a missing debit due to its low degree before step $s$. We find an additional missing debit by investigating step $s + 1$. Our claim holds: since $(x, v)$ is a transfer, after step $s$ the degree $d(w) \leq \Delta - 2$ is smaller than that of $u$ before step $s$ and using $\Delta \geq 4$ we get $d(w) \geq \Delta - 3 \geq 1$, since at most two edges incident with $w$ are removed in step $s$. Let $y \neq w$ be the neighbor of $u$ or $v$ being selected with degree $d(y) \leq \Delta - 2$ next. Recall that step $s$ removes an edge $\{x, y\}$ with $x \in \{u, v\}$. If $y$ is not a node of $X$, then $\{x, y\}$ is a $C$-edge. But since both $x, y$ are $M$-covered, edge $\{x, y\}$ is not a transfer and we have found an additional missing debit to one of $u, v$. A contradiction. Now assume that $y$ is a node of $X$. At step $s + 1$ node $y$ is incident with its $M$-edge, maybe with debits and possibly with its $M^*$-edge. Observe that no matter if the $M^*$-edge of $y$ is already removed, node $y$ has a missing debit since $d(y) \leq \Delta - 2$. We have found an additional missing debit and obtain another contradiction.
Fig. 5. Creation of a singleton with three debits ($\Delta = 3$). Gray nodes are path endpoints and their degrees drop, dashed nodes have degree one after creation.

We prove \([d]\). By \([a]\) to \([c]\) it suffices to prove the case that degrees are bounded by $\Delta = 3$ and $d(u) = \Delta - 1 = 2$ holds. Since $u$ has a missing debit and at most one debit is missing for $X$ we get that $u$ has exactly one debit, say to $u'$, and $v$ has exactly two debits, say to $v', v''$. Note that either $u' \in \{v', v''\}$ or $u' \not\in \{v', v''\}$. In step $s$ the degrees of $u', v', v''$ drop from at least $d(u) = 2$ to at most $\Delta - 2 = 1$, by definition of transfers. No other degrees drop, see Figure 5.

Why does our claim hold? Since $\{|u', v', v''\}| \geq 2$ but $u$ is incident with only one of $u', v', v''$, there is $w \in \{u', v', v''\}$ which is incident only with $v$, hence $w$ is not isolated and has minimum degree $d(w) = 1$ after step $s$. Hence one of $u', v', v''$ is selected in step $s + 1$, since no other degrees dropped. A contradiction, since $u', v', v''$ are augmenting path endpoints.

Recall that to prove Theorem 1 we have to verify the total debit bounds claimed in (1) and (4) for singletons respectively augmenting paths. Our performance guarantee is proven for $\Delta$-regular graphs with $\Delta \geq 4$: a singleton $X$ has two missing debits by Lemma 2 (d), hence total debits to $X$ are at most $d_X - c_X = d_X \leq 2(\Delta - 1) - 2$ as required in (1); an augmenting path $X$ has two missing debits by Lemma 2 (c) and $c_X \geq 2(\Delta - 2)$ credits by (3), so total debits are at most $d_X - c_X \leq 2m_X(\Delta - 2) - 2 - 2(\Delta - 2)$ as claimed in (4).

To prove Theorem 4 it remains to consider graphs of degree at most $\Delta = 3$. Lemma 2 (d) shows that a singleton $X$ has total debits at most $d_X - c_X = d_X \leq 2(\Delta - 1) - 2 = 2$ as required in (1). Thus we focus on an augmenting path $X$ for the rest of the analysis. By (3), the two path endpoints of $X$ receive at least $c_X \geq 2$ credits. Observe that (1) is satisfied with total debits at most $d_X - c_X \leq 2m_X - 4$ if $X$ has two missing debits, which is true by Lemma 2 (a) and (b) if $X$ is created in a step selecting a node $u$ of degree $d(u) \neq \Delta - 1 = 2$. So assume that $d(u) = 2$ holds. By Observation 4, a debit to $u$ is missing. If there is an additional debit missing, then we are done. However, each of the $2m_X - 1$ other nodes of $X$ can have a debit, see e.g. Figure 6. Lemma 3 concludes our analysis since total debits are bounded by $d_X - c_X \leq 2m_X - 4$ as required in (4).

Fig. 6. An augmenting path with $d_X = 2m_X - 1$ debits ($\Delta = 3$).
Lemma 3. Let degrees be bounded by $\Delta = 3$. Let $X$ be an augmenting path with $d_X = 2m_X - 1$ debits. At least $c_X \geq 3$ credits are given to $X$.

Proof. Assume that $c_X \leq 2$ holds. We show a contradiction. Recall that by (3) two credits to $X$ are minimum and that by (2) each path endpoint of $X$ has exactly one credit.

Since $X$ has $d_X = 2m_X - 1$ debits, by Lemma 2 (a) and (b) the creation step of $X$ selects a node $u$ with degree $d(u) = \Delta - 1 = 2$. So $u$ has no debit and all other $M$-covered nodes of $X$ have a debit. Assume that $u$ is matched with $v$.

Fig. 7. Creation of a $\frac{1}{2}$-path. Gray nodes are path endpoints and their degrees drop, dashed nodes have degree one after creation.

Assume that $m_X = 1$, i.e., $X$ is a $\frac{1}{2}$-path. In the creation step of $X$ only degrees of the destination node $w_v$ of the debit to $v$ and of both path-endpoints $w, w'$ of $X$ drop, see gray nodes in Figure 7. In particular, their degrees drop to at most one, by definition of transfers, and might now be minimum degrees. At least one of $d(w_v), d(w), d(w')$ drops to exactly one (dashed nodes) and thus one of $w, w, w'$ is selected next. We obtain a contradiction since $w, w, w'$ are augmenting path endpoints.

Fig. 8. Removing the last $M^*$-edge at an end of an augmenting path.

Now assume that $m_X \geq 2$ holds. Consider the step $s'$ of MINGREEDY in which the second (and last) $M^*$-edge incident with a path-endpoint $w$ of $X$ is removed from $G$, say when $u'$ is selected and matched with $v'$. Since by assumption $u'$ is incident to a debit and to its $M$-edge $\{u', v'\}$, it holds that $d(u') \geq 2$. But $d(u') = 3$ does not hold, since then $d(w) = 3$ and the $M^*$-edge of $w$ is removed first from $G$: node $w$ has two credits, i.e., one more than assumed. We get $d(u') = 2$. Only degrees of $w$ and the destination nodes $w_{u'}, w_{v'}$ of the debits to $u'$ respectively $v'$ drop, see Figure 8. In particular, their degrees drop to at most one and might now be minimum. At least one of $d(w), d(w_{u'}), d(w_{v'})$ drops to exactly one and hence one of $w, w_{u'}, w_{v'}$ is selected next. We obtain a contradiction since $w, w_{u'}, w_{v'}$ are augmenting path endpoints. \qed
4 Maximum Degree $\Delta$

Theorem 2. MinGreedy achieves an approximation ratio of at least $\frac{\Delta-1/2}{2\Delta-2}$ on graphs with degrees at most $\Delta$.

The Issue. Why does our performance guarantee not match our conjectured approximation ratio of $\frac{\Delta-1}{2\Delta-3}$? Recall that we required in (4) that a $\frac{1}{2}$-path $X$ has total debits of at least

$$d_X - c_X \leq 2(\Delta - 2) - 2(\Delta - 2) - 2 = -2,$$

i.e., that $X$ has at least two more credits than debits. An issue for graphs with degrees bounded by $\Delta$ is that a $\frac{1}{2}$-path $X$ may have fewer excess credits, see e.g. Figure 9 where the $\frac{1}{2}$-path $X$ has only one excess credit (for now, ignore the gray transfer). But with only one excess credit, the local approximation ratio of $X$ is smaller than $\frac{\Delta-1}{2\Delta-3}$. How can $X$ be cured? Since in Section 3, a $\frac{1}{2}$-path received sufficient $M$-funds only over transfer edges contained in $G$, the curious reader might wonder: can $G$ be altered such that $M^*$ and $M$ are unchanged but there are enough transfer edges? We only note here, that there are more complex examples of too few transfer edges for which this approach does not succeed.

![Fig. 9. A $\frac{1}{2}$-path $X$ with only one credit from another component. $M$-edges are picked in order indicated by small numbers, i.e., from left to right, selecting fat nodes. The gray transfer is not an edge of $G$ and does not effect computations of MinGreedy.](image)

Our Approach. Since $X$ has only one excess credit, we have to double the amount of $M$-funds transferred to $X$. But doubling $\theta$ also doubles $M$-funds drawn from other components: our total debit bounds given for e.g. singletons would now be too large to obtain good local approximation ratios. Therefore we double credited $M$-funds only for $\frac{1}{2}$-paths like $X$. We do this by means of an additional fake transfer to $w$, which does not exist in $G$ and does not have impact on an execution of MinGreedy. In the example of Figure 9, the fake credit to $w$ comes from $u$ and is drawn gray.

But still, fake transfers are no guarantee for at least two more credits than debits to a $\frac{1}{2}$-path. The $\frac{1}{2}$-path $X$ in Figure 10 has only four credits, which is not enough to compensate for its $\Delta-2$ debits. Our strategy is therefore to cancel all transfers debiting $X$: if $v$ is a node of $X$ and $(v, w)$ is a debit to $X$, then $(v, w)$ does not transfer $M$-funds anymore. We only note here that configurations of too many debits may occur also for longer augmenting paths. Therefore we proceed...
Fig. 10. A $\frac{1}{2}$-path $X$ with too few credits. Dashed transfers are canceled.

analogously for all augmenting paths $X$ and cancel transfers debiting nodes of the $M$-edge of $X$ which is picked first by $\text{MIGREEDY}$, i.e., at creation of $X$.

We will show that a $\frac{1}{2}$-path $X$ has at least one uncanceled credit. Since all debits to $X$ are canceled, it suffices to add at most one fake credit to $X$:

$$d_X - c_X \leq 0 - 1 - 1 = -2$$

holds and $X$ has two excess credits, as desired.

4.1 Increased Total Debits

Of course, there is the danger of adding too many fake transfers and other components having too many debits. Indeed, our total debit bounds in (1) and (4) for singletons respectively augmenting paths have become invalid. In our analysis we show slightly increased total debit bounds, in particular, our new bounds are exactly one larger (and are tight): a singleton $X$ can have total debits up to

$$d_X - c_X = d_X \leq 2(\Delta - 1) - 2 + 1$$

and total debits to an augmenting path $X$ with $m_X \geq 2$ are bounded by at most

$$d_X - c_X \leq 2m_X(\Delta - 2) - 2(\Delta - 2) - 2 + 1.$$  \hfill (7)

**Optimizing Transferred $M$-Funds.** Unfortunately, (6) and (7) only allow for trivial bounds on local approximation ratios, as can be seen for a singleton $X$ with maximum debits $d_X = 2(\Delta - 2) + 1$ (recall that $\theta = \frac{1}{2(2\Delta - 3)}$):

$$\alpha_X = \frac{1 - \theta(d_X - c_X)}{1} = 1 - \frac{2(\Delta - 2) + 1}{2(2\Delta - 3)} = \frac{1}{2}.$$  

So, to increase local approximation ratios of singletons and long augmenting paths we have to decrease $\theta$. Of course we have to take care that local approximation ratios of $\frac{1}{2}$-paths are not decreased too much. What $\theta$ should we choose? Parametrized by $\theta$, we lower bound local approximation ratios for all components and then optimize $\theta$. 

13
1/2-paths: \[ \alpha_X = \frac{1 - \theta(d_X - c_X)}{2} \geq \frac{1 + 2\theta}{2} = \frac{1}{2} + \theta \tag{5'} \]

singletons: \[ \alpha_X = \frac{1 - \theta(d_X - c_X)}{1} \geq 1 - 2\theta(\Delta - 2) - \theta \tag{6'} \]

augmenting paths: \[
\alpha_X = \frac{m_X - \theta(d_X - c_X)}{m_X + 1} \\
\geq \frac{m_X - \theta(1 + 2m_X(\Delta - 2) - 2(\Delta - 1))}{m_X + 1} \\
= \frac{m_X(1 - 2\theta(\Delta - 2)) - \theta \cdot (1 - 2(\Delta - 1))}{m_X + 1} \\
= 1 - 2\theta(\Delta - 2) + \frac{2\theta(2\Delta - 3) - 1 - \theta}{m_X + 1} \tag{7'}
\]

Recall that in the last bound for augmenting paths we have \( m_X \geq 2 \). So if we choose \( \theta \geq \frac{1}{2(2\Delta - 2)} \), then \( 2\theta(2\Delta - 3) - 1 \geq -2\theta \) holds and we can simplify the bound to

\[ \alpha_X \geq 1 - 2\theta(\Delta - 2) - \frac{3\theta}{3} = 1 - 2\theta(\Delta - 2) - \theta \tag{7} \]

Combining (5') to (7'), we set \( \frac{1}{2} + \theta = 1 - 2\theta(\Delta - 2) - \theta \) and obtain \( \theta = \frac{1}{2(2\Delta - 2)} \).

Hence the local approximation ratio of any component \( X \) is lower bounded by

\[ \alpha_X \geq \frac{1}{2} + \theta = \frac{1}{2} + \frac{1}{2(2\Delta - 2)} = \frac{2\Delta - 1}{2(2\Delta - 2)} = \frac{\Delta - 1/2}{2\Delta - 2}, \]

which proves our claimed performance guarantee for MinGreedy.

4.2 Fake Transfers

To show the total debit bounds claimed for singletons and augmenting paths in (6) respectively (7), we first have to develop the definition of fake transfers. Therefore we study the properties of 1/2-paths for which fake credits need to be added.

We set up an example to accompany our discussion. In Figure [11], steps of MinGreedy are indicated by small numbers besides picked edges: in the second step MinGreedy creates the 1/2-path \( X \) and in the third step creates the augmenting path \( Y \); in the fifth step MinGreedy selects node \( u \) and picks the \( M \)-edge of a singleton. Following our approach, we cancel debits to \( X \) and to the \( M \)-edge of \( Y \) picked at creation of \( Y \). Since \( X \) has only one uncanceled incoming transfer \((v, w)\), a fake transfer \((u, w)\) is added. Note that when the degree of the path endpoint \( w \) of \( X \) drops to \( d(w) = 1 \) MinGreedy creates the augmenting path \( Y \). Note also, that before \( w \) is isolated when MinGreedy selects \( u \), the degree of \( u \) drops to \( d(u) = 1 \) as well. (In Figure [9], too, the degrees of \( w, u \) drop to \( d(w) = d(u) = 1 \) before \( u \) is selected and \( w \) is isolated. There however, both degrees drop to one when MinGreedy creates the 1/2-path \( X \).)
Lemma 4. Let \( X \) be a \( \frac{1}{2} \)-path and \( w, w' \) its path endpoints. Consider the step \( s \) which picks the \( M \)-edge of \( X \). At least one of \( w, w' \) is not isolated after step \( s \).

Proof. Assume that step \( s \) isolates both \( w, w' \). All edges being removed from \( G \) are between the matched nodes and \( w, w' \), in particular the picked edge connects the matched nodes. It is easy to check that one of \( w, w' \) has smaller degree than both matched nodes, a contradiction to the definition of \( \text{MinGreedy} \).

Observe that after the \( M \)-edge of a \( \frac{1}{2} \)-path \( X \) is picked a non-isolated path endpoint \( w \) of \( X \) has lost its \( M^* \)-edge and is incident only to \( C \)-edges. As a consequence of the next result, node \( w \) has incoming credits.

Lemma 5. Let \( w \) be a path endpoint of an augmenting path \( X \) and assume that \( w \) is incident only to \( C \)-edges. Consider the step \( s \) which isolates \( w \) and assume that step \( s \) picks an \( M \)-edge of a component other than \( X \). If at step \( s \) the degree of \( w \) is \( d(w) = i \), then \( w \) has \( i \) uncanceled credits from nodes matched by step \( s \).

Proof. We show that step \( s \) does not create an augmenting path. Then the statement follows since transfers are canceled only for steps creating augmenting paths. Observe that the degree of \( w \) is at most \( d(w) \leq 2 \), since \( w \) is being isolated and at most two incident edges are removed. Hence step \( s \) selects a node \( u \) of degree at most \( d(u) \leq 2 \).

If \( d(u) = 1 \), then \( u \) does not have both an incident \( M \)-edge and \( M^* \)-edge and hence step \( s \) does not create an augmenting path. Since \( w \) gets isolated but is not connected to \( u \), the other matched node is connected to \( w \) by exactly one transfer. If \( d(u) = 2 \), then \( d(w) = 2 \). Since \( u \) is not a node of \( X \) and \( u \) is connected to \( w \), node \( u \) is not incident to an \( M \)-edge and an \( M^* \)-edge as would be needed to create an augmenting path. Here, both \( u, v \) are connected to \( w \) by a transfer.

We apply Lemmas 4 and 5 to analyze the exact configuration of a \( \frac{1}{2} \)-path \( X \) in need for a fake credit. Recall our claim that \( X \) has exactly one uncanceled credit, say \( (v, w) \) to the path endpoint \( w \) of \( X \). Let \( w' \) be the other path endpoints of \( X \) and assume that step \( s \) picks the \( M \)-edge of \( X \). First, we verify our claim that one fake transfer is sufficient to cure \( X \).
(c1) $X$ has at least one uncanceled credit; by Lemma 4 one of $w, w'$, say $w$, is not isolated by step $s$ and by Lemma 5 node $w$ has an uncanceled credit.

Next we investigate the steps isolating the path endpoints $w, w'$ of $X$.

(c2) One of $w, w'$, say $w'$, is isolated in step $s$: if $w'$ is not being isolated then also $w'$ has an uncanceled credit by Lemma 5.

(c3) Node $w$ is isolated in a step $s' > s$ picking an $M$-edge of another component: node $w$ is not isolated after the only $M$-edge of $X$ is picked.

(c4) At step $s' > s$ which isolates $w$ we have $d(w) = 1$: if $d(w) = 2$ then $w$ has two uncanceled credits by Lemma 5.

What are the steps leading to $w$ getting isolated eventually?

(c5) If the degree of $w$ is $d(w) > 1$ after step $s$, then until step $s'$ the degree of $w$ drops to $d(w) = 1$ only in steps creating augmenting paths: since after step $s$ only $C$-edges incident to $w$ are removed, other kinds of steps produce more uncanceled credits to $w$.

(c6) We stress that the degree of $w$ drops to $d(w) = 1$ in a step creating an augmenting path: either the $M$-edge of $X$ is picked (Figure 9) or by (c5) another augmenting path created (Figure 11).

Assume that step $s'$ selects node $u$ and matches $u$ with $v$.

(c7) Node $u$ is selected when it has degree $d(u) = 1$: node $u$ is selected when $d(u) \leq d(w) = 1$ holds.

(c8) Node $u$ has no “real” debits: apply (c7) and Observation 1.

(c9) The transfer $(v, w)$ exists in $G$: in step $s'$ the degree of $w$ drops from $d(w) = 1$ to $d(w) = 0$: since $d(u) = 1$, node $u$ is not connected to $w$ and hence $v$ is connected to $w$.

So $(v, w)$ is the only uncanceled credit to $X$. Which fake credit to $X$ should we add? Since by (c8) node $u$ has no debits, we credit $w$ in a fake transfer $(u, w)$ from $u$.

**Definition 2.** Let $X$ be a $\frac{1}{2}$-path with exactly one uncanceled credit $(v, w)$ to a path endpoint $w$. We call $(u, w)$ a fake transfer from the $M$-neighbor $u$ of $v$.

Now every $\frac{1}{2}$-path $X$ has no debits and either at least two uncanceled credits or exactly one uncanceled and exactly one fake credit: $X$ has at least two more credits than debits, as claimed in (5). To conclude our analysis, it remains to verify the total debit bounds given in (6) and (7) for singletons respectively augmenting paths. We conclude this section observing kinds of nodes without (fake) debits.

(c10) Node $v$ has no fake debits: fake transfers are added only out of $u$.

(c11) Nodes of the $M$-edge $\{x, x'\}$ picked at creation of an augmenting path $X$ have no debits at all: real debits to $x, x'$ are canceled, fake debits are added only to nodes selected by degree-1 steps.
4.3 Total Debit Bounds For Singletons

For a singleton $X$ we claimed in (6) total debits of at most
\[ d_X - c_X = d_X \leq 2(\Delta - 1) - 2 + 1. \]

If there are no fake transfers out of nodes of $X$, then Lemma 2 (d) gives an upper bound of at most $d_X \leq 2(\Delta - 1) - 2$ uncanceled debits and we are done. So assume that $X$ has nodes $u, v$ and fake debits to $u$. By (c8), node $u$ does not have real debits. By (c10), node $v$ does not have fake debits. Node $v$ has at most $\Delta - 1$ uncanceled debits, call them $(v, w_1), \ldots, (v, w_{\Delta - 1})$. We show in Lemma 7 b) that at most $\Delta - 2$ of the $\frac{1}{\Delta}$-paths of the $w_i$ need a fake credit, hence at most $\Delta - 2$ fake debits to $u$ are added, see Figure 12 for an illustration.

So $d_X \leq (\Delta - 1) + (\Delta - 2) = 2(\Delta - 1) - 1$ holds and we are done. We prepare

\[ \leq \Delta - 1 \]
\[ \leq \Delta - 2 \]

Fig. 12. Maximum debits to a singleton. Not all edges are drawn.

the bound of Lemma 7.

Lemma 6. Let $\{u, v\}$ be an $M$-edge and $v$ have uncanceled transfers $(v, w_1), \ldots, (v, w_n)$. Assume that fake transfers $(u, w_1), \ldots, (u, w_n)$ are added. Before $u, v$ are matched, the degrees of all $w_i$ drop to $d(w_i) = 1$ in the same step $s_i$.

Proof. Recall that by (c4) the following holds for the destination nodes $w_i$ of the uncanceled transfers $(v, w_i)$: the same step $s$ isolates all $w_i$ when all $d(w_i) = 1$. By (c6), for all $i$ the degree of $w_i$ drops to $d(w_i) = 1$ in a step $s_i$ creating an augmenting path picking an $M$-edge $e_i$. Recall that step $s_i$ selects a node of degree at least two.

Assume that there is a step when for $j \neq k$ we have $d(w_j) = 1$ and $d(w_k) \geq 2$. Until $w_j$ is isolated in step $s$, MinGreedy performs only degree-1 steps and hence step $s_k$ does not happen until after step $s$. Therefore $d(w_k) \geq 2$ when $w_j$ is isolated, a contradiction since all $w_i$ are isolated by the same step when all $d(w_i) = 1$. So all $d(w_i)$ are decreased to $d(w_i) = 1$ by the same step $s_i = s_j = \cdots = s_k$ creating an augmenting path by picking an $M$-edge $e_1 = \cdots = e_k$. \(\square\)
Lemma 7. Let \( \{u, v\} \) be an \( M \)-edge, assume that the fake transfers \((u, w_1), \ldots, (u, w_n)\) are added and the degrees of all \( w_i \) drop to \( d(w_i) = 1 \) at step \( s_i \).

a) For each \( w_i \) step \( s_j \) removes an edge \( \{w_i, x\} \in C \).

b) It holds that \( n \leq \Delta - 2 \).

c) If \( n = \Delta - 2 \), then step \( s_j \) removes exactly \( \Delta - 2 \) edges of \( C \).

Proof. We prove a). By (c6) step \( s_j \) creates an augmenting path \( X \), since at step \( s_j \) the degrees of the \( w_i \) drop to \( d(w_i) = 1 \). If some \( w_i \) is not a path endpoint of \( X \), then the statement is clear. So assume that \( w_i \) is a path endpoint of \( X \). Since \( s_j \) selects a node of degree at least two we have \( d(w_i) \geq 2 \) at step \( s_j \). But \( w \) is incident to at most one edge of \( X \) and hence to at least one \( C \)-edge.

We prove b) and c). W.l.o.g. step \( s_j \) selects \( u \) and \( v \) with \( v \). Step \( s_j \) selects \( u \) when \( 2 \leq d(u) \leq 3 \) since an augmenting path is created and the degrees of the \( w_i \) drop to \( d(w_i) = 1 \) from at most \( d(w_i) \leq 3 \).

Assume that \( X \) is incident only to its \( M \)- and \( M^* \)-edge. So the \( n \) distinct \( C \)-edges being removed by a) are incident to \( v \). Since \( v \) is also incident to its \( M \)- and \( M^* \)-edge, we get \( n \leq d(v) - 2 \leq \Delta - 2 \). Exactly \( n \) many \( C \)-edges, namely \( \{v_1, w_1\}, \ldots, \{v_n, w_n\} \), are removed, which holds in particular for \( n = \Delta - 2 \).

Assume that \( X \) is incident to at most its \( M \)- and \( M^* \)-edge. So the \( n \) distinct \( M \)- and \( M^* \)-edge are removed by a) are incident to \( v \). Since \( v \) is also incident to its \( M \)- and \( M^* \)-edge, we get \( n \leq d(u) - 2 \leq \Delta - 2 \). Exactly \( n \) many \( C \)-edges, namely \( \{v_1, w_1\}, \ldots, \{v_n, w_n\} \), are removed, which holds in particular for \( n = \Delta - 2 \).

Assume that \( X \) is incident to at most its \( M \)- and \( M^* \)-edge. Assume that \( n = \Delta - 2 \) holds. Using \( \Delta \geq 4 \) we get \( d(u) = 2 + \Delta - 2 > 3 = d(u) \), a contradiction. (Hence if \( n = \Delta - 2 \), then case \( d(u) = 2 \) applies, where exactly \( n \) many \( C \)-edges are removed.) \( \square \)

4.4 Total Debit Bounds For Augmenting Paths

For an augmenting path \( X \) with \( m_X \geq 2 \) we claim in [8] total debits of at most

\[
d_X - c_X \leq 2m_X(\Delta - 2) - 2(\Delta - 2) - 2 + 1 = 2(m_X - 1)(\Delta - 2) - 1.
\]

Consider an \( M \)-edge \( \{u, v\} \) of \( X \). Each of \( u, v \) has at most \( \Delta - 2 \) uncanceled debits. So if there are no fake debits, then both \( u, v \) have at most \( 2(\Delta - 2) \) uncanceled debits. If one of \( u, v \), say \( u \), has fake debits, then by [c8] and [c10] node \( u \) has no real debits and \( v \) has no fake debits. Since at most one fake debit to \( u \) is added for each uncanceled debit to \( v \), we obtain again that both \( u, v \) have at most \( 2(\Delta - 2) \) debits. So total debits to \( X \) are at most \( 2m_X(\Delta - 2) \). But since by [c11] nodes of the \( M \)-edge picked at creation of \( X \) have no debits at all, we get \( d_X \leq 2(m_X - 1)(\Delta - 2) \). If there is another debit missing, then we are done again. Thus the following result completes our proof.
Lemma 8. Let $X$ be an augmenting path with $m_X \geq 2$. If $X$ has its maximum number of $d_X = 2(m_X - 1)(\Delta - 2)$ debits, then $X$ has at least one credit.

Proof. If a debit to $X$ credits a path endpoint of $X$ (an “internal” transfer of $X$), then we are done. So assume that all debits to $X$ go to path endpoints of other components. Observe that when $X$ is created, at least one path endpoint of $X$, say $w$, is not isolated, since the $M^*$-edge of at most one path endpoint is removed. If $w$ is isolated in a step picking an $M$-edge of another component, then by Lemma 8 there is a credit to $w$ and we are done.

So assume that $w$ is isolated when a step $s$ picks an $M$-edge $\{u, v\}$ of $X$. Recall that by assumption there are no internal transfers, in particular neither of $u, v$ has an internal transfer to $w$. Hence the degree of $w$ is $d(w) = 1$ and $w$ is adjacent to exactly one of $u, v$ by an edge of $X$, which must be the $M^*$-edge of $w$. W.l.o.g. step $s$ selects node $u$. Since $d(w) = 1$, we get $d(u) = 1$ and the situation is as depicted in Figure 13: edge $\{u, v\}$ is the $M$-edge of $X$ nearest to $w$, edge $\{v, w\}$ is the $M^*$-edge of $w$, node $u$ is not connected to $w$ and the degrees of $u, w$ are $d(u) = d(w) = 1$.

Since node $u$ is selected with degree $d(u) = 1$, step $s$ does not create $X$. So each of $u, v$ has $\Delta - 2$ debits, since only the nodes of the $M$-edge of $X$ picked at creation have no debits. But node $u$ is selected with degree $d(u) = 1$, hence $u$ has no real debits. Therefore node $u$ has $\Delta - 2$ fake debits, call them $(u, w_1), \ldots, (u, w_{\Delta-2})$.

By Lemma 2, the degrees of all $w_i$ drop to $d(w_i) = 1$ in a step $s_i \leq s$ creating an augmenting path. So at step $s$ we have $d(w) = d(w_1) = \ldots = d(w_{\Delta-2}) = 1$. Since the degree of $w$ in $G$ is at least two by Proposition 2, a step $s' < s$ decreases the degree of $w$ to $d(w) = 1$ by removing an incident edge of $C$.

- Assume that $s'_i = s_i$. Since step $s'_i$ does not remove the $M^*$-edge $\{w, v\}$, which is removed later by step $s$, step $s'_i$ removes an edge of $C$ incident to $w$. Also, by Lemma 2, step $s'_i$ removes $\Delta - 2$ distinct edges of $C$ incident to all $w_i$. But since $w \neq w_i$ for all $i$, step $s'_i$ removes at least $\Delta - 1$ many edges of $C$. A contradiction to Lemma 2.

- Assume that $s'_i \neq s_i$. Since in step $s'_i$ the degree of $w$ drops to $d(w) = 1$ and step $s_i$ selects a node of degree at least 2, step $s'_i$ happens before step $s_i$. Since step $s_i$ decreases the degrees of all $w_i$ to $d(w_i) = 1$, step $s'_i$ is a degree-1 step. Consequently, step $s'_i$ does not create $X$. Also, step $s'_i$ does not pick
any other $M$-edge of $X$: step $s'_4$ does not remove $\{w, v\}$, which is removed by step $s$, nor does $s'_4$ remove an edge of $C$ incident to $w$, which would imply an internal transfer. Hence step $s'_4$ picks the $M$-edge of a component other than $X$ and does not create an augmenting path, since step $s'_4$ is a degree-1 step. So there is a credit to $w$ coming from the $M$-edge picked by step $s'_4$ and therefore from another component. \qed

5 Inapproximability Results

MinGreedy is ignorant to its input instance in the sense that a step of MinGreedy does not exploit knowledge gathered about the input in previous steps: e.g. the neighbors of the selected node $u$ are not remembered in order to “explore” the neighborhood of $u$ later. In a step of MinGreedy, an arbitrary node of minimum degree, who is located in an unknown place in the graph, is matched to an arbitrary neighbor.

A question arises naturally: are the worst case performance guarantees given above for MinGreedy optimal, i.e., is there a greedy matching algorithm which always computes larger matchings than proven for MinGreedy? In particular, a greedy matching algorithm in question may in each step utilize all previously gathered knowledge in very sophisticated node and edge selection routines.

Adaptive priority algorithms [BNR02] in the vertex model [DI04] define a large class of deterministic greedy matching algorithms, which we denote as $\mathcal{APV}$. $\mathcal{APV}$-algorithms do not have resource constraints and formalize the essential properties of greedy algorithms: to what extent can the input be unveiled in a single step, what are the possible irrevocable decisions for the constructed solution to be done after part of the input is revealed? In particular, an $\mathcal{APV}$-algorithm may gather and process much data about its input instances and deduce knowledge to be used in clever future steps. Therefore $\mathcal{APV}$-algorithms seem much stronger than MinGreedy.

Nevertheless, we construct graphs with degrees bounded by $\Delta$ for which a matching of size at most $\frac{\Delta - 1}{2}$ is optimal. So our $\frac{2}{3}$ lower bound for $\Delta = 3$ and our $\frac{\Delta - 1}{2}$ lower bound for $\Delta$-regular graphs are tight: the very simple MinGreedy algorithm has optimal worst case performance among $\mathcal{APV}$-algorithms. For graphs of degree at most $\Delta$ our $\frac{\Delta - 1}{2}$ lower bound shows that MinGreedy has good worst case performance.

For an $\mathcal{APV}$ algorithm $A$ the input graph is represented as a set of adjacency lists, e.g. $\{\langle u; v, w \rangle, \langle v; u, w, \ldots \rangle\}$ is the triangle on $u, v, w$ (where an arbitrarily ordered list of neighbors appears after a semicolon). Algorithm $A$ has no a priori knowledge about the input graph $G$ and starts with an empty matching.

In a step, algorithm $A$ selects a node by specifying a total priority order on all possible adjacency lists. Then $A$ receives the highest priority adjacency list of a still non-isolated node $u$, say $(u; v, w, \ldots)$. (A node is called isolated, if it, or each of its neighbors, is matched.) Lastly, algorithm $A$ selects a matching partner for $u$ from $\{v, w, \ldots\}$ and then changes to the next step.
Note that matched nodes are not removed from the adjacency lists of $G$. This increases the power of $A$: a neighbor of an already received node $v$ may be requested and hence $A$ is able to explore the neighborhood of $v$ and is not oblivious to the parts of $G$ being processed. Observe that MINGREEDY is an APV-algorithm: MINGREEDY remembers which nodes are already matched, requests non-isolated nodes with maximum numbers of matched neighbors (regardless of which neighbors there actually are) and ignores the neighborhood of all received nodes.

**Theorem 3.** Let $A$ be an APV-algorithm. There is an input graph of degree at most $\Delta$ for which $A$ computes a matching of size at most $\frac{\Delta - 1}{2\Delta - 3}$ times optimal.

**Proof.** We describe the construction of a hard input instance $G$ for algorithm $A$ as a game played between $A$ and an adversary $B$. As $A$ unveils $G$ only bit by bit, adversary $B$ may actually construct $G$ on the fly, thereby reacting to the various moves of $A$ such that $G$ has a much larger matching than the solution of $A$. Of course, all adjacency lists presented by $B$ during the whole game have to be consistent with the final graph $G$ constructed by $B$.

The game consists of the *regular game*, which lasts for $s = \Delta - 3$ steps, followed by the *endgame*, which has two steps.

During the regular game, adversary $B$ maintains the following invariant: each node $v$ that is not yet isolated has an adjacency list of one of the following types.

Type 1: $\langle v; v_1, \ldots, v_d \rangle$ where $v$ and $v_1, \ldots, v_d$ are unknown, i.e., they did not occur in a previously received adjacency list, and $3 \leq d \leq \Delta$.

Type 2: $\langle v; v_1, v_2 \rangle$ where $v$ and $v_1, v_2$ are unknown.

Type 3: $\langle v; v_1, v_2, v_3 \rangle$ where $v$ and $v_1, v_2$ are unknown and $v_3$ is known, i.e., node $v_3$ was received by $A$ in a previous adjacency list.

Observe, that all nodes in $G$ have degree at least two.

Consider the very first step of $A$. Since all nodes are still unknown, all nodes have adjacency lists of type 1 or 2. Hence the invariant holds. Consider step $i$ and assume that the invariant holds. Adversary $B$ presents the highest ranked adjacency list that is of type 1, 2 or 3. Call that adjacency list $a_i$.

**Fig. 14.** A connected component of a hard instance

**Fig. 15.** The core of a hard instance (Gray nodes are unknown, fat frontier nodes. The dashed edge is an example for $\{m_i = v, r_j = v_3\}$ in case 3.)
Case 1: $a_i = \langle v; v_1, \ldots, v_d \rangle$ is a type-1 adjacency list. Since all nodes in $a_i$ are unknown, we may w.l.o.g. assume that $A$ matches $v$ with $v_1$. Adversary $B$ constructs the connected component $C$ depicted in Figure 14 which consists only of nodes of types 1 and 2. All nodes of $C$ are isolated in the next step, hence the invariant is maintained. Observe that within $C$ the maximum matching $M^*$ scores two edges (the double edges $\{v, v_2\}, \{v_1, v_d\}$ in Figure 14) whereas the matching $M$ computed by $A$ scores just one edge (the crossed edge $\{v, v_1\}$).

Since in Case 1 algorithm $A$ requests only unknown nodes, adversary $B$ is able to trick $A$ into unveiling part of $G$ from which $A$ cannot gather knowledge about the rest of $G$. Can $A$ act smarter? Assume that $A$ has already received the adjacency lists of the middle nodes $m_1, \ldots, m_k$ of the triangles $\{l_j, m_j, r_j\}$ of known nodes connected by frontier nodes $r_j$ and unknown nodes $u_j$ to the still unknown center of $G$, see Figure 15. If $A$ requests an unknown node with two unknown neighbors, then $B$ easily tricks $A$ by constructing a new triangle $\{l_i, m_i, r_i\}$.

Case 2: $a_i = \langle v; v_1, v_2 \rangle$ is a type-2 adjacency list. Again, all nodes of $a_i$ are unknown and we may assume that $A$ matches $v$ with $v_1$. Adversary $B$ constructs a triangle $\{l_i, m_i, r_i\}$ with $l_i = v_2, m_i = v, r_i = v_1$ and inserts the edge $\{r_i, u_i\}$, with a new unknown node $u_i$, to connect the triangle to the unknown center. Observe that before nodes $m_i, r_i$ are matched, nodes $m_i, l_i$ are of type 2 and $r_i, u_i$ are of type 1. After matching $m_i, r_i$, nodes $l_i, m_i, r_i$ are isolated and $u_i$ turns into a type-3 node. Hence the invariant still holds. Again, $M^*$ scores two edges, namely $\{l_i, m_i\}, \{r_i, u_i\}$, and $M$ scores the edge $\{m_i, r_i\}$.

Now assume that $A$ tries to explore the neighborhood of known nodes. Observe that the only adjacency lists with a known node are of type 3 and have exactly one unknown node: since the known nodes $l_j, m_j, r_j$ are already isolated, an unknown node can only be explored in the neighborhood of frontier nodes. Again, adversary $B$ tricks $A$ with a new triangle $\{l_i, m_i, r_i\}$.

Case 3: $a_i = \langle v; v_1, v_2, v_3 \rangle$ is a type-3 adjacency list. Since $v_3$ is known, $v_3$ occurred in a previously presented adjacency list. Observe that in our construction so far, the only type-3 nodes are unknown neighbors of known frontier nodes. So $v$ is the neighbor of a frontier node $r_j = v_3$ with $j < i$.

Is algorithm successful in exploring the unknown neighbor $u_j$ of $r_j$, i.e., does $v = u_j$ hold? Not necessarily, since $B$ may on the fly construct further neighbors of $r_j$. Why? Since $r_j$ gets matched as soon as it becomes known, algorithm $A$ never gets to see the adjacency list of $r_j$ and consequently $A$ can never tell if it already knows all neighbors of $r_j$. (Adversary $B$ uses this trick here as well as in the end game.)

Since $v_3 = r_j$ is matched and $v_1, v_2$ are unknown, we may assume that $A$ matches $v$ with $v_1$. Adversary $B$ behaves exactly as in case 2 and constructs the triangle $\{l_i = v_2, m_i = v, r_i = v_1\}$ and inserts the edge $\{r_i, u_i\}$ where $u_i$ is a new unknown node. To complete the devious trick, adversary $B$ also inserts the edge $\{m_i = v, r_j = v_3\}$ (see e.g. the dashed edge in Figure 15). Before $m_i, r_i$ are matched, node $l_i$ is of type 2, nodes $r_i, u_i$ are of type 1 and $m_i = v$ is of type 3. After matching $m_i, r_i$, nodes $l_i, m_i, r_i$ are isolated and $u_i$ turns into a type-3
node. \( u_j \) is still of type 3. Hence the invariant still holds. As in case 2, \( M^* \) scores \( \{l, m_i\}, \{r, u_i\} \) and \( M \) scores \( \{m_i, r_i\} \).

This concludes the regular game. In the first step of the endgame adversary \( B \) makes algorithm \( A \) match \( a \) with \( b \). Hence in the next and last step algorithm \( A \) matches \( c \). So algorithm \( A \) scores two edges in the center, whereas three edges are optimal. As desired, we get

\[
|M| = s + 2 = \Delta - 1 \quad \text{and} \quad |M^*| = 2s + 3 = 2\Delta - 3.
\]

Observe that our invariant still holds in the first step of the endgame. Again, adversary \( B \) presents the highest ranked adjacency list of type 1, 2 or 3. Observe that \( a \) and \( c \) are the only type-1 nodes left, since the \( u_j \) have known neighbors and are of type 3 and all other nodes have degree two and are of type 2. The degree of \( a \) and \( c \) is \( \delta \leq 3 + s \), since both \( a, c \) have three center neighbors and each step of the regular game adds at most one neighbor to \( a \) respectively \( c \). Let \( a, c \) be the adjacency list received in step \( s + 1 = \Delta - 2 \).

**Case 4a:** \( a = (v; v_1, \ldots, v_3) \) is a type-1 adjacency list. Since all nodes of \( a \) are unknown we may assume that \( A \) matches \( v \) with \( v_1 \). Adversary \( B \) chooses \( v = a, v_1 = b \) and \( v_2, \ldots, v_3 \) as the remaining neighbors of \( a \).

**Case 4b:** \( a = (v; v_1, v_2) \) is a type-2 adjacency list. Since all nodes of \( a \) are unknown we may assume that \( A \) matches \( v \) with \( v_1 \). Adversary \( B \) sets \( v = b, v_1 = a \) and \( v_2 = d \).

**Case 4c:** \( a = (v; v_1, v_2, v_3) \) is a type-3 adjacency list. As in case 3, the known node \( v_3 \) is some matched frontier node \( r_j, j < \Delta - 2 \) and we may assume that \( A \) matches \( v \) with \( v_1, v_2 \) are unknown. As in case 3, adversary \( B \) does not present the adjacency list of the unknown node \( u_j \). Instead, \( B \) makes \( b \) a neighbor of \( r_j \) by inserting \( \{v_3 = r_j, b\} \)—now \( b \) has three neighbors—and sets \( v = b, v_1 = a \) and \( v_2 = d \).

Adversary \( B \) does not violate degree constraints. Nodes introduced in case 1 have degree at most \( d \leq \Delta \). All other degrees are at most three, but for \( a, c \) and frontier nodes \( r_j \). As discussed, nodes \( a, c \) have degree at most \( \delta = 3 + s \leq \Delta \). Frontier node have degree at most \( 3 + (s - 1) + 1 = \Delta \), since in each but the first step of the regular game and in step \( s + 1 \) at most one incident edge is added. \( \square \)

### 6 Conclusion

We have analyzed the worst case approximation ratio of the well-known MINGREEDY algorithm on graphs of bounded degree. Our performance guarantees of \( \frac{\Delta}{3} \) for graphs of degree at most three and of \( \frac{\Delta - 1}{2\Delta - 3} \) for \( \Delta \)-regular graphs are tight. In particular, MINGREEDY is optimal in the large class of \( APV \)-algorithms. We also proved a performance guarantee of \( \frac{\Delta - 1/2}{2\Delta - 2} \) for graphs of degree at most \( \Delta \), and we conjecture that also in this case MINGREEDY is optimal among \( APV \)-algorithms and achieves a worst case approximation ratio of at least \( \frac{\Delta - 1}{2\Delta - 3} \).

Our worst case performance guarantees are stronger than the best known worst case bounds on the expected approximation ratio for the well-known greedy matching algorithms GREEDY, MRG and SHUFFLE, if degrees are small.
**Open Questions.** Is MinGreedy optimal among APV-algorithms on graphs of degree at most $\Delta$?

What bounds for MinGreedy can be shown for more restricted graph classes, e.g. bipartite graphs?

Recall that the expected approximation ratio of the randomized MinGreedy algorithm is $\frac{1}{2} + o(1)$ w.h.p. on graphs of arbitrarily large degree. Does randomized MinGreedy have an expected approximation ratio strictly better than $\frac{\Delta - 1}{2\Delta - 3}$ if degrees are bounded by $\Delta$?

**Acknowledgements.** I thank Georg Schnitger for many helpful discussions.

**References**

ADFS95. Jonathan Aronson, Martin Dyer, Alan Frieze, and Stephen Suen. Randomized greedy matching. ii. *Random Structures & Algorithms*, 6(1):55–73, 1995.

BNR02. Allan Borodin, Morten N. Nielsen, and Charles Rackoff. (incremental) priority algorithms. In *Proc. 13th SODA*, pages 752–761, 2002.

CCWZ14. T.-H. Hubert Chan, Fei Chen, Xiaowei Wu, and Zhichao Zhao. Ranking on arbitrary graphs: Rematch via continuous lp with monotone and boundary condition constraints. In Chandra Chekuri, editor, *SODA*, pages 1112–1122. SIAM, 2014.

DF91. Martin E. Dyer and Alan M. Frieze. Randomized greedy matching. *Random Structures & Algorithms*, 2(1):29–46, 1991.

DI04. Sashka Davis and Russell Impagliazzo. Models of greedy algorithms for graph problems. In *Proc. 15th SODA*, pages 381–390, 2004.

FRS93. Alan Frieze, A. J. Radcliffe, and Stephen Suen. Analysis of a simple greedy matching algorithm on random cubic graphs. In *Proc. 4th SODA*, pages 341–351, 1993.

GT12. Gagan Goel and Pushkar Tripathi. Matching with our eyes closed. In *Proc. 53rd FOCS*, pages 718–727, 2012.

KS81. Richard M. Karp and Michael Sipser. Maximum matchings in sparse random graphs. In *Proc. 22nd FOCS*, pages 364–375, 1981.

KVV90. Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proc. 22nd STOC*, pages 352–358, 1990.

Mag97. Jakob Magun. Greedy matching algorithms, an experimental study. In *Proceedings of the 1st Workshop on Algorithm Engineering*, volume 6, pages 22–31, 1997.

MP97. Zefi Miller and Dan Pritikin. On randomized greedy matchings. *Random Struct. Algorithms*, 10(3):353–383, 1997.

MS04. Marcin Mucha and Piotr Sankowski. Maximum matchings via gaussian elimination. In *FOCS*, pages 248–255. IEEE Computer Society, 2004.

MV80. Silvio Micali and Vijay V. Vazirani. An $o(\sqrt{n}\log(n))$ algorithm for finding maximum matching in general graphs. In *FOCS*, pages 17–27. IEEE Computer Society, 1980.

Pol12. Matthias Poloczek. *Greedy algorithms for max sat and maximum matching: their power and limitations*. PhD thesis, Institut für Informatik, Goethe-Universität Frankfurt am Main, 2012. [http://d-nb.info/1036608425](http://d-nb.info/1036608425)
Tin84. G. Tinhofer. A probabilistic analysis of some greedy cardinality matching algorithms. *Annals of Operations Research*, 1(3):239–254, 1984.

Vaz12. Vijay V. Vazirani. An improved definition of blossoms and a simpler proof of the mv matching algorithm. *CoRR*, abs/1210.4594, 2012.