Mathematical Programming
https://doi.org/10.1007/s10107-025-02233-0

FULL LENGTH PAPER

Series B

A 4/3-approximation for the maximum leaf spanning arborescence problem in DAGs

Meike Neuwohner[1]

Received: 15 July 2024 / Accepted: 17 April 2025

Abstract

The Maximum Leaf Spanning Arborescence problem (MLSA) in directed acyclic graphs (dags) is defined as follows: Given a directed acyclic graph G and a vertex $r \in V(G)$ from which every other vertex is reachable, find a spanning arborescence rooted at r maximizing the number of leaves (vertices with out-degree zero). The MLSA in dags is known to be APX-hard as reported by Nadine Schwartges, Spoerhase, and Wolff (Approximation and Online Algorithms, Springer, Berlin Heidelberg, 2012) and the best known approximation guarantee of $\frac{7}{5}$ is due to Fernandes and Lintzmayer (J. Comput. Syst. Sci. 135: 158–174,2023): They prove that any α-approximation for the *hereditary* 3-*set packing problem*, a special case of weighted 3-set packing, yields a $\max\{\frac{4}{3}, \alpha\}$-approximation for the MLSA in dags, and provide a $\frac{7}{5}$-approximation for the hereditary 3-set packing problem. In this paper, we improve upon this result by providing a $\frac{4}{3}$-approximation for the hereditary 3-set packing problem, and, thus, the MLSA in dags. The algorithm that we study is a simple local search procedure considering swaps of size up to 10 and can be analyzed via a two-stage charging argument. We further provide a clear picture of the general connection between the MLSA in dags and set packing by rephrasing the MLSA in dags as a *hereditary set packing problem*. With a much simpler proof, we extend the reduction by Fernandes and Lintzmayer and show that an α-approximation for the *hereditary* k-*set packing problem* implies a $\max\{\frac{k+1}{k}, \alpha\}$-approximation for the MLSA dags. On the other hand, we provide lower bound examples proving that our approximation guarantee of $\frac{4}{3}$ is best possible for local search algorithms with constant improvement size.

Keywords Maximum Leaf Spanning Arborescence problem · Weighted set packing · Hereditary set packing · Local search

Mathematics Subject Classification 68W25 · 05C65 · 05C70 · 90C27

This is the full version of the paper [19], which appeared as an extended abstract in the proceedings of IPCO 2024.

✉ Meike Neuwohner
work@meike-neuwohner.de

1 Department of Mathematics, London School of Economics and Political Science, London, UK

Published online: 24 May 2025

1 Introduction

Given a (simple) directed graph $G = (V, E)$ and a root vertex $r \in V$, we call a subgraph T of G a *spanning r-arborescence in G* if it satisfies the following conditions:

(i) T is a *spanning subgraph* of G, that is, $V(T) = V$.
(ii) r does not have any entering arc in T and each $v \in V \setminus \{r\}$ has exactly one entering arc in T.
(iii) Each vertex in V is reachable from r via a directed path in T.

We call a vertex v a *leaf* of T if v does not have any leaving arc in T.

The *Maximum Leaf Spanning Arborescence problem (MLSA)* is defined as follows:

Definition 1 *(Maximum Leaf Spanning Arborescence problem)*

Input: A directed graph G, $r \in V(G)$ such that every vertex of G is reachable from r via a directed path.
Task: Find a spanning r-arborescence in G with the maximum number of leaves possible.

It plays an important role in the context of broadcasting: Given a network consisting of a set of nodes containing one distinguished source and a set of available arcs, a message needs to be transferred from the source to all other nodes along a subset of the arcs, which forms (the arc set of) an arborescence rooted at the source. As internal nodes do not only need to be able to receive, but also to re-distribute messages, they are more expensive. Hence, it is desirable to have as few of them as possible, or equivalently, to maximize the number of leaves.

The special case of the MLSA where every arc may be used in both directions is called the *Maximum Leaves Spanning Tree problem (MLST)*. In this setting, the complementary task of minimizing the number of non-leaves is equivalent to the *Minimum Connected Dominating Set problem (MCDS)*. Both the MLST and the MCDS are NP-hard, even if the input graph is 4-regular or planar with maximum degree at most 4 (see [13], problem ND2). The MLST has been shown to be APX-hard [12],[1] even when restricted to cubic graphs [3]. The state-of-the-art for the MLST is an approximation guarantee of 2 [22].

While an optimum solution to the MLST gives rise to an optimum solution to the MCDS and vice versa, the MCDS turns out to be much harder to approximate: Ruan et al. [20] have obtained an $(\ln \Delta + 2)$-approximation, where Δ denotes the maximum degree in the graph. A reduction from Set Cover (with bounded set sizes) further shows that unless P = NP, the MCDS is hard to approximate within a factor of $\ln \Delta - \mathcal{O}(\ln \ln \Delta)$ [14, 24]. An analogous reduction further yields the same hardness result for the problem of computing a spanning arborescence with the minimum number of non-leaves in a rooted acyclic digraph of maximum out-degree Δ.

In this paper, we study polynomial-time approximation algorithms for (a special case of) the MLSA. For general digraphs, the best that is known is a $\min\{\sqrt{\mathrm{OPT}}, 92\}$-approximation [5, 6]. Moreover, there is a line of research focusing on FPT-algorithms for the MLSA [1, 2, 5].

[1] Note that MaxSNP-hardness implies APX-hardness, see [16].

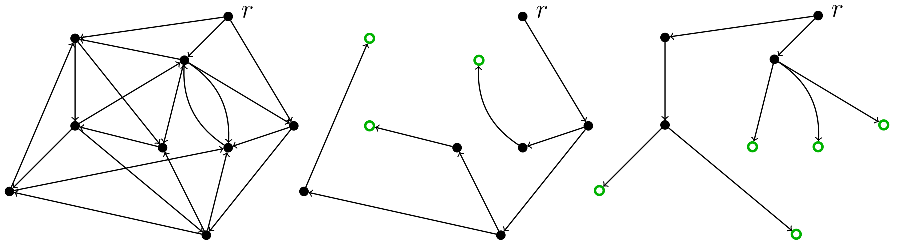


Fig. 1 Illustration of the Maximum Leaf Spanning Arborescence problem. The leftmost picture shows a simple directed graph $G = (V, E)$, together with a vertex $r \in V$ from which every other vertex is reachable. The middle picture illustrates a spanning r-arborescence in G with 3 leaves (indicated by green, empty circles). The rightmost picture shows a spanning r-arborescence in G with 5 leaves

The special case where the graph G is assumed to be a dag (directed acyclic graph) has been proven to be APX-hard by Schwartges, Spoerhase and Wolff [21]. They further provided a 2-approximation, which was then improved to $\frac{3}{2}$ by Fernandes and Lintzmayer [9]. Recently, the latter authors managed to enhance their approach to obtain a $\frac{7}{5}$-approximation [10], which has been unchallenged so far. In this paper, following the approach by Fernandes and Lintzmayer, we improve on these results and obtain a $\frac{4}{3}$-approximation for the MLSA in dags.

Fernandes and Lintzmayer [10] tackle the MLSA in dags by reducing it, up to an approximation guarantee of $\frac{4}{3}$, to a special case of the weighted 3-set packing problem, which we call the *hereditary* 3-*set packing problem*. Fernandes and Lintzmayer [10] prove it to be NP-hard via a reduction from 3-Dimensional Matching [15].

Definition 2 *(weighted k-set packing problem)*

Input: A family $\mathcal{S}$ of non-empty sets, each of cardinality at most k, $w : \mathcal{S} \to \mathbb{R}_{\geq 0}$
Task: Compute a disjoint sub-collection $A \subseteq \mathcal{S}$ maximizing the total weight $w(A) := \sum_{s \in A} w(s)$.

We call a set family $\mathcal{S}$ *hereditary* if for every $s \in \mathcal{S}$, $\mathcal{S}$ contains all non-empty subsets of s.

Definition 3 *(hereditary* 3-*set packing problem)* An instance of the *hereditary* 3-*set packing problem* is an instance $(\mathcal{S}, w)$ of the weighted 3-set packing problem, where $\mathcal{S}$ is a hereditary family and $w(s) = |s| - 1$ for all $s \in \mathcal{S}$.

As the weights can be deduced from the set sizes, we will omit them in the following and simply denote an instance of the hereditary 3-set packing problem by $\mathcal{S}$ (instead of $(\mathcal{S}, w)$). We remark that there are two natural ways to encode an instance $\mathcal{S}$ of the hereditary 3-set packing problem, either by providing a list of all sets in $\mathcal{S}$, or by only listing the collection $\mathcal{S}_{max}$ of inclusion-wise maximal ones. As $\sum_{s \in \mathcal{S}} |s| \leq 4 \cdot \sum_{s \in \mathcal{S}_{max}} |s|$, both choices result in the same notion of polynomial running time. However, when considering larger set sizes in Section 2, polynomial running time will always mean polynomial with respect to $\sum_{s \in \mathcal{S}_{max}} |s|$.

Theorem 4 ([10]) *Let $\alpha \geq 1$ and assume that there is a polynomial-time α-approximation algorithm for the hereditary 3-set packing problem. Then there exists a polynomial-time* $\max\{\alpha, \frac{4}{3}\}$*-approximation for the MLSA in dags.*

For $k \leq 2$, the weighted k-set packing problem can be solved in polynomial time via a reduction to the Maximum Weight Matching problem [7]. In contrast, for $k \geq 3$, even the special case where $w \equiv 1$, the *unweighted k-set packing problem*, is NP-hard because it generalizes the 3-Dimensional Matching problem [15]. The technique that has proven most successful in designing approximation algorithms for both the weighted and the unweighted k-set packing problem is *local search*. The general idea is to iteratively increase the weight of a feasible solution by adding and removing a bounded number of sets in such a way that the solution remains feasible. More precisely, when adding a collection X of pairwise disjoint sets to a feasible solution A, we have to delete all sets from A that intersect sets in X. We call these sets the *neighborhood of X in A* and denote it by $N(X, A) := \{a \in A : \exists x \in X : a \cap x \neq \emptyset\}$. A disjoint set collection X constitutes a *local improvement of A* if $w(X) > w(N(X, A))$, i.e., if adding the sets in X and deleting the sets in the neighborhood results in a solution of larger weight.

The state-of-the-art is a $\min\{\frac{k+1-\tau_k}{2}, 0.4986 \cdot (k+1) + 0.0208\}$-approximation for the weighted k-set packing problem, where $\tau_k \geq 0.428$ for $k \geq 3$ and $\lim_{k\to\infty} \tau_k = \frac{2}{3}$ [18, 23]. Note that the guarantee of 1.786 for $k = 3$ is worse than the guarantee of $\frac{7}{5}$ that Fernandes and Lintzmayer achieve for the hereditary 3-set packing problem.

In order to obtain the approximation guarantee of $\frac{7}{5}$, Fernandes and Lintzmayer perform local search with respect to a modified weight function. In addition to certain improvements of constant size, they incorporate another, more involved class of local improvements that are related to alternating paths in a certain auxiliary graph. This leads to a rather complex analysis because in addition to charging arguments similar to ours, more intricate considerations regarding the structure of the auxiliary graph are required.

In this paper, we study a local search algorithm that considers local improvements consisting of up to 10 sets with respect to an objective that first maximizes the weight of the current solution, and second the number of sets of weight 2 that are contained in it. The intuition behind this is that we would like our solution to cover less set elements in total, (potentially) resulting in fewer intersections with sets from an optimum solution and making it "easier" to find local improvements. While one set of weight 2 only contains 3 set elements, two disjoint sets of weight 1, together, contain 4 set elements, making them less attractive to pick.

We show that the above-mentioned algorithm yields a polynomial-time $\frac{4}{3}$-approximation for the hereditary 3-set packing problem. In particular, this results in a polynomial-time $\frac{4}{3}$-approximation for the MLSA in dags. In doing so, we manage to tap the full potential of Theorem 4. Moreover, this work serves as a starting point in identifying, understanding, and exploiting structural properties of set packing instances that arise naturally from other combinatorial problems. Studying these instance classes may ultimately turn reductions to set packing instances into a more powerful tool in the design of approximation algorithms.

The remainder of this paper is organized as follows: In Section 2, we introduce the *hereditary k-set packing problem* and the *hereditary set packing problem*, which constitute natural extensions of the hereditary 3-set packing problem to sets of size at most k and arbitrary set sizes, respectively. We provide a simple, approximation-preserving reduction from the MLSA in dags to the hereditary set packing problem. Then, we show that for every $k \geq 1$, a polynomial-time α-approximation for the hereditary k-set packing problem implies a polynomial-time $\max\{\alpha, \frac{k+1}{k}\}$-approximation for the hereditary set packing problem, and thus, also for the MLSA in dags. These results yield a clear picture of the connections between the MLSA in dags, the hereditary set packing problem and the bounded size variants. Moreover, we obtain a significantly simplified proof of Theorem 4.

The lower bound of $\frac{k+1}{k}$ strictly decreases as k grows, and converges to 1 as k tends to infinity. Hence, it appears natural to ask whether a better approximation ratio than $\frac{4}{3}$ can be achieved by reducing to the hereditary k-set packing problem with $k \geq 4$ instead. In Sect. 3, we show, however, that an algorithm for the hereditary k-set packing problem that only considers local improvements of constant size cannot yield a better approximation ratio than $2 - \frac{2}{k}$. Note that $k \mapsto \max\{\frac{k+1}{k}, 2 - \frac{2}{k}\}$ has a unique minimum at $k = 3$, where it attains a value of $\frac{4}{3}$. As such, the approximation guarantee of $\frac{4}{3}$ is optimal for the approach we consider.

Finally, in Sect. 4, we present a simple local search based $\frac{4}{3}$-approximation for the hereditary 3-set packing problem.

2 A set packing problem in disguise

In this section, we point out that the MLSA in dags is, at its core, a set packing problem. In Sect. 2.1, we formally introduce the hereditary set packing problem and provide a simple approximation-preserving reduction from the MLSA in dags to it. In Sect. 2.2, we then show that up to an approximation guarantee of $\frac{k+1}{k}$, we can reduce further to a setting where all sets in our instance contain at most k elements ($k \geq 1$). The special case $k = 3$ yields a simple and self-contained proof of Theorem 4.

2.1 Reducing the MLSA in DAGs to hereditary set packing

The *hereditary set packing problem* is defined as follows:

Definition 5 *(hereditary set packing problem)*

Input: a hereditary family $\mathcal{S}$ of non-empty sets
Task: Compute a disjoint sub-collection $A \subseteq \mathcal{S}$ maximizing $w(A) = \sum_{s \in A} w(s)$, where $w(s) := |s| - 1$.

In order to avoid an unnecessary, potentially exponential overhead in the encoding length, we will assume in the following that a hereditary set family $\mathcal{S}$ is implicitly given by only storing the inclusion-wise maximal sets in $\mathcal{S}$ explicitly.

Our main result for this section is given by the following theorem:

Theorem 6 *Let $\alpha \geq 1$. If there is a polynomial-time α-approximation algorithm for the hereditary set packing problem, then there is a polynomial-time α-approximation algorithm for the MLSA in dags.*

In order to phrase our reduction from the MLSA in dags to the hereditary set packing problem, we require the following definition:

Definition 7 Let $G = (V, E)$ be a directed graph. For $v \in V$, we define $\Gamma_G^+(v)$ and $\Gamma_G^-(v)$ to be the set of out- and in-neighbors of v, respectively, that is,

$$\Gamma_G^+(v) := \{w \in V : (v, w) \in E\} \text{ and } \Gamma_G^-(v) := \{w \in V : (w, v) \in E\}.$$

If G is clear from the context, we may omit the subscript G and just write $\Gamma^+(v)$ and $\Gamma^-(v)$, respectively.

The following proposition tells us that finding a spanning r-arborescence in G can be interpreted as a set partitioning problem:

Proposition 8 *Let $G = (V, E)$ be a dag and let $r \in V$ be a vertex from which every other vertex is reachable. Let further T be a spanning subgraph of G. The following are equivalent:*

(a) *T is a spanning r-arborescence in G.*
(b) *$\Gamma_T^-(r) = \emptyset$ and $|\Gamma_T^-(v)| = 1$ for every $v \in V \setminus \{r\}$.*
(c) *The sets $(\Gamma_T^+(v))_{v \in V}$ form a partition of $V \setminus \{r\}$.*

Proof Clearly, (b) and (c) are equivalent. Moreover, by definition of a spanning r-arborescence, (a) implies (b). Hence, we are left with showing that any spanning subgraph T of G that complies with (b) constitutes a spanning r-arborescence in G. To this end, it remains to check that every vertex is reachable from r via a directed path in T. But this follows from the fact that every vertex other than r has an entering arc in T: As G does not contain any directed cycle, we can simply follow the entering arcs backwards until we reach r. □

Moreover, it is easy to see that the number of leaves of a spanning r-arborescence T can be expressed in terms of the sizes of the out-neighborhoods in T.

Proposition 9 ([6], Proof of Lemma 2.1) *Let T be an arborescence. Then the number of leaves of T equals $1 + \sum_{v \in V(T): \Gamma_T^+(v) \neq \emptyset} (|\Gamma_T^+(v)| - 1)$.* □

By Proposition 8 and Proposition 9, finding a spanning r-arborescence with the maximum number of leaves is equivalent to partitioning $V \setminus \{r\}$ into a collection $\mathcal{S}$ of subsets of the out-neighborhoods of the vertices in V, maximizing the total weight $\sum_{s \in \mathcal{S}} (|s| - 1)$. Given that adding additional elements to the sets cannot decrease the objective value, we may actually relax the condition that the sets in $\mathcal{S}$ *partition* $V \setminus \{r\}$ to the weaker requirement that they are pairwise disjoint. This motivates the following definition:

Definition 10 Let $(G = (V, E), r)$ be an instance of the MLSA in dags. We define the *hereditary set family associated with* G to be

$$\mathcal{S}_G := \{U \subseteq V : \exists v \in V : \emptyset \neq U \subseteq \Gamma_G^+(v)\}.$$

Note that we can compute the inclusion-wise maximal sets in $\mathcal{S}_G$ in polynomial time $\mathcal{O}(|V|^3)$ by determining the inclusion-wise maximal ones among the sets $\{\Gamma_G^+(v) : v \in V\}$.

In the following, we formally present the reduction from the MLSA in dags to the hereditary set packing problem. Proposition 11 shows that a spanning r-arborescence with ℓ leaves can be converted into a solution to $\mathcal{S}_G$ of objective value $\ell - 1$. Conversely, Lemma 12 tells us that given a solution to $\mathcal{S}_G$ of objective value t, we can, in polynomial-time, compute a spanning r-arborescence in G with at least $t + 1$ leaves.

Proposition 11 *Let $(G = (V, E), r)$ be an instance of the MLSA in dags and let T be a spanning r-arborescence in G with ℓ leaves.*

Define $A_T := \{\Gamma_T^+(v) : v \in V, \Gamma_T^+(v) \neq \emptyset\}$. Then A_T is a feasible solution to $\mathcal{S}_G$ with objective value $\sum_{s \in A_T}(|s| - 1) = \ell - 1$.

Proof As in an arborescence, each vertex has at most 1 entering arc, the sets in A_T are pairwise disjoint. By Proposition 9, we have

$$\sum_{s \in A_T}(|s| - 1) = \sum_{v \in V : \Gamma_T^+(v) \neq \emptyset}(|\Gamma_T^+(v)| - 1) = \ell - 1.$$

□

Lemma 12 *Let $(G = (V, E), r)$ be an instance of the MLSA in dags and let A be a feasible solution to $\mathcal{S}_G$. Then we can, in polynomial time, construct a spanning r-arborescence in G with at least $1 + \sum_{s \in A}(|s| - 1)$ many leaves.*

Proof For $s \in A$, pick v_s such that $s \subseteq \Gamma_G^+(v_s)$. For $v \in V \setminus (\{r\} \cup \bigcup_{s \in A} s)$, pick an arbitrary entering arc e_v. Note that such an arc exists since every vertex is reachable from r via a directed path in G.

Define a spanning subgraph T of G via $V(T) := V$ and

$$E(T) := \{(v_s, w) : w \in s \in A\} \cup \left\{e_v : v \in V \setminus \left(\{r\} \cup \bigcup_{s \in A} s\right)\right\}.$$

By definition of $\mathcal{S}_G$, T is a subgraph of G. As the sets in A are pairwise disjoint, we have $|\Gamma_T^-(v)| = 1$ for every $v \in V \setminus \{r\}$. Finally, as G is acyclic and every vertex is reachable from r, r does not have any in-neighbor in G. In particular, $\Gamma_T^-(r) = \emptyset$. By Proposition 8, T is a spanning r-arborescence in G.

Let T' be the spanning subgraph of T with arc set $E(T') := \{(v_s, w) : w \in s \in A\}$. By Proposition 9, the number of leaves of T is

$$\begin{aligned}
&1 + \sum_{v \in V : \Gamma^+_T(v) \neq \emptyset} (|\Gamma^+_T(v)| - 1) = 1 + \sum_{v \in V} \max\{0, |\Gamma^+_T(v)| - 1\} \\
&\geq 1 + \sum_{v \in V} \max\{0, |\Gamma^+_{T'}(v)| - 1\} = 1 + \sum_{v \in V : \Gamma^+_{T'}(v) \neq \emptyset} (|\Gamma^+_{T'}(v)| - 1) \\
&= 1 + \sum_{v \in V : \Gamma^+_{T'}(v) \neq \emptyset} |\Gamma^+_{T'}(v)| - |\{v \in V : \Gamma^+_{T'}(v) \neq \emptyset\}| \\
&= 1 + \sum_{s \in A} |s| - |\{v_s : s \in A\}| \geq 1 + \sum_{s \in A} |s| - |A| = 1 + \sum_{s \in A} (|s| - 1).
\end{aligned}$$

□

Now, we are ready to prove Theorem 6.

Proof of Theorem 6 Assuming a polynomial-time α-approximation algorithm for the hereditary set packing problem, we obtain a polynomial-time α-approximation for the MLSA in dags as follows:

For a given instance (G, r), we first, in polynomial time, compute the representation of $\mathcal{S}_G$ by its inclusion-wise maximal sets. Next, we apply the α-approximation algorithm for the hereditary set packing problem to obtain an α-approximate solution A to $\mathcal{S}_G$. Finally, we employ Lemma 12 to construct a spanning r-arborescence T in G with at least $1 + \sum_{s \in A}(|s| - 1)$ many leaves.

In order to show that T is an α-approximate solution to the MLSA, denote the optimum value for (G, r) by OPT. Note that OPT ≥ 1. By Proposition 11, there exists a feasible solution to $\mathcal{S}_G$ of objective value OPT $- 1$. As a consequence, we have

$$\sum_{s \in A} (|s| - 1) \geq \alpha^{-1} \cdot (\mathrm{OPT} - 1).$$

This yields

$$1 + \sum_{s \in A} (|s| - 1) \geq \alpha^{-1} + \sum_{s \in A} (|s| - 1) \geq \alpha^{-1} + \alpha^{-1} \cdot (\mathrm{OPT} - 1) = \alpha^{-1} \cdot \mathrm{OPT}.$$

□

To conclude this section, we remark that not every instance of the hereditary set packing problem corresponds to an instance of the MLSA in dags: To this end, note that for an instance $(G = (V, E), r)$ of the MLSA in dags, we have $\bigcup_{s \in \mathcal{S}_G} s = V \setminus \{r\}$ since every vertex other than r has an entering arc. Moreover, the number of inclusion-wise maximal sets in $\mathcal{S}_G$ is bounded by $|V| - 1 = |V \setminus \{r\}| = |\bigcup_{s \in \mathcal{S}_G} s|$ (as G is a dag, at least one vertex does not have any leaving arcs). Now, consider the instance $\mathcal{S} = \binom{\{a,b,c,d\}}{1} \cup \binom{\{a,b,c,d\}}{2}$ consisting of all 1- and 2-element subsets of the 4-element ground set $\{a, b, c, d\}$. This instance contains $\binom{4}{2} = 6 > 4$ inclusion-wise maximal sets. As such, it cannot correspond to an instance of the MLSA in dags.

2.2 Reduction to bounded set sizes

In this section, we show that for every $k \geq 1$, up to an approximation guarantee of $\frac{k+1}{k}$, we can reduce the hereditary set packing problem to the special case where all set sizes are bounded by k. The precise statement is given by Theorem 14.

Definition 13 *(hereditary k-set packing problem)* The hereditary k-set packing problem is the restriction of the hereditary set packing problem to instances with sets of size at most k.

Note that this definition coincides with Definition 3 for $k = 3$.

Theorem 14 *Let $k \geq 1$. If there is a polynomial-time α-approximation algorithm for the hereditary k-set packing problem, then there is a polynomial-time* $\max\{\alpha, \frac{k+1}{k}\}$*-approximation algorithm for the hereditary set packing problem.*

Note that Theorem 4 follows by combining Theorem 6 and Theorem 14 for $k = 3$.

Proof of Theorem 14 Assuming a polynomial-time α-approximation algorithm for the hereditary k-set packing problem, we obtain a polynomial-time $\max\{\alpha, \frac{k+1}{k}\}$-approximation algorithm for the hereditary set packing problem as follows.

Given an instance $\mathcal{S}$ of the hereditary set packing problem, let

$$\mathcal{S}_{\geq k+1} := \{s \in \mathcal{S} : |s| \geq k + 1\}.$$

As a first step, we compute a maximal solution $M \subseteq \mathcal{S}_{\geq k+1}$: To this end, we initialize $M = \emptyset$. We then traverse the inclusion-wise maximal sets in $\mathcal{S}$ in an arbitrary order. For each maximal set s, we check whether $|s \setminus \bigcup_{s' \in M} s'| \geq k + 1$, and if yes, we add $s \setminus \bigcup_{s' \in M} s'$ to M.

We define $U := \bigcup_{s \in M} s$. Let $\mathcal{S}' := \{s \setminus U : s \in \mathcal{S}, s \setminus U \neq \emptyset\}$. By maximality of M, $\mathcal{S}'$ is an instance of the hereditary k-set packing problem. Moreover, the inclusion-wise maximal sets in $\mathcal{S}'$ are the inclusion-wise maximal ones among the sets $s \setminus U$, where $s \in \mathcal{S}$ is inclusion-wise maximal, and can, hence, be computed in polynomial time.

We apply the α-approximation algorithm for the hereditary k-set packing problem to $\mathcal{S}'$ and obtain a solution A'.

Finally, we output $A := M \cup A'$.

By construction, the sets in A are pairwise disjoint. Hence, it remains to prove that A is a $\max\{\alpha, \frac{k+1}{k}\}$-approximate solution. To this end, let B be an optimum solution for $\mathcal{S}$ and define $B' := \{b \setminus U : b \in B, b \setminus U \neq \emptyset\}$. Then B' is a feasible solution to $\mathcal{S}'$, which yields

$$\sum_{b \in B} |b \setminus U| - |B| \leq \sum_{b \in B} |b \setminus U| - |B'| = \sum_{b \in B'} (|b| - 1) \leq \alpha \cdot \sum_{a \in A'} (|a| - 1). \quad (1)$$

As the sets in M are pairwise disjoint and of cardinality at least $k+1$, we obtain $\sum_{m \in M} |m| = |U|$ and $|M| \leq \frac{1}{k+1} \cdot |U|$. Using that the sets in B are pairwise disjoint as well, we have

$$\sum_{b \in B} |b \cap U| \leq |U| = \frac{k+1}{k} \cdot \left(|U| - \frac{1}{k+1} \cdot |U|\right) \leq \frac{k+1}{k} \cdot \sum_{m \in M} (|m| - 1). \quad (2)$$

Adding (1) and (2) results in

$$\begin{aligned}\sum_{b \in B} (|b| - 1) &= \sum_{b \in B} |b \setminus U| - |B| + \sum_{b \in B} |b \cap U| \\ &\leq \alpha \cdot \sum_{a \in A'} (|a| - 1) + \frac{k+1}{k} \cdot \sum_{m \in M} (|m| - 1) \\ &\leq \max\left\{\alpha, \frac{k+1}{k}\right\} \cdot \sum_{a \in A} (|a| - 1),\end{aligned}$$

proving the desired approximation guarantee. □

3 Lower bound

In this section, we show that we cannot obtain a better approximation guarantee than $2 - \frac{2}{k}$ for the hereditary k-set packing problem via a local search algorithm that only considers local improvements of constant size. To this end, we first recap the definition of neighborhood from the introduction.

Definition 15 *(neighborhood)* Let X and Y be two set families. We define the *neighborhood* of X in Y to be

$$N(X, Y) := \{y \in Y : \exists x \in X : x \cap y \neq \emptyset\}.$$

Moreover, for a single set x, we write $N(x, Y) := N(\{x\}, Y)$.

The main result for this section is given by the following theorem.

Theorem 16 *Let $k \geq 3$ and $n, t \geq 1$. There exist*

- *an instance $\mathcal{S}$ of the hereditary k-set packing problem with $|\mathcal{S}| \geq n$ and*
- *feasible solutions A and B*

with the following properties:

- *For every $X \subseteq \mathcal{S} \setminus A$ with $|X| \leq t$ and such that the sets in X are pairwise disjoint, we have $w(X) < w(N(X, A))$. In particular, A is locally optimal with respect to local improvements of size at most t.*
- *$w(B) = \left(2 - \frac{2}{k}\right) \cdot w(A)$.*

Theorem 16 implies that an algorithm for the hereditary k-set packing that only considers local improvements of size at most t cannot achieve a better approximation ratio than $2 - \frac{2}{k}$. Indeed, when applied to the instance from Theorem 16, such an algorithm might find the (locally optimal) solution A by starting with the empty solution, and then, for $|A|$ iterations, applying local improvements of size 1 that add another set from A (and do not remove any set). However, the weight of A is by a factor of $2 - \frac{2}{k}$ smaller than the weight of B, so it is by a factor of at least $2 - \frac{2}{k}$ smaller than the weight of an optimum solution.

For the proof of Theorem 16, we first establish the following proposition, which is a direct consequence of a result by Erdős and Sachs [8]. It is also (implicitly) proven in [17].

Proposition 17 *Let $k \geq 3$ and $n, t \geq 1$. There is a simple $(2, k)$-regular bipartite graph G with $|V(G)| \geq n$ and $\mathrm{girth}(G) \geq k \cdot t + 1$, where $\mathrm{girth}(G)$ denotes the girth of G, i.e., the minimum length of a cycle in G.*

Proof Let $N := \max\{n, (k-1)^{k \cdot t}\}$. By [8], there exists a k-regular graph H on $|V(H)| \geq N$ vertices such that

$$\mathrm{girth}(H) \geq \frac{\log(|V(H)|)}{\log(k-1)} - 1 \geq \frac{\log(N)}{\log(k-1)} - 1 \geq k \cdot t - 1.$$

Let G be the bipartite vertex-edge-incidence graph of H, that is,

$$V(G) = V(H) \cup E(H) \text{ and } E(G) = \{\{v, e\} : v \in e \in E(H)\}.$$

Then G is a simple $(2, k)$-regular bipartite graph with $|V(G)| \geq |V(H)| \geq n$. As for every cycle $v_1, e_1, \ldots, v_k, e_k$ in G (where $v_1, \ldots, v_k \in V(H)$ and $e_1, \ldots, e_k \in E(H)$), $v_1, \ldots, v_k$ is a cycle in H, we have

$$\mathrm{girth}(G) \geq 2 \cdot \mathrm{girth}(H) \geq 2 \cdot k \cdot t - 2 \geq k \cdot t + 1,$$

where we used $k \geq 3$ and $t \geq 1$ for the last inequality. □

Proof of Theorem 16 Let $G = (V, E)$ be a simple $(2, k)$-regular bipartite graph with $|V| \geq n$ and $\mathrm{girth}(G) \geq k \cdot t + 1$. Let V_A and V_B be the two bipartitions of G, where every vertex in A has degree 2, and every vertex in B has degree k.

Let $\delta(v)$ denote the set of incident edges of vertex v in G and let $\mathcal{S} := \{s \subseteq E : \exists v \in V : \emptyset \neq s \subseteq \delta(v)\}$ consist of the non-empty subsets of the sets of incident edges of vertices in G. As every vertex in G has degree at most k, $\mathcal{S}$ is an instance of the hereditary k-set packing problem.

Define $A := \{\delta(v) : v \in V_A\}$ and $B := \{\delta(v) : v \in V_B\}$. As V_A and V_B are independent sets in G, A and B both consist of pairwise disjoint sets. As every vertex in V_A has degree 2 and every vertex in V_B has degree k, we have

$$w(A) = \sum_{v \in V_A} (|\delta(v)| - 1) = \frac{1}{2} \sum_{v \in V_A} |\delta(v)| = \frac{1}{2} \cdot |E|, \text{ and}$$

$$w(B) = \sum_{v \in V_B} (|\delta(v)| - 1) = \frac{k-1}{k} \sum_{v \in V_B} |\delta(v)| = \frac{k-1}{k} \cdot |E|.$$

This yields $w(B) = \frac{2\cdot(k-1)}{k} \cdot w(A) = \left(2 - \frac{2}{k}\right) \cdot w(A)$.

It remains to show that A is locally optimal. To this end, let $X \subseteq \mathcal{S} \setminus A$ such that the sets in X are pairwise disjoint and $|X| \leq t$. We need to prove that $w(X) < w(N(X, A))$.

First of all, we may assume that X does not contain any set $s \in \mathcal{S}$ with $|s| = 1$ since $w(s) = 0$ for such a set. In particular, as $X \subseteq \mathcal{S} \setminus A$ and $A = \{\delta(v) : v \in V_A\}$ consists of sets of size 2, we can infer that there is no $x \in X$ such that $x \subseteq \delta(v)$ for some $v \in V_A$. Consequently, for each $x \in X$, there is a (unique) $v_x \in V_B$ such that $x \subseteq \delta(v_x)$.

Define $E_X := \bigcup_{x \in X} x$ to be the collection of edges contained in the sets $x \in X$ and denote by $V_X := \bigcup_{e \in E_X} e$ the set of endpoints of these edges. Then

$$V_X \cap V_B = \{v_x : x \in X\} \text{ and } N(X, A) = \{\delta(v) : v \in V_X \cap V_A\}. \tag{3}$$

Using that all sets in A have a size of 2 and a weight of 1, we can infer that

$$w(N(X, A)) = |N(X, A)| = |V_X \cap V_A|. \tag{4}$$

As $|X| \leq t$, we know that $|E_X| \leq k \cdot |X| \leq k \cdot t$ and since the girth of G is at least $k \cdot t + 1$, (V_X, E_X) is a forest. As such, we have

$$|V_X| \geq |E_X| + 1. \tag{5}$$

Hence, we obtain

$$\begin{aligned} w(N(X, A)) &\overset{(4)}{=} |V_X \cap V_A| = |V_X| - |V_X \cap V_B| \overset{(3)}{\geq} |V_X| - |X| \\ &\overset{(5)}{\geq} 1 + |E_X| - |X| \overset{(*)}{=} 1 + \sum_{x \in X} (|x| - 1) = 1 + w(X) > w(X), \end{aligned}$$

where the inequality marked $(*)$ follows from the fact that the sets in X are pairwise disjoint. □

4 A 4/3-approximation for the hereditary 3-set packing problem

In this section, we present a polynomial-time $\frac{4}{3}$-approximation for the hereditary 3-set packing problem. For convenience, in the following, we will ignore the sets of size 1 and weight 0 contained in an instance $\mathcal{S}$ of the hereditary 3-set packing problem because we can always remove them from any feasible solution without changing its weight.

Algorithm 1: 4/3-approximation for hereditary 3-set packing

Input: an instance $\mathcal{S}$ of the hereditary 3-set packing problem
Output: a disjoint sub-collection of $\mathcal{S}$

$A \leftarrow \emptyset$
while $\exists$ *local improvement* X *of* A *of size at most* 10 **do**
  $A \leftarrow (A \setminus N(X, A)) \cup X$
return A

In order to phrase our algorithm, we formally introduce the notion of local improvement that we consider. It aims at maximizing first the weight of the solution we find, and second the number of sets of weight 2 contained in it.

Definition 18 *(local improvement)* Let $\mathcal{S}$ be an instance of the hereditary 3-set packing problem and let A be a feasible solution. We call a disjoint set collection $X \subseteq \mathcal{S}$ a *local improvement of* A *of size* $|X|$ if

- $w(X) > w(N(X, A))$ or
- $w(X) = w(N(X, A))$ and X contains more sets of weight 2 than $N(X, A)$.

We analyze Algorithm 1, which starts with the empty solution and iteratively searches for a local improvement of size at most 10 (and performs the respective swap) until no more exists. We first observe that it runs in polynomial time.

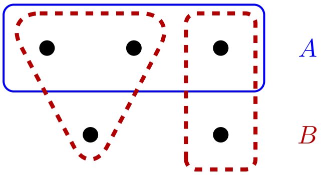
Proposition 19 *Algorithm 1 can be implemented to run in polynomial time.*

Proof A single iteration can be performed in polynomial time via brute-force enumeration. Thus, it remains to bound the number of iterations. By our definition of a local improvement, $w(A)$ can never decrease throughout the algorithm. Initially, we have $w(A) = 0$, and moreover, $w(A) \leq w(\mathcal{S}) \leq 2 \cdot |\mathcal{S}|$ holds throughout. As all weights are integral, we can infer that there are at most $2 \cdot |\mathcal{S}|$ iterations in which $w(A)$ strictly increases. In between two consecutive such iterations, there can be at most $|\mathcal{S}|$ iterations in which $w(A)$ remains constant since the number of sets of weight 2 in A strictly increases in each such iteration. All in all, we can bound the total number of iterations by $\mathcal{O}(|\mathcal{S}|^2)$. $\square$

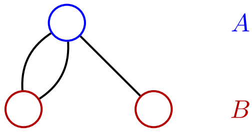
The remainder of this section is dedicated to the proof of Theorem 20, which implies that Algorithm 1 constitutes a $\frac{4}{3}$-approximation for the hereditary 3-set packing problem.

Theorem 20 *Let* $\mathcal{S}$ *be an instance of the hereditary* 3*-set packing problem and let* $A \subseteq \mathcal{S}$ *be a feasible solution such that there is no local improvement of* A *of size at most* 10*. Let further* $B \subseteq \mathcal{S}$ *be an optimum solution. Then* $w(B) \leq \frac{4}{3} \cdot w(A)$.

Let $\mathcal{S}$, w, A and B be as in the statement of the theorem. Our goal is to distribute the weights of the sets in B among the sets in A they intersect in such a way that no set in A receives more than $\frac{4}{3}$ times its own weight. We remark that each set in B must intersect at least one set in A because otherwise, it would constitute a local improvement of size 1.



(a) The figure displays two collections A (blue, solid) and B (red, dashed) consisting of pairwise disjoint sets of cardinality 2 or 3. Black dots represent set elements.



(b) The figure shows the conflict graph of $A\dot{\cup}B$. Vertices from A are drawn in blue at the top, vertices from B are drawn in red at the bottom.

Fig. 2 Construction of the conflict graph

In order to present our weight distribution, we introduce the notion of the *conflict graph*, which allows us to phrase our analysis using graph terminology. A similar construction is used in [10].

Definition 21 *(conflict graph)* The conflict graph G is defined as follows: Its vertex set is the disjoint union of A and B, i.e., $V(G) = A\dot{\cup}B$. Its edge set is obtained by adding, for each pair $(a, b) \in A \times B$, $|a \cap b|$ parallel edges connecting a to b.

See Fig. 2 for an illustration. We remark that for $X \subseteq B$, $N(X, A)$ agrees with the (graph) neighborhood of X in the bipartite graph G. Analogously, for $Y \subseteq A$, $N(Y, B)$ equals the neighborhood of Y in G. In the following, we will simultaneously interpret sets from $A\dot{\cup}B$ as the corresponding vertices in G and talk about their degree, their incident edges and their neighbors. We make the following observation.

Proposition 22 *Let $v \in V(G)$ correspond to the set $s \in A \cup B$. Then v has at most $|s|$ incident edges in G.*

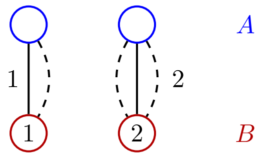
Proof As A and B both consist of pairwise disjoint sets, each element of s can induce at most one incident edge of v. □

4.1 Step 1 of the weight distribution

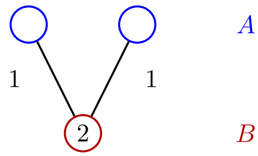
Our weight distribution proceeds in two steps. In order to describe how the weight of the vertices in B is distributed among their neighbors in A, we will say that a vertex in B *sends* a certain amount of its weight *along edges* to their endpoints in A. The first step works as follows:

Definition 23 *(Step 1 of the weight distribution)* Let B_1 consist of all sets $v \in B$ with *exactly one neighbor* in A. Each $v \in B_1$ sends its full weight to its unique neighbor in A.

Let further B_2 consist of those $v \in B$ with $w(v) = 2$ and *exactly two incident edges*, with the additional property that they connect to *two distinct sets* from A. Each $v \in B_2$ sends half of its weight (i.e., 1) along each of its edges.



(a) Every set in B_1 sends its whole weight to its unique neighbor in A (to which it may be connected via multiple edges).



(b) Every set in B_2 sends one unit of weight to each of its neighbors in A.

Fig. 3 The first step of the weight distribution

See Fig. 3 for an illustration. Observe that in the first stage, $u \in A$ receives weight precisely from the sets in $N(u, B_1 \cup B_2)$.

We first prove Lemma 24, which tells us that we can represent the total amount of weight a collection $U \subseteq A$ receives in the first step as the weight of a disjoint set collection X with $N(X, A) \subseteq U$. The construction of X will allow us to combine X with sub-collections of $B \setminus (B_1 \cup B_2)$ to obtain local improvements.

Lemma 24 *Let $U \subseteq A$. There is $X \subseteq \mathcal{S}$ with the following properties:*

(24.1) $N(X, A) \subseteq U$.
(24.2) $w(X)$ *equals the total amount of weight that U receives in the first step.*
(24.3) *There is a bijection $N(U, B_1 \cup B_2) \leftrightarrow X$ mapping $v \in B_1 \cup B_2$ to itself or to one of its two-element subsets.*

Proof We obtain X as follows: We start with $X = \emptyset$ and first add those sets in $N(U, B_1 \cup B_2)$ to X that send all of their weight to U (i.e., whose neighborhood in A is contained in U). This includes all sets in $N(U, B_1)$. Second, for each set $v \in B_2$ that has one incident edge to $u \in U$ and one incident edge to $r \in A \setminus U$, we add its two-element subset $v \setminus r$ to X. By construction, (24.1)-(24.3) hold. See Fig. 4 for an illustration. □

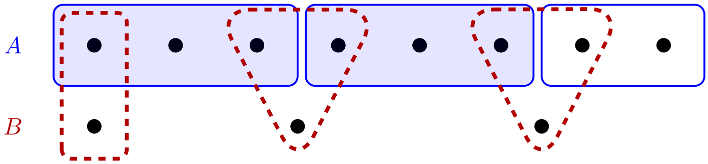
Corollary 25 *No set in A receives more than its own weight in the first step.*

Proof Assume towards a contradiction that $u \in A$ receives more than $w(u)$ in the first step. Apply Lemma 24 with $U = \{u\}$ to obtain a collection $X \subseteq \mathcal{S}$ subject to (24.1)-(24.3). Then $w(X) > w(u) = w(N(X, A))$ and (24.3) and Proposition 22 imply that X is a disjoint set family with $|X| \leq 3$. Hence, X constitutes a local improvement of size at most $3 < 10$. This contradicts our assumption that there is no local improvement of A of size at most 10. □

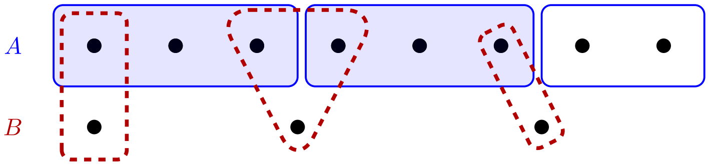
4.2 Removing "covered" sets

Definition 26 Let C consist of those sets from A that receive exactly their own weights in the first step.

The intuitive idea behind our analysis is that the sets in C are "covered" by the sets sending weight to them in the sense of Lemma 24. Hence, we can "remove" the



(a) The left red set is contained in B_1 and sends its whole weight to the unique set from A it intersects. The two triangular red sets are contained in B_2. The left one only intersects sets in A that are contained in U, whereas the right one also intersects a set in $A \setminus U$.



(b) The set collection X (red, dashed) we construct in the proof of Lemma **??** contains the left and the middle red set because they send all of their weight to U. For the right triangular set, we remove the element in which it intersects a set from $A \setminus U$. Then, we add the resulting set of cardinality 2 to X.

Fig. 4 Illustration of the construction in the proof of Lemma 24. Fig. 4a shows a collection $U \subseteq A$ of sets (blue, filled, solid), the collection $N(U, B_1 \cup B_2)$ (red, dashed) of sets the sets in U receive weight from in the first step, and further sets from A (blue, not filled, solid) the sets in $N(U, B_1 \cup B_2)$ send weight to. Fig. 4b illustrates the construction of the set collection X

sets in C from our current solution A and the sets in $B_1 \cup B_2$ from our optimum solution B. If we can find a local improvement in the remaining instance, we will use Lemma 24 to transform it into a local improvement in the original instance, leading to a contradiction. See Lemma 27 for an example of how to apply this reasoning. Hence, exploiting the fact that there cannot be a local improvement in the remaining instance, we can design the second step of the weight distribution in such a way that overall, no set in A receives more than $\frac{4}{3}$ times its own weight.

4.3 Step 2 of the weight distribution

In order to define the second step of the weight distribution, we make the following observations:

Lemma 27 *There is no $v \in B \setminus (B_1 \cup B_2)$ with $w(N(v, A \setminus C)) < w(v)$.*

Proof Assume towards a contradiction that there is $v \in B \setminus (B_1 \cup B_2)$ with $w(N(v, A \setminus C)) < w(v)$. Apply Lemma 24 to $U := N(v, C)$ to obtain X subject to (24.1)-(24.3). By (24.3), $X \dot\cup \{v\}$ consists of pairwise disjoint sets. Proposition 22 further yields $|N(v, C)| \leq |v| \leq 3$, and, thus, $|X| = |N(N(v, C), B_1 \cup B_2)| \leq 9$ by (24.3). Finally, $w(X) = w(N(v, C))$ by (24.2) and since sets from C receive exactly their own weights in the first step. Hence, (24.1) yields

$$w(X \cup \{v\}) = w(X) + w(v) > w(N(v, C)) + w(N(v, A \setminus C)) = w(N(X \cup \{v\}, A)).$$

So $X \cup \{v\}$ is a local improvement of A of size at most 10, a contradiction. □

Proposition 28 *Let $v \in B \setminus (B_1 \cup B_2)$. Then*

(i) v has at least one neighbor in $A \setminus C$.
(ii) If $w(v) = 1$, then v has exactly two neighbors in A.
(iii) If $w(v) = 2$, then v has three incident edges.

Proof (i) follows from Lemma 27. For (ii) and (iii), we remind ourselves that each $v \in B$ has at most $|v|$ neighbors/incident edges, but at least 1 neighbor in A by Proposition 22 and since $\{v\}$ would constitute a local improvement otherwise. (ii) holds since $v \in B_1$ otherwise. For (iii), we observe that in case v has at most 2 incident edges, then either v has only one neighbor in A, or two distinct neighbors to which it is connected by a single edge each. In either case, we have $v \in B_1 \cup B_2$. □

Definition 29 *(Step 2 of the weight distribution)* Let $v \in B \setminus (B_1 \cup B_2)$ with $w(v) = 1$.

(a) If v has a neighbor in C, then this neighbor receives $\frac{1}{3}$ and the neighbor in $A \setminus C$ receives $\frac{2}{3}$.
(b) Otherwise, both neighbors in $A \setminus C$ receive $\frac{1}{2}$.

Now, let $v \in B \setminus (B_1 \cup B_2)$ with $w(v) = 2$.

(c) If v has degree 1 to $A \setminus C$, then v sends $\frac{1}{3}$ along each edge to C and $\frac{4}{3}$ to the neighbor in $A \setminus C$. Note that this neighbor must have a weight of 2 by Lemma 27.
(d) If v has degree 2 to $A \setminus C$, v sends 1 along each edge to a vertex in $A \setminus C$ of weight 2, $\frac{2}{3}$ along each edge to a vertex in $A \setminus C$ of weight 1, and the remaining amount to the neighbor in C.
(e) If all three incident edges of v connect to $A \setminus C$, then v sends $\frac{2}{3}$ along each of these edges.

We denote the set of vertices to which case ℓ with $\ell \in \{a, b, c, d, e\}$ applies by B_ℓ.

See Fig. 5 for an illustration.

4.4 No set in *C* receives more than 4/3 times its weight

Lemma 30 *Let $v \in B_d$ and let $u \in N(v, C)$ be the unique neighbor of v in C. If u receives more than $\frac{1}{3}$ from v, then $w(u) = 2$ and u has exactly one incident edge to $B \setminus (B_1 \cup B_2)$.*

Proof Denote the endpoints of the two edges connecting v to $A \setminus C$ by u_1 and u_2. Assume u receives more than $\frac{1}{3}$ from v. Then $w(u_1) = w(u_2) = 1$. In particular, u_1 and u_2 are distinct by Lemma 27. Apply Lemma 24 to $U := \{u\}$ to obtain X subject to (24.1)-(24.3). Then by (24.3), $Y := X \dot\cup \{v\}$ is a disjoint collection of sets. Moreover, Proposition 22 yields

$$|X| \overset{(24.3)}{=} |N(u, B_1 \cup B_2)| \leq |u| \leq 3.$$

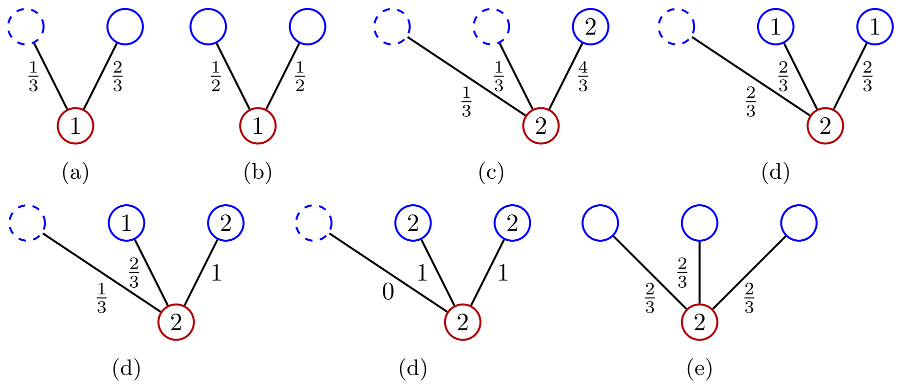


Fig. 5 Illustration of the second step of the weight distribution. Blue circles in the top row indicate sets from A, if they are dashed, the corresponding set is contained in C. Red circles in the bottom row indicate sets from $B\backslash(B_1 \cup B_2)$. The number within a circle indicates the weight of the corresponding set in case it is relevant. Even though drawn as individual circles, the endpoints in A of the incident edges of a set $v \in B \setminus (B_1 \cup B_2)$ need not be distinct. For example, in (e), two of the sets represented by the blue circles may agree, in which case the corresponding set receives $\frac{4}{3}$

Hence, $|Y| \leq 4$. By (24.2) and as $u \in C$ receives its own weight in the first step, we get $w(u) = w(X)$. Thus, $w(u_1) + w(u_2) = 1 + 1 = 2 = w(v)$ results in

$$w(N(Y, A)) \overset{(24.1)}{=} w(u) + w(u_1) + w(u_2) = w(X) + w(v) = w(Y).$$

As Y does not constitute a local improvement, $N(Y, A) = \{u_1, u_2, u\}$ contains at least as many vertices of weight 2 as Y. As $w(u_1) = w(u_2) = 1$, but $w(v) = 2$, this implies that $w(u) = 2$ and that all elements of X have a weight of 1. By (24.2), this implies $|X| = 2$, and by (24.3), u intersects sets from $B_1 \cup B_2$ in at least two distinct elements in total. In particular, $\{u, v\}$ is the only edge connecting u to $B \setminus (B_1 \cup B_2)$ by Proposition 22. □

Lemma 31 *Each set in C receives at most $\frac{4}{3}$ times its own weight during our weight distribution.*

Proof First, let $u \in C$ with $w(u) = 1$. Then u receives 1 in the first step and has at most one incident edge to $B \setminus (B_1 \cup B_2)$. Via this edge, u receives at most $\frac{1}{3}$, which is clear for the cases (a) and (c), and follows from Lemma 30 for case (d). Thus, u receives at most $\frac{4}{3} = \frac{4}{3} \cdot w(u)$ in total.

Next, let $u \in C$ with $w(u) = 2$. Then u receives 2 in the first step and u has at most two incident edges to $B \setminus (B_1 \cup B_2)$. If u has two incident edges to $B \setminus (B_1 \cup B_2)$, then u can receive at most $\frac{1}{3}$ via each of them: This is clear for the cases (a) and (c), and follows from Lemma 30 for case (d). Thus, u receives at most $\frac{8}{3} = \frac{4}{3} \cdot w(u)$ in total. If u has one incident edge to $B \setminus (B_1 \cup B_2)$, then the maximum amount u can receive via this edge is $\frac{2}{3}$. Again, u receives at most $\frac{8}{3}$ in total. □

4.5 No set in $A \setminus C$ receives more than 4/3 times its weight

In order to make sure that no vertex from $A \setminus C$ receives more than $\frac{4}{3}$ times its own weight, we need Lemma 32, which essentially states the following:

- If a vertex $u \in A \setminus C$ with $w(u) = 2$ receives $\frac{4}{3}$ from a vertex in B_c, then it does not receive weight from any further vertex in $B_1 \cup B_2 \cup B_c \cup B_d$.
- A vertex $u \in A \setminus C$ with $w(u) = 2$ may, in total, receive at most 2 units of weight from vertices in $B_1 \cup B_2 \cup B_d$.

Lemma 32 *Let $u \in A \setminus C$ with $w(u) = 2$. Denote the set of vertices $v \in B_d$ that are connected to u by one/two parallel edges by D_1 and D_2, respectively.*
Then $|N(u, B_1 \cup B_2)| + 2|N(u, B_c)| + |D_1| + 2|D_2| \leq 2$.

Our strategy to prove Lemma 32 can be summarized as follows: We show that similar to Lemma 24, we can represent the term $2|N(u, B_c)| + |D_1| + 2|D_2|$ as the weight of a disjoint set collection Y with $N(Y, A \setminus C) \subseteq \{u\}$. Y consists of subsets of sets in $B \setminus (B_1 \cup B_2)$.

We then apply Lemma 24 to $U := N(Y, C) \cup \{u\}$ to obtain a set collection X. We argue that if $|N(u, B_1 \cup B_2)| + 2|N(u, B_c)| + |D_1| + 2|D_2| > 2 = w(u)$, then $X \cup Y$ constitutes a local improvement. In order to arrive at the desired contradiction, we need to initially restrict our attention to a minimal sub-family $\bar{Y} \subseteq N(u, B_c \cup B_d)$ with $|N(u, B_1 \cup B_2)| + 2|\bar{Y} \cap B_c| + |\bar{Y} \cap D_1| + 2|\bar{Y} \cap D_2| > 2$, which allows us to conclude that $|X \cup Y| \leq 10$.

Proof of Lemma 32 Assume towards a contradiction that

$$|N(u, B_1 \cup B_2)| + 2|N(u, B_c)| + |D_1| + 2|D_2| \geq 3.$$

Note that $|N(u, B_1 \cup B_2)| \leq 1$ because $u \notin C$ and u receives at least one unit of weight per neighbor in $B_1 \cup B_2$. Pick an inclusion-wise minimal set $\bar{Y} \subseteq N(u, B_c \cup B_d)$ such that

$$|N(u, B_1 \cup B_2)| + 2|\bar{Y} \cap B_c| + |\bar{Y} \cap D_1| + 2|\bar{Y} \cap D_2| \geq 3. \tag{6}$$

Then

$$|N(u, B_1 \cup B_2)| + 2|\bar{Y} \cap B_c| + |\bar{Y} \cap D_1| + 2|\bar{Y} \cap D_2| = 3, \text{ or} \tag{7}$$

$$\bar{Y} \cap D_1 = \emptyset \text{ and } |N(u, B_1 \cup B_2)| + 2|\bar{Y} \cap B_c| + 2|\bar{Y} \cap D_2| = 4. \tag{8}$$

We construct a set collection Y as follows: We start with $Y = \emptyset$ and first add all sets contained in $\bar{Y} \cap (B_c \cup D_2)$ to Y. Note that for such a set v, $N(v, A \setminus C) = \{u\}$ (see Fig. 5). Second, for each $v \in \bar{Y} \cap D_1$, let v' be the set of cardinality 2 containing the element in which v intersects a set from C, and the element in which v intersects u.

Add v' to Y. Then Y has the following properties:

$$N(Y,A) \subseteq C \cup \{u\} \tag{9}$$

$$|Y| = |\bar{Y} \cap B_c| + |\bar{Y} \cap D_1| + |\bar{Y} \cap D_2| \tag{10}$$

$$w(Y) = 2|\bar{Y} \cap B_c| + |\bar{Y} \cap D_1| + 2|\bar{Y} \cap D_2| \overset{(6)}{\geq} 3 - |N(u, B_1 \cup B_2)| \tag{11}$$

$$|N(Y,C)| \leq 2|\bar{Y} \cap B_c| + |\bar{Y} \cap D_1| + |\bar{Y} \cap D_2|. \tag{12}$$

The inequality (12) holds since each vertex in B_c has at most 2 neighbors in C, and each vertex in B_d has at most one neighbor in C (see Fig. 5).

Let $U := N(Y,C) \cup \{u\}$. Apply Lemma 24 to obtain X subject to (24.1)-(24.3). Then by (24.2), we get

$$w(X) \geq w(N(Y,C)) + |N(u, B_1 \cup B_2)| \tag{13}$$

because each set in $N(Y,C)$ receives its weight in the first step, and u receives at least 1 per neighbor in $B_1 \cup B_2$. By (24.3) and since the sets in Y constitute disjoint subsets of sets in $B \setminus (B_1 \cup B_2)$, $X \dot{\cup} Y$ is a family of pairwise disjoint sets. We would like to show that $X \cup Y$ yields a local improvement of size at most 10. By (13) and (11), we obtain

$$\begin{aligned} w(X \cup Y) &= w(X) + w(Y) \geq 3 + w(N(Y,C)) \\ &> w(u) + w(N(Y,C)) \geq w(N(X \cup Y, A)), \end{aligned}$$

where $N(X \cup Y, A) \subseteq N(Y,C) \cup \{u\}$ follows from (24.1) and (9). Thus, it remains to show that $|X \cup Y| \leq 10$. By (24.3), we have

$$\begin{aligned} |X| = |N(U, B_1 \cup B_2)| &\leq |N(u, B_1 \cup B_2)| + |N(N(Y,C), B_1 \cup B_2)| \\ &\leq |N(u, B_1 \cup B_2)| + 2|N(Y,C)|. \end{aligned} \tag{14}$$

For the last inequality, we used Proposition 22, which tells us that each set $z \in N(Y,C)$ has degree at most 3 in G. In addition, z must intersect at least one set from Y, and thus, from $\bar{Y}$. In particular, z has at least one incident edge to $B \setminus (B_1 \cup B_2) \supseteq \bar{Y}$, and, thus, at most two incident edges to $B_1 \cup B_2$. Hence, we obtain

$$\begin{aligned} |Y| + |X| &\overset{(14)}{\leq} |Y| + |N(u, B_1 \cup B_2)| + 2|N(Y,C)| \\ &\underset{(12)}{\overset{(10)}{\leq}} \underbrace{|N(u, B_1 \cup B_2)| + 5|\bar{Y} \cap B_c| + 3|\bar{Y} \cap D_1| + 3|\bar{Y} \cap D_2|}_{=:(*)}. \end{aligned}$$

If (7) holds, we can bound $(*)$ by 3 times the right-hand side of (7) and deduce an upper bound of 9. In case (8) is satisfied, we can bound $(*)$ by $\frac{5}{2}$ times the right-hand side of (8) and obtain an upper bound of 10. Thus, we have found a local improvement of size at most 10, a contradiction. □

Lemma 33 *Each set $u \in A \setminus C$ receives at most $\frac{4}{3}$ times its own weight during our weight distribution.*

Proof If $w(u) = 1$, then u cannot receive any weight in the first step because otherwise, it would receive at least 1 and be contained in C. Moreover, u has at most two incident edges and receives at most $\frac{2}{3}$ via either of them in the second step.

Next, consider the case where $w(u) = 2$. If u receives $\frac{4}{3}$ from a vertex in B_c, then by Lemma 32, there is no further vertex in $B_1 \cup B_2 \cup B_c \cup B_d$ from which u receives weight. As u receives at most $\frac{2}{3}$ per edge in all remaining cases, u receives at most $\frac{4}{3} + 2 \cdot \frac{2}{3} = \frac{8}{3} = \frac{4}{3} \cdot w(u)$. Finally, assume that $N(u, B_c) = \emptyset$. In the first step, u can receive at most 1 in total (otherwise, $u \in C$) and this can only happen if u has a neighbor in $B_1 \cup B_2$. The maximum amount u can receive through one edge in the second step is 1, and this can only happen in situation (d). By Lemma 32, there are at most 2 edges via which u receives 1. Moreover, u can receive at most $\frac{2}{3}$ via the remaining edges. Again, we obtain an upper bound of $1 + 1 + \frac{2}{3} = \frac{8}{3}$ on the total weight received. □

Combining Lemma 31 and Lemma 33 proves Theorem 20. Together with Proposition 19 and Theorem 4, we obtain Corollary 34.

Corollary 34 *There is a polynomial-time $\frac{4}{3}$-approximation algorithm for the MLSA in dags.*

5 Conclusion

In this paper, we have presented a simple local search-based $\frac{4}{3}$-approximation for the MLSA in dags, improving upon the previous state-of-the-art of $\frac{7}{5}$ due to Fernandes and Lintzmayer [10]. Our result is based on a reduction to the hereditary 3-set packing problem given in [10]. Given that in [10], the reduction is performed in a rather complicated ad hoc fashion requiring several pages of analysis, the connection between the MLSA in dags and the hereditary 3-set packing problem remains rather opaque. In this work, we have shown via a very simple reduction that the MLSA in dags is, at its core, a hereditary set packing problem. We have further explored the general connection between approximation guarantees for the hereditary set packing problem and its restriction to instances with bounded set sizes. More precisely, we have seen that an α-approximation algorithm for the hereditary k-set packing problem implies a $\max\{\alpha, \frac{k+1}{k}\}$-approximation for the hereditary set packing problem. The relation between approximation guarantees for the hereditary 3-set packing problem and the MLSA in dags obtained by Fernandes and Lintzmayer [10] corresponds to the special case $k = 3$.

Finally, we have established a lower bound of $2 - \frac{2}{k}$ on the approximation guarantee achieved by a local search algorithm for the hereditary k-set packing problem that only considers local improvements of constant size.

As a result, we can conclude that the approximation guarantee of $\frac{4}{3}$ is best possible for the type of algorithm we consider.

Whether a better guarantee than $\frac{4}{3}$ can be, for example, obtained via a reduction to the hereditary k-set packing problem with $k \geq 4$ and an algorithm that considers local improvements of super-constant size remains a question for future research. Note that the state-of-the-art approximation algorithms for the unweighted k-set packing problem crucially rely on also considering well-structured local improvements of logarithmic size [4, 11].

Finally, it would be interesting to see whether there are other problems that can, in a natural way, be interpreted as a special type of set packing problem that allows for improved approximation guarantees.

Acknowledgements Meike Neuwohner was supported by the Engineering and Physical Sciences Research Council, part of UK Research and Innovation, grant ref. EP/X030989/1.

Data availability statement No data are associated with this article. Data sharing is not applicable to this article.

References

1. Alon, Noga, Fomin, Fedor V., Gutin, Gregory, Krivelevich, Michael, Saurabh, Saket: Spanning directed trees with many leaves. SIAM J. Discret. Math. **23**(1), 466–476 (2009). https://doi.org/10.1137/070710494
2. Binkele-Raible, Daniel, Fernau, Henning, Fomin, Fedor V., Lokshtanov, Daniel, Saurabh, Saket, Villanger, Yngve: Kernel(s) for problems with no kernel: on out-trees with many leaves. ACM Trans. Algorithms (2012). https://doi.org/10.1145/2344422.2344428
3. Bonsma, Paul: Max-leaves spanning tree is APX-hard for cubic graphs. J. Discrete Algorithms **12**, 14–23 (2012). https://doi.org/10.1016/j.jda.2011.06.005
4. Cygan, Marek: Improved Approximation for 3-Dimensional Matching via Bounded Pathwidth Local Search. In 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA, pp. 509–518. IEEE Computer Society, (2013)
5. Daligault, Jean, Thomassé, Stéphan.: On finding directed trees with many leaves. In: Chen, Jianer, Fomin, Fedor V. (eds.) Parameterized and Exact Computation, pp. 86–97. Springer, Berlin Heidelberg (2009)
6. Drescher, Matthew, Vetta, Adrian: An approximation algorithm for the maximum leaf spanning arborescence problem. ACM Trans. Algorithms (2010). https://doi.org/10.1145/1798596.1798599
7. Edmonds, Jack: Maximum matching and a polyhedron with 0,1-vertices. J. Res. Nat. Bureau Stand. Sect. B Math. Math. Phys. **69B**, 125–130 (1965). https://doi.org/10.6028/jres.069b.013
8. Erdős, Paul, Sachs, Horst: Reguläre Graphen gegebener Taillenweite mit minimaler Knotenzahl. Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe, 12(3):251–257 (1963)
9. Fernandes, Cristina G., Lintzmayer, Carla N.: Leafy spanning arborescences in dags. Discret. Appl. Math. **323**, 217–227 (2022). https://doi.org/10.1016/j.dam.2021.06.018
10. Fernandes, Cristina G., Lintzmayer, Carla N.: How heavy independent sets help to find arborescences with many leaves in dags. J. Comput. Syst. Sci. **135**, 158–174 (2023). https://doi.org/10.1016/j.jcss.2023.02.006

11. Fürer, Martin, Yu, Huiwen: Approximating the k-Set Packing Problem by Local Improvements. In: International Symposium on Combinatorial Optimization, pp. 408–420. (2014)
12. Galbiati, G., Maffioli, F., Morzenti, A.: A short note on the approximability of the maximum leaves spanning tree problem. Inf. Process. Lett. **52**(1), 45–49 (1994). https://doi.org/10.1016/0020-0190(94)90139-2
13. Garey, Michael R., Johnson, David S.: *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, (1990)
14. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. Algorithmica **20**(4), 374–387 (1998). https://doi.org/10.1007/PL00009201
15. Karp, Richard M.: Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations. Plenum Press, (1972)
16. Khanna, Sanjeev, Motwani, Rajeev, Sudan, Madhu, Vazirani, Umesh: On syntactic versus computational views of approximability. SIAM J. Comput. **28**(1), 164–191 (1998). https://doi.org/10.1137/S0097539795286612
17. Neuwohner, Meike: The limits of local search for weighted k-set packing. Math. Prog. (2023). https://doi.org/10.1007/s10107-023-02026-3
18. Neuwohner, Meike: Passing the limits of pure local search for weighted k-set packing. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1090–1137. Society for Industrial and Applied Mathematics, (2023)
19. Neuwohner, Meike: A 4/3-approximation for the maximum leaf spanning arborescence problem in dags. In Jens Vygen and Jarosław Byrka, editors, *Integer Programming and Combinatorial Optimization*, volume 14679 of *Lecture Notes in Computer Science*, pp. 337–350. Springer, (2024)
20. Ruan, Lu., Hongwei, Du., Jia, Xiaohua, Weili, Wu., Li, Yingshu, Ko, Ker-I.: A greedy approximation for minimum connected dominating sets. Theoret. Comput. Sci. **329**(1–3), 325–330 (2004). https://doi.org/10.1016/j.tcs.2004.08.013
21. Schwartges, Nadine, Spoerhase, Joachim, Wolff, Alexander: Approximation algorithms for the maximum leaf spanning tree problem on acyclic digraphs. In: Solis-Oba, Roberto, Persiano, Giuseppe (eds.) Approximation and Online Algorithms, pp. 77–88. Springer, Berlin Heidelberg (2012)
22. Solis-Oba, Roberto, Bonsma, Paul S., Lowski, Stefanie: A 2-approximation algorithm for finding a spanning tree with maximum number of leaves. Algorithmica **77**, 374–388 (2015). https://doi.org/10.1007/s00453-015-0080-0
23. Thiery, Theophile, Ward, Justin: An improved approximation for maximum weighted k-set packing. In: Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1138–1162. Society for Industrial and Applied Mathematics, (2023)
24. Trevisan, Luca: Non-approximability results for optimization problems on bounded degree instances. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC '01, pp. 453-461, New York, NY, USA, (2001)