Theory Comput Syst (2016) 58:60–93
DOI 10.1007/s00224-014-9565-5

Degree-Constrained Graph Orientation: Maximum Satisfaction and Minimum Violation

Yuichi Asahiro · Jesper Jansson · Eiji Miyano · Hirotaka Ono

Received: 3 February 2014 / Accepted: 23 August 2014 / Published online: 19 December 2014

Abstract A *degree-constrained graph orientation* of an undirected graph G is an assignment of a direction to each edge in G such that the outdegree of every vertex in the resulting directed graph satisfies a specified lower and/or upper bound. Such graph orientations have been studied for a long time and various characterizations of their existence are known. In this paper, we consider four related optimization problems introduced in reference (Asahiro et al. LNCS **7422**, 332–343 (2012)): For any fixed non-negative integer W, the problems MAX W-LIGHT, MIN W-LIGHT, MAX W-HEAVY, and MIN W-HEAVY take as input an undirected graph G and ask for an orientation of G that maximizes or minimizes the number of vertices with outdegree at most W or at least W. As shown in Asahiro et al. (LNCS **7422**, 332–343 (2012)), the problems' computational complexity vary with W. Here, we present

An extended abstract of this paper appeared in *Proceedings of the 11th International Workshop on Approximation and Online Algorithms* (WAOA 2013), volume 8447 of *Lecture Notes in Computer Science*, pp. 24–36, Springer International Publishing Switzerland, 2014.

Y. Asahiro (✉)
Department of Information Science, Kyushu Sangyo University, Higashi-ku, Fukuoka 813-8503, Japan
e-mail: asahiro@is.kyusan-u.ac.jp

J. Jansson
Laboratory of Mathematical Bioinformatics, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan
e-mail: jj@kuicr.kyoto-u.ac.jp

E. Miyano
Department of Systems Design and Informatics, Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502, Japan
e-mail: miyano@ces.kyutech.ac.jp

H. Ono
Department of Economic Engineering, Kyushu University, Higashi-ku, Fukuoka 812-8581, Japan
e-mail: hirotaka@econ.kyushu-u.ac.jp

several new positive and negative results related to their polynomial-time approximability, thereby extending the results from Asahiro et al. (LNCS **7422**, 332–343 (2012)).

Keywords Graph orientation · Degree constraint · (In)approximability · Submodular function · Greedy algorithm

1 Introduction

Let $G = (V, E)$ be an undirected (multi-)graph. An *orientation of* G is a function that maps each undirected edge $\{u, v\}$ in E to one of the two possible directed edges (u, v) and (v, u). For any orientation Λ of G, define $\Lambda(E) = \bigcup_{e \in E}\{\Lambda(e)\}$ and let $\Lambda(G)$ denote the directed graph $(V, \Lambda(E))$. For any vertex $u \in V$, the *outdegree of* u *under* Λ is defined as $d^+_\Lambda(u) = |\{(u, v) : (u, v) \in \Lambda(E)\}|$, i.e., the number of outgoing edges from u in $\Lambda(G)$. For any non-negative integer W, a vertex $u \in V$ is called W-light in $\Lambda(G)$ if $d^+_\Lambda(u) \le W$, and W-heavy in $\Lambda(G)$ if $d^+_\Lambda(u) \ge W$. For any $U \subseteq V$, if all the vertices in U are W-light (resp., W-heavy), we say that U is W-light (resp., W-heavy).

The optimization problems MAX W-LIGHT, MIN W-LIGHT, MAX W-HEAVY, and MIN W-HEAVY, where W is any fixed non-negative integer, were introduced in [5]. In each problem, the input is an undirected (multi-)graph $G = (V, E)$ and the objective is to output an orientation Λ of G such that:

- MAX W-LIGHT: $\left|\{u \in V : d^+_\Lambda(u) \le W\}\right|$ is maximized
- MIN W-LIGHT: $\left|\{u \in V : d^+_\Lambda(u) \le W\}\right|$ is minimized
- MAX W-HEAVY: $\left|\{u \in V : d^+_\Lambda(u) \ge W\}\right|$ is maximized
- MIN W-HEAVY: $\left|\{u \in V : d^+_\Lambda(u) \ge W\}\right|$ is minimized

We write $n = |V|$ and $m = |E|$ for the input graph G. The *degree of* u *in* G, denoted by $d(u)$, is the number of edges that are incident to u in G, and we define $\delta = \min\{d(u) : u \in V\}$ and $\Delta = \max\{d(u) : u \in V\}$.

Observe that MAX W-LIGHT and MIN $(W + 1)$-HEAVY are *supplementary problems* in the sense that an exact algorithm for one gives an exact algorithm for the other although their polynomial-time approximability properties may differ. The same observation holds for the pair MIN W-LIGHT and MAX $(W + 1)$-HEAVY.

The computational complexity of MAX W-LIGHT, MIN W-LIGHT, MAX W-HEAVY, and MIN W-HEAVY was investigated for different values of W in [5]. As observed in [5], the special case of MAX 0-LIGHT is identical to the well-known MAXIMUM INDEPENDENT SET problem, and its supplementary problem MIN 1-HEAVY is identical to MINIMUM VERTEX COVER. Thus, allowing the value of W to vary yields a natural generalization of MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER. However, for many values of W, the computational complexity was unknown. In this paper, we establish a number of new results on the polynomial-time approximability of these problems.

1.1 New results

Below is a list of previous results from [5] and the new results presented in this paper. See Table 1 for a summary.

- Max W-Light: It is known that Max 0-Light cannot be approximated within a ratio of $n^{1-\varepsilon}$ for any positive constant ε in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ [5, 30]. Theorem 4 in Section 5.1 below proves that for every fixed $W \geq 1$, Max W-Light cannot be approximated within $(n/W)^{1-\varepsilon}$ in polynomial time, unless $\mathcal{P} = \mathcal{NP}$. On the positive side, Theorem 1 in Section 3 provides a polynomial-time $(n/(2W+1))$-approximation algorithm for Max W-Light.
- Min W-Heavy: Min 1-Heavy cannot be approximated within 1.3606 in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ [5, 10]. Corollary 1 in Section 5.2 extends this inapproximability result to hold for Min W-Heavy for every fixed $W \geq 2$. We also show how to approximate Min W-Heavy within a ratio of $\ln(\Delta - W + 1) + 1$ in polynomial time for every fixed $W \geq 2$ in Theorem 3 in Section 4.2.

Table 1 Summary of the results from [5] and the new results in this paper

W		
	Max W-Light	Min $(W+1)$-Heavy
$= 0$	Identical to Maximum Independent Set [5]	Identical to Minimum Vertex Cover [5]
≥ 1	Tractable for trees [5], $(n/(2W+1))$-approximable (Theorem 1, Section 3), $(n/W)^{1-\varepsilon}$-inapproximable (Theorem 4, Section 5.1)	Tractable for trees [5], $(\ln(\Delta - W) + 1)$-approximable (Theorem 3, Section 4.2), 1.3606-inapproximable (Corollary 1, Section 5.2)
	Min W-Light	Max $(W+1)$-Heavy
$= 0$	Tractable [5]	Tractable [5]
≥ 0	Tractable for outerplanar graphs [5]	Tractable for outerplanar graphs [5]
≥ 1	$(\ln(W+1)+1)$-approximable (Theorem 2, Section 4.1)	2-approximable for planar graphs [5], $(W+2)$-approximable [5]
≥ 2	$\mathcal{NP}$-hard for planar graphs [5]	$\mathcal{NP}$-hard for planar graphs [5]
large	$(\ln(W+1) - O(\log\log W))$ -inapproximable (Corollary 2, Section 5.3)	$(n/W)^{1/2-\varepsilon}$-inapproximable (Theorem 7, Section 5.4), $W^{1-\varepsilon}$-inapproximable (Corollary 3, Section 5.4)

"Tractable" means "exactly solvable in polynomial time"

– MIN W-LIGHT: A polynomial-time $(W+1)$-approximation algorithm was given in [5]. Theorem 2 in Section 4.1 improves the approximation ratio to $\ln(W+1)+1$ for any $W \geq 1$. Moreover, Corollary 2 in Section 5.3 shows that for sufficiently large values of W, MIN W-LIGHT is $\mathcal{NP}$-hard to approximate within $\ln(W+1) - O(\log\log W)$, implying that our $(\ln(W+1)+1)$-approximation is almost tight.
– MAX W-HEAVY: It was shown in [5] that MAX 1-HEAVY and MIN 0-LIGHT are in $\mathcal{P}$, but MAX $(W+1)$-HEAVY and MIN W-LIGHT are $\mathcal{NP}$-hard for every fixed $W \geq 2$. Theorem 7 in Section 5.4 strengthens the latter result for MAX W-HEAVY by proving that for sufficiently large values of W, the problem is $\mathcal{NP}$-hard to approximate within $(n/W)^{1/2-\varepsilon}$ for any $\varepsilon > 0$. Furthermore, Corollary 3 in Section 5.4 shows that it is also $\mathcal{NP}$-hard to approximate within $W^{1-\varepsilon}$ for any $\varepsilon > 0$ and $W = \Theta(n^{1/3})$. (Note that the best known polynomial-time approximation ratio for MAX W-HEAVY is $W+1$ [5].)
– The computational complexity of MAX 2-HEAVY and MIN 1-LIGHT for arbitrary input graphs has still not been resolved, but this paper considers two special cases: (i) $\Delta \leq 3$; and (ii) $\delta \geq 4$. Corollary 9 in Section 6.2 and Corollary 8 in Section 6.1 demonstrate that both problems can be solved in polynomial time for case (i) and case (ii), respectively.

1.2 Motivation

Graph orientations that optimize certain objective functions involving the resulting directed graph or that satisfy some special property such as acyclicity [26] or k-edge connectivity [9, 21, 24] have many applications to graph theory, combinatorial optimization, scheduling (load balancing), resource allocation, and efficient data structures. For example, an orientation that minimizes the maximum outdegree [3, 8, 11, 19, 28] can be used to support fast vertex adjacency queries in a sparse graph by storing each edge in exactly one of its two incident vertices' adjacency lists while ensuring that all adjacency lists are short [8]. There are many other optimization criteria for graph orientations besides these; see [4] or chapter 61 in [25] for more details and additional references.

Degree-constrained graph orientations [14, 15, 18, 20], studied in this paper, are one particular type of graph orientations that arise when a lower degree bound $W^l(v)$ and/or an upper degree bound $W^u(v)$ for each vertex v in the input graph are specified, and the outdegree of v in any valid graph orientation is required to lie in the range $W^l(v), \ldots, W^u(v)$. Obviously, a graph does not always have such an orientation, and in this case, one might want to compute an orientation that best fits the outdegree constraints according to some well-defined criteria [4, 5]. In case $W^l(v) = 0$ and $W^u(v) = W$ for every vertex v in the input graph, where W is a non-negative integer, and the objective is to maximize (resp., minimize) the number of vertices that satisfy (resp., violate) the outdegree constraints, then we obtain MAX W-LIGHT (resp., MIN $(W+1)$-HEAVY). Similarly, if $W^l(v) = W+1$ and $W^u(v) = \infty$ for every vertex v in the input graph, then we obtain MAX $(W+1)$-HEAVY and MIN W-LIGHT.

2 Preliminaries

Let G be an undirected graph. We denote the vertex set and the edge set of G by $V(G)$ and $E(G)$, respectively. For any $U \subseteq V(G)$, the subgraph of G induced by U is denoted by $G[U]$. For any fixed integer $W \geq 0$, an orientation of a graph is called a *W-orientation* if the maximum outdegree is at most W. If a W-orientation exists, the graph is said to be *W-orientable*. For any $S \subseteq V(G)$, we let $E(S)$ denote the subset of edges whose both endpoints belong to S, i.e., $G[S] = (S, E(S))$. Also, for any two disjoint subsets $S, T \subseteq V$, we write $E(S, T)$ to denote the subset of all edges with one endpoint belonging to S and the other to T. The ratio $|E(S)|/|S|$ is called the *density of S*. The *maximum density ρ_G* of an undirected graph G is defined by $\rho_G = \max_{S \subseteq V} \left\lceil \frac{|E(S)|}{|S|} \right\rceil$. [1] We denote a subgraph $G[V(G) \setminus S]$ of G whose vertex set and edge set are $V(G) \setminus S$ and $E(V(G) \setminus S)$, respectively, by $G \setminus S$. Finally, an orientation Λ of an undirected graph G is called *Eulerian* if $d^+_\Lambda(v) = d(v) - d^+_{\Lambda(v)}$, i.e., if the outdegree equals the indegree for every vertex v in $V(G)$.

It is known [14] that finding the maximum density of a graph is equivalent to finding the smallest integer W such that the graph is W-orientable:

Proposition 1 ([14]) *Any graph G is W-orientable if and only if $\rho_G \leq W$.*

Throughout the paper, we use the notation K_i to denote the complete graph with i vertices for any positive integer i. The following immediate consequence of Proposition 1 plays an important role:

Proposition 2 *The complete graph K_{2W+1} has an orientation in which the indegree and the outdegree of every vertex are equal to W.*

Note that the orientation referred to in Proposition 2 is Eulerian.

Proposition 3 (p. 91 of [25]) *Given a graph G with all degrees even, an Eulerian orientation of G can be found in $O(m)$ time.*

We extend the notion of the maximum density to oriented graphs as follows:

Proposition 4 *Consider an undirected graph G and an orientation Λ of G, and assume that m' edges in $E(U, V(G) \setminus U)$ for a subset U of vertices are oriented outwards from U to $V(G) \setminus U$ in Λ. Then the average outdegree of the vertices in U is $(|E(U)| + m')/|U|$. As a result, there exists a vertex $v \in U$ with $d^+_\Lambda(v) \geq \lceil (|E(U)| + m')/|U| \rceil$.*

[1]The average degree of a vertex in the subgraph $G[S]$ is given by the density $|E(S)|/|S|$, which implies that there is a vertex of degree at least this value. Since $|E(S)|/|S|$ is not always an integer, the maximum degree of the graph $G[S]$ is at least $\lceil |E(S)|/|S| \rceil$. This is why we use the ceiling function in the definition of the maximum density.

3 An approximation algorithm for MAX W-LIGHT with $W \geq 1$

Since MAX 0-LIGHT is identical to MAXIMUM INDEPENDENT SET [5], it cannot be approximated within a ratio of $n^{1-\varepsilon}$ in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ [30], while it can be approximated within a ratio of $O(n(\log\log n)^2/(\log n)^3)$ [12]. This section shows how to approximate MAX W-LIGHT for any $W \geq 1$ within a ratio of $n/(2W+1)$ in polynomial time. This bound is almost tight due to the inapproximability ratio $(n/W)^{1-\varepsilon}$ derived in Section 5.1 below.

First, consider the following simple algorithm, named EulerO, for orienting the edges of an input graph G based on Proposition 3 (see also [13]):

Algorithm EulerO
Input: An undirected graph $G = (V, E)$
Output: An orientation Λ of G

1. While there exist vertices having odd degrees, select any pair of two such vertices and insert an edge between them.
2. Let G^+ be the obtained multigraph with all degrees even and let E^+ be the set of added edges in Step 1, i.e., $E^+ := E(G^+) \setminus E(G)$.[2]
3. Find an Eulerian orientation Λ^+ of G^+.
4. Output $\Lambda := \Lambda^+ \setminus \Lambda^+(E^+)$.

By using the algorithm in Proposition 3 as a subroutine in Step 3, the total running time of Algorithm EulerO becomes $O(m)$. As will be discussed in Section 6.1, EulerO can solve certain restricted instances of MAX W-LIGHT, MIN W-HEAVY, MIN W-LIGHT, and MAX W-HEAVY exactly. For now, we just observe the following property:

Lemma 1 *In the orientation output by Algorithm* EulerO, *every vertex has outdegree at most* $\lceil \Delta/2 \rceil$, *where* Δ *is the maximum degree of* G.

Proof Let v be any vertex in G. The degree of v in G is at most Δ. By the construction of G^+, if the degree of v in G is even then its degree will be the same in G^+, but if the degree of v in G is odd then its degree in G^+ equals its degree in G plus one. Then, according to the definition of an Eulerian orientation, the outdegree of v in Λ^+ is at most $\lceil \Delta/2 \rceil$. Removing E^+ in Step 4 sometimes decreases but never increases the outdegree of v in the resulting $\Lambda(G)$. In other words, the outdegree of v in $\Lambda(G)$ is at most $\lceil \Delta/2 \rceil$. □

The next algorithm, PickUp, will be used to approximate MAX W-LIGHT.

Algorithm PickUp
Input: An undirected graph $G = (V, E)$
Output: An orientation Λ of G

1. Pick any $\min\{2W+1, n\}$ vertices in G. Let the set of chosen vertices be U.
2. Apply Algorithm EulerO to $G[U]$.
3. Orient every edge in $E \setminus E(U)$ that is incident to U towards U.
4. Orient all remaining edges arbitrarily, and output the resulting orientation Λ of G.

[2] If there is no vertex of odd degree in G then $G^+ = G$ and $E^+ = \emptyset$.

We have:

Theorem 1 *Algorithm* PickUp *is a linear-time* $(n/(2W+1))$*-approximation algorithm for* MAX W-LIGHT.

Proof The maximum degree Δ' in $G[U]$ is at most $2W$ because $|U| \le 2W+1$. Furthermore, if Δ' is odd then $\Delta'+1 \le 2W$. Thus, $W \ge \lceil \Delta'/2 \rceil$ holds. By Lemma 1, every vertex belonging to U has outdegree at most $\lceil \Delta'/2 \rceil$ in $\Lambda(G)$ and will therefore be W-light in $\Lambda(G)$. The number of W-light vertices in any optimal orientation of G is at most n, so the approximation ratio of PickUp is $n/(2W+1)$. The time complexity of PickUp is the same as that of EulerO, i.e., $O(m)$. □

4 Greedy algorithms for MIN W-LIGHT and MIN $(W+1)$-HEAVY

This section presents greedy algorithms for MIN W-LIGHT and MIN $(W+1)$-HEAVY with unbounded W. They both use the same framework, involving the theory of submodular functions, but adopt different criterion functions.

4.1 Greedy algorithm for MIN W-LIGHT

We first explain the main idea of the greedy algorithm for MIN W-LIGHT. The algorithm chooses vertices from G, one at a time, which are successively added to an initially empty set S. This process continues until G admits an orientation in which all remaining vertices (i.e., belonging to $V(G) \setminus S$) are $(W+1)$-heavy. The criterion for choosing which vertex to insert into S in each iteration is defined in terms of the solution to the following problem:

Problem ATTAINMENT OF $(W+1)$-HEAVY ORIENTATION ($P_1(G, W, S)$):

Output the value

$$\max_{\Lambda \in \mathcal{O}(G)} \sum_{v \in V \setminus S} \min\{W+1,\, d^+_\Lambda(v)\},$$

where $\mathcal{O}(G)$ is the set of all orientations of G.

As will be shown in Lemma 2, the problem $P_1(G, W, S)$ can be solved in polynomial time via the maximum flow problem. Motivated by this fact, define $g_1(S)$ to be the value of the solution to $P_1(G, W, S)$ plus $|S| \cdot (W+1)$. We call g_1 a *criterion function*. It is easy to see that $g_1(S) = g_1(V) = n(W+1)$ if and only if there exists an orientation of G in which every vertex in $V \setminus S$ is $(W+1)$-heavy. In addition, g_1 is a non-decreasing submodular function by Lemma 4 shown below. Assuming that Lemmas 2 and 4 hold, we can prove:

Theorem 2 MIN W-LIGHT *can be approximated within a ratio of* $\ln(W+1)+1$ *in* $O((mn+m^{1.5}\min\{m^{0.5}, \log m \log W\})n)$ *time.*

Proof It is known that optimization problems of the form $\min_{S\subseteq V}\{|S| : g(S) = g(V)\}$ can be approximated within a ratio of $H(\max_{i\in V}\{g(\{i\}) - g(\emptyset)\}) \leq \ln(\max_{i\in V}\{g(\{i\}) - g(\emptyset)\}) + 1$, where $H(i)$ is the ith Harmonic number, by the following greedy algorithm if g is a non-decreasing submodular function [29]:

1. Set $S := \emptyset$.
2. If $g(S) = g(V)$, then output S and halt.
3. Find an $i \in V \setminus S$ that maximizes $g(S \cup \{i\}) - g(S)$ and update $S := S \cup \{i\}$.
4. Goto Step 2.

Since $\ln(\max_{i\in V}\{g_1(\{i\}) - g_1(\emptyset)\}) \leq \ln(W+1)$, we obtain a $(\ln(W+1)+1)$-approximation algorithm by adopting g_1 as g in the above algorithm.

The algorithm executes at most n iterations of Step 3, each one running in $O(m^{1.5}\min\{m^{0.5}, \log m \log W\}) + O(mn)$ time, where the $O(m^{1.5}\min\{m^{0.5}, \log m \log W\})$-term is for computing the maximum flow for $g_1(S)$ according to Lemma 2, and the $O(mn)$-term is for finding an augmenting path to compute $g_1(S \cup \{i\})$ from $g_1(S)$ n times. $\square$

The rest of this section proves the lemmas needed to complete the proof of Theorem 2. First, we show the polynomial-time solvability of $P_1(G, W, S)$.

Lemma 2 ATTAINMENT OF $(W+1)$-HEAVY ORIENTATION*($P_1(G, W, S)$) can be solved in* $O(m^{1.5}\min\{m^{0.5}, \log m \log W\})$ *time.*

Proof The problem $P_1(G, W, S)$ can be reduced to the maximum flow problem as follows. Construct a flow network $\mathcal{N}_1(G, W, S)$ whose set of vertices is $\{s, t\} \cup E(G) \cup V(G)$ and whose set of arcs is $\{(s, e) : e \in E\} \cup \{(e, u), (e, v) : e = \{u, v\} \in E(G)\} \cup \{(u, t) : u \in V(G)\}$. The number n_1 of vertices in $\mathcal{N}_1(G, W, S)$ is $m + n + 2$, and the number m_1 of arcs is $3m + n$. The capacities of the arcs are defined by:

$$
\begin{aligned}
cap((s, e)) &= 1 \quad \text{for } e \in E(G),\\
cap((e, u)) &= 1 \quad \text{for } u \in e \in E(G),\\
cap((u, t)) &= \begin{cases} 0 & \text{for } u \in S,\\ W+1 & \text{for } u \in V(G) \setminus S. \end{cases}
\end{aligned}
$$

Refer to Fig. 1 (i) and (ii) for an example of the construction with $W = 2$. In Fig. 1 (ii), only the capacities of arcs of the form (u, t) are shown.

Now, we can see that the value of the solution to $P_1(G, W, S)$ corresponds to the maximum flow in the network $\mathcal{N}_1(G, W, S)$. For any $e(= \{u, v\}) \in E(G)$, if there is a flow of size one from s to e, then we may assume that will pass through exactly one of u and v by the integrality theorem. This is interpreted as follows:

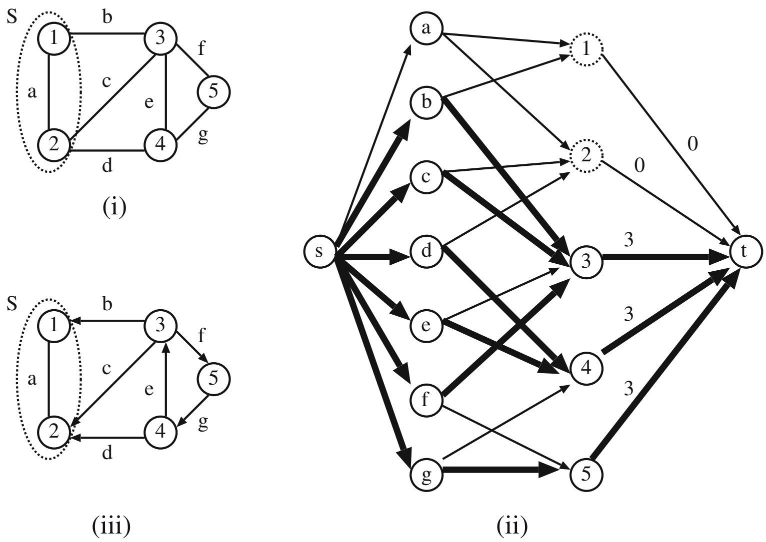

Fig. 1 (i) A graph G and a set $S = \{1, 2\}$; (ii) The constructed network $\mathcal{N}_1(G, 2, \{1, 2\})$ and a maximum flow; (iii) An orientation of G corresponding to the maximum flow, where the direction of the edge $a = \{1, 2\}$ can be determined arbitrarily

The edge $e = \{u, v\}$ in G is oriented as (u, v) if the unit of flow exiting e passes through u, and as (v, u) otherwise. The amount of flow passing through u is the minimum of $W + 1$ and the outdegree of u of the corresponding orientation. The optimal value of $P_1(G, W, S)$ is thus obtained by solving the maximum flow problem on $\mathcal{N}_1(G, W, S)$. As an example, a maximum flow of value 6 in the flow network in Fig. 1 (ii) is depicted by bold lines, and a corresponding orientation of G giving the solution to $P_1(G, W, S)$ is shown in Fig. 1 (iii).

The running time is analyzed as follows. Since $m_1 = O(n_1^{16/15-\varepsilon})$, the maximum flow algorithm from [23] can be applied to the flow network $\mathcal{N}_1(G, W, S)$. Its running time is $O(n_1 m_1)$, where n_1 and m_1 are the number of vertices and arcs in the input network. We have $n_1 = O(m)$ and $m_1 = O(m)$, so this takes $O(m^2)$ time. Alternatively, the maximum flow algorithm in [17] can be applied, which takes $O(\min\{n_1^{2/3}, m_1^{1/2}\} m_1 \log(n_1^2/m_1) \log U_1)$ time, where U_1 is the maximum capacity of an arc in the input network. Since the maximum capacity of an arc in $\mathcal{N}_1(G, W, S)$ is $O(W)$, this algorithm takes $O(m^{1.5} \log m \log W)$ time. To summarize, $P_1(G, W, S)$ can be solved in $O(m^{1.5} \min\{m^{0.5}, \log m \log W\})$ time. □

By the optimality of the maximum flow, there is a simple characterization of an optimal orientation. The next lemma is based on an observation about a certain type of paths: If there is a directed path from a $(W + 2)$-heavy vertex in $V \setminus S$ or a vertex in S to a W-light vertex in $V \setminus S$, then flipping the directions of the edges of the path gives a better orientation for $P_1(G, W, S)$.

Lemma 3 *Λ is an optimal orientation of $P_1(G, W, S)$ if and only if there is no directed path in $\Lambda(G)$ from a $(W+2)$-heavy vertex in $V \setminus S$ or a vertex in S to a W-light vertex in $V \setminus S$.*

Proof (If part) We prove the contrapositive, i.e., if Λ is not optimal, then there is a directed path from a $(W+2)$-heavy vertex in $V \setminus S$ or a vertex in S to a W-light vertex in $V \setminus S$. Consider an optimal orientation Λ^*. Since $\Lambda \neq \Lambda^*$, there is a directed path P starting from a vertex u and ending at a vertex v in $\Lambda(G)$, while the set of vertices of P forms a directed path in the opposite direction in $\Lambda^*(G)$. Note that P may be a single edge. For simplicity, assume that the only difference between Λ and Λ^* is a part of P. Since Λ^* is obtained by flipping the directions of the edges of P in Λ, $d^+_{\Lambda^*}(u) = d^+_\Lambda(u) - 1$ and $d^+_{\Lambda^*}(v) = d^+_\Lambda(v) + 1$ hold.

Since Λ is not optimal, i.e., $\sum_{v \in V \setminus S} \min\{W+1, d^+_\Lambda(v)\} < \sum_{v \in V \setminus S} \min\{W + 1, d^+_{\Lambda^*}(v)\}$, one of the following two conditions must hold: (i) $u, v \in V \setminus S$, $d^+_\Lambda(u) \geq W+2$ and $d^+_\Lambda(v) \leq W$; and (ii) $u \in S$, $v \in V \setminus S$, and $d^+_\Lambda(v) \leq W$. (If both of u and v are included in S, the value of $\sum_{v \in V \setminus S} \min\{W+1, d^+_\Lambda(v)\}$ does not change even if we flip the direction of P.) Thus, there is a directed path from a $(W+2)$-heavy vertex in $V \setminus S$ or a vertex in S to a W-light vertex in $V \setminus S$. The discussion is similar for the case where at least two paths differ between Λ and Λ^*.

(Only-if part) Again, we prove the contrapositive, that is, if there is a directed path from any $(W+2)$-heavy vertex in $V \setminus S$ or any vertex in S to a W-light vertex in $V \setminus S$, then Λ is not optimal. Let the start and end vertices of the path be u and v, and consider an orientation Λ' in which the directions of the edges of the path are flipped and the rest is the same as Λ. As in the discussion for the (If part) above, $d^+_{\Lambda'}(v) = d^+_\Lambda(v) + 1$ and $d^+_{\Lambda'}(u) = d^+_\Lambda(u) - 1$. Here, in case u is in $V \setminus S$ and is $(W+2)$-heavy, we have $\min\{W+1, d^+_\Lambda(u)\} = \min\{W+1, d^+_{\Lambda'}(u)\} = W+1$. In addition, since v is W-light in $\Lambda(G)$, the inequality $d^+_\Lambda(v) < d^+_{\Lambda'}(v) \leq W+1$ also holds. This implies that $\sum_{v \in V \setminus S} \min\{W+1, d^+_\Lambda(v)\} < \sum_{v \in V \setminus S} \min\{W+1, d^+_{\Lambda'}(v)\}$, i.e., Λ is not optimal. $\square$

Finally, we show that g_1 is a non-decreasing submodular function.

Lemma 4 *g_1 is a non-decreasing submodular function, that is, it satisfies (non-decreasingness) $g_1(S \cup \{i\}) - g_1(S) \geq 0$ for any $S \subseteq V$ and $i \in V \setminus S$, and (submodularity) $g_1(S) + g_1(T) \geq g_1(S \cap T) + g_1(S \cup T)$ for any $S, T \subseteq V$.*

Proof For any two disjoint subsets $S, S' \subseteq V$ of vertices, denote:

$$\alpha(S, S') = \min \left\{ \sum_{v \in S'} \min\{W+1, d^+_\Lambda(v)\} \ : \ \Lambda \in OptO(P_1(G, W, S)) \right\}, \qquad (1)$$

where $OptO(P_1(G, W, S))$ is the set of all optimal orientations of $P_1(G, W, S)$. We first show that the following equality holds for any disjoint $S, S' \subseteq V$:

$$g_1(S \cup S') - g_1(S) \ = \ |S'| \cdot (W+1) - \alpha(S, S'). \qquad (2)$$

Let $\Lambda_{S,S'}$ be an orientation of G that achieves $\alpha(S, S')$. By Lemma 3 and the optimality of $\Lambda_{S,S'}$ for $P_1(G, W, S)$, there is no directed path from any $(W + 2)$-heavy vertex in $V \setminus S$ or any vertex in S to a W-light vertex in $\Lambda_{S,S'}(G)$. Also, there exists no directed path from any vertex in S' to a W-light vertex in $V \setminus (S \cup S')$; otherwise it would contradict that $\Lambda_{S,S'}$ minimizes $\sum_{v\in S'} \min\{W + 1, d^+_\Lambda(v)\}$. These facts imply that $\Lambda_{S,S'}$ is also an optimal orientation for $P_1(G, W, S \cup S')$. Thus, we have:

$$\begin{aligned} g_1(S) &= |S| \cdot (W + 1) + \sum_{v\in V\setminus S} \min\{W + 1, d^+_{\Lambda_{S,S'}}(v)\} \\ &= (|S \cup S'| - |S'|) \cdot (W + 1) + \sum_{v\in V\setminus(S\cup S')} \min\{W + 1, d^+_{\Lambda_{S,S'}}(v)\} \\ &\quad + \sum_{v\in S'} \min\{W + 1, d^+_{\Lambda_{S,S'}}(v)\} \\ &= g_1(S \cup S') - |S'| \cdot (W + 1) + \sum_{v\in S'} \min\{W + 1, d^+_{\Lambda_{S,S'}}(v)\}, \end{aligned}$$

which proves (2). Note that the first equality above is based on the optimality of $\Lambda_{S,S'}$, while the second one is based on the fact that S and S' are disjoint. Then, by combining (1) and (2), $g_1(S \cup \{i\}) - g_1(S) = (W + 1) - \alpha(S, \{i\}) \geq 0$ holds for any $i \in V \setminus S$, which implies the non-decreasing property of g_1.

Next, we prove the submodularity of g_1. An equivalent condition is that:

$$g_1(S \cup \{i\}) - g_1(S) \;\geq\; g_1(S \cup \{i, j\}) - g_1(S \cup \{j\}) \tag{3}$$

for every $S \subseteq V$ and every $i, j \in V \setminus S$. By (2), we have:

$$\begin{aligned} g_1(S \cup \{i\}) - g_1(S) &= (W + 1) - \alpha(S, \{i\}), \\ g_1(S \cup \{j\}) - g_1(S) &= (W + 1) - \alpha(S, \{j\}), \\ g_1(S \cup \{i, j\}) - g_1(S) &= 2(W + 1) - \alpha(S, \{i, j\}). \end{aligned}$$

Combining these equations, we obtain the following:

$$\begin{aligned} &g_1(S \cup \{i\}) - g_1(S) + g_1(S \cup \{j\}) - g_1(S \cup \{i, j\}) \\ &= -\alpha(S, \{i\}) - \alpha(S, \{j\}) + \alpha(S, \{i, j\}). \end{aligned}$$

It is straightforward to see that $\alpha(S, \{i\}) + \alpha(S, \{j\}) \leq \alpha(S, \{i, j\})$ holds by the definition of α. This implies condition (3), i.e., the submodularity of g_1. □

Remark 1 The submodularity of g_1 can be also established by matroid theory. Such a proof is given in Appendix A. Although it is in some sense simpler, it has been placed in the appendix because the above proof and the proof of Lemma 7 in the next subsection are more closely related.

4.2 Greedy algorithm for MIN $(W+1)$-HEAVY

To obtain a greedy algorithm for MIN $(W+1)$-HEAVY, we follow the same strategy as for MIN W-LIGHT in Section 4.1. However, the proofs become slightly more complicated because here we need to rely on *minimum cost flows* instead of maximum flows. [3]

We first construct a criterion function g_2, analogous to g_1 for MIN W-LIGHT in Section 4.1. For this purpose, define the following problem:

Problem EXCESS OF W-LIGHT ORIENTATION ($P_2(G, W, S)$):

Output the value

$$\min_{\Lambda \in \mathcal{O}(G)} \sum_{v \in V \setminus S} \max\{0,\, d^+_\Lambda(v) - W\},$$

where $\mathcal{O}(G)$ is the set of all orientations of G.

Lemma 5 below describes how to solve $P_2(G, W, S)$ in polynomial time via the minimum cost flow problem. Let $h(S)$ be the value of the solution to $P_2(G, W, S)$ and define the criterion function $g_2(S) = h(\emptyset) - h(S)$. Then $g_2(V) = g_2(S) = h(\emptyset)$ if and only if $G \setminus S$ is W-orientable. Lemma 7 will show that g_2 is a non-decreasing submodular function, yielding the main result of this subsection:

Theorem 3 MIN $(W+1)$-HEAVY *can be approximated within a ratio of* $\ln(\Delta - W) + 1$ *in* $O(nm^3 \log n)$ *time.*

Proof By an argument similar to the one previously used for MIN W-LIGHT, the greedy algorithm in the proof of Theorem 2 with g set to g_2 finds a $(\ln(\max_{i \in V}\{g_2(\{i\}) - g_2(\emptyset)\}) + 1)$-approximate solution to the problem of finding a smallest possible $S \subseteq V$ that satisfies $g_2(S) = g_2(V)$, i.e., a smallest $S \subseteq V$ such that there exists an orientation of G in which every vertex in $V \setminus S$ is W-light. This is precisely the MIN $(W+1)$-HEAVY problem. Since $g_2(\emptyset) = 0$ and $g_2(\{i\}) \leq \Delta - W$, the approximation ratio is $\ln(\Delta - W) + 1$.

As for the running time, the greedy algorithm needs to solve the EXCESS OF W-LIGHT ORIENTATION problem at most n times. By Lemma 5, the running time of the greedy algorithm becomes $O(nm^3 \log n)$. $\square$

The rest of this subsection fills in the technical details.

[3] Recall that *the minimum cost flow problem* (see, e.g., [22]) takes as input a flow network with a specified capacity u_i and cost c_i for each arc a_i, and asks for a flow from the source to the sink of some specified size that has the minimum cost, where the cost is defined as $\sum_{a_i} c_i x_i$ and where x_i is the amount of the flow along the arc a_i.

Lemma 5 EXCESS OF W-LIGHT ORIENTATION *($P_2(G, W, S)$) can be solved in $O(m^3 \log n)$ time.*

Proof Reduce the problem $P_2(G, W, S)$ to the minimum cost flow problem as follows. Construct a flow network $\mathcal{N}_2(G, W, S)$ whose set of vertices is $\{s, t\} \cup E(G) \cup V(G)$ and whose set of arcs is $\{(s, e) : e \in E\} \cup \{(e, u), (e, v) : e = \{u, v\} \in E(G)\} \cup \{(u, t)_1, (u, t)_2 : u \in V(G)\}$. We assume that $d(u) \geq W + 1$ for every vertex u, since otherwise all edges incident to u can be oriented outward, and we can remove such edges in advance. The capacities of the arcs are defined by:

$$
\begin{aligned}
cap((s, e)) &= 1 \ \text{ for } e \in E(G),\\
cap((e, u)) &= 1 \ \text{ for } u \in e \in E(G),\\
cap((u, t)_1) &= \begin{cases} d(u) & \text{for } u \in S,\\ W & \text{for } u \in V(G) \setminus S, \end{cases}\\
cap((u, t)_2) &= \begin{cases} 0 & \text{for } u \in S,\\ d(u) - W & \text{for } u \in V(G) \setminus S. \end{cases}
\end{aligned}
$$

The costs of the arcs are defined by:

$$
\begin{aligned}
cost((s, e)) &= 0 \ \text{ for } e \in E(G),\\
cost((e, u)) &= 0 \ \text{ for } u \in e \in E(G),\\
cost((u, t)_1) &= 0 \ \text{ for } u \in V(G),\\
cost((u, t)_2) &= 1 \ \text{ for } u \in V(G).
\end{aligned}
$$

See Fig. 2 for an example of the construction with $W = 1$.

Then the value of the solution to $P_2(G, W, S)$ corresponds to the minimum cost flow in the network $\mathcal{N}_2(G, W, S)$ of size m from the source s to the sink t. To illustrate, in Fig. 2(ii), a minimum cost flow of cost 1 is indicated by bold lines. Fig. 2 (iii) shows a corresponding orientation of G that yields the solution to $P_2(G, W, S)$.

The minimum cost flow problem for an input flow network with n_1 vertices and m_1 arcs can be solved in $O(n_1^2 m_1 \log n_1)$ time [22]. Since $\mathcal{N}_2(G, W, S)$ has $n_1 = O(m)$ vertices and $m_1 = O(m)$ arcs and $m = O(n^2)$, the time complexity is $O(m^3 \log n)$. □

The optimality of the minimum cost flow problem can be characterized by the nonexistence of a negative cycle in its residual network [1], interpreted as in the following lemma:

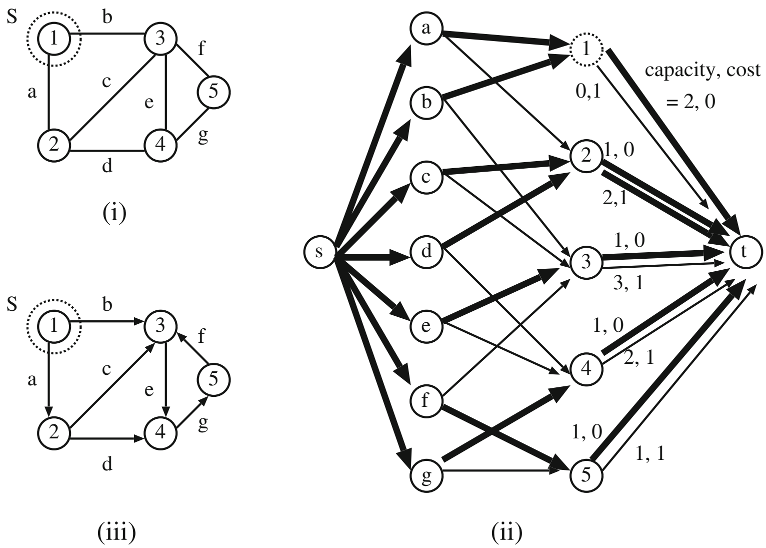

Fig. 2 (i) A graph G and a set $S = \{1\}$; (ii) The constructed network $\mathcal{N}_2(G, 1, \{1\})$ and a minimum cost flow; (iii) An orientation of G corresponding to the minimum cost flow

Lemma 6 *Λ is an optimal orientation of $P_2(G, W, S)$ if and only if there is no directed path in $\Lambda(G)$ from a $(W+1)$-heavy vertex in $V \setminus S$ to a $(W-1)$-light vertex in $V \setminus S$ or a vertex in S.*

Proof This proof is analogous to the proof of Lemma 3.
(If part) We prove the contrapositive, i.e., if Λ is not optimal, then there is a directed path from a $(W+1)$-heavy vertex in $V \setminus S$ to a $(W-1)$-light vertex in $V \setminus S$ or a vertex in S. Proceed exactly as in the first paragraph of the proof of Lemma 3. Here however, since Λ is not optimal, we have $\sum_{v \in V \setminus S} \max\{0, d^+_\Lambda(v) - W\} > \sum_{v \in V \setminus S} \max\{0, d^+_{\Lambda^*}(v) - W\}$, so that either one of the following two conditions must hold: (i) $u, v \in V \setminus S$, $d^+_\Lambda(u) \geq W+1$ and $d^+_\Lambda(v) \leq W-1$; and (ii) $u \in S$, $d^+_\Lambda(u) \geq W+1$, and $v \in V \setminus S$. This means that there is a directed path from a $(W+1)$-heavy vertex in $V \setminus S$ to a $(W-1)$-light vertex in $V \setminus S$ or a vertex in S. In the case where at least two paths differ between Λ and Λ^*, the discussion is similar.

(Only-if part) We prove the contrapositive, that is, if there is a directed path from a $(W+1)$-heavy vertex in $V \setminus S$ to a $(W-1)$-light vertex in $V \setminus S$ or a vertex in S, then Λ is not optimal. Just like before, let the start and end vertices of the path be u and v, and consider an orientation Λ' in which the directions of all edges on the path are flipped and the rest is the same as Λ. Then $d^+_{\Lambda'}(v) = d^+_\Lambda(v) + 1$ and $d^+_{\Lambda'}(u) = d^+_\Lambda(u) - 1$ still hold, and in case v is in $V \setminus S$ and $(W-1)$-light, we also have $\max\left\{0, d^+_\Lambda(v) - W\right\} = \max\left\{0, d^+_{\Lambda'}(v) - W\right\} = 0$. In addition, since u is $(W+1)$-heavy in $\Lambda(G)$, $d^+_\Lambda(u) > d^+_{\Lambda'}(u) \geq W$ holds. This implies that $\sum_{v \in V \setminus S} \max\left\{0, d^+_\Lambda(v) - W\right\} > \sum_{v \in V \setminus S} \max\left\{0, d^+_{\Lambda'}(v) - W\right\}$, i.e., Λ is not optimal. $\square$

Finally, we prove that g_2 is a non-decreasing submodular function by using the arguments from the proof of Lemma 4.

Lemma 7 *$g_2(S)$ satisfies: (non-decreasingness) $g_2(S \cup \{i\}) - g_2(S) \geq 0$ for any $S \subseteq V$ and $i \in V \setminus S$, and (submodularity) $g_2(S) + g_2(T) \geq g_2(S \cap T) + g_2(S \cup T)$ for any $S, T \subseteq V$.*

Proof For any two disjoint subsets $S, S' \subseteq V$, denote:

$$\beta(S, S') = \max \left\{ \sum_{v \in S'} \max\left\{0, d^+_\Lambda(v) - W\right\} \ : \ \Lambda \in OptO(P_2(G, W, S)) \right\}, \quad (4)$$

where $OptO(P_2(G, W, S))$ is the set of all optimal orientations of $P_2(G, W, S)$. We first show that for any disjoint $S, S' \subseteq V$:

$$g_2(S \cup S') - g_2(S) \ = \ \beta(S, S') \quad (5)$$

holds. Let $\Lambda'_{S,S'}$ be an orientation that achieves $\beta(S, S')$ (note that $\Lambda'_{S,S'}$ is different from $\Lambda_{S,S'}$ used in the proof of Lemma 4). By Lemma 6 and the optimality of $\Lambda'_{S,S'}$ for $P_2(G, W, S)$, there is no directed path from a W-heavy vertex in $V \setminus S$ to a $(W-1)$-light vertex in $V \setminus S$ or a vertex in S. Furthermore, $\Lambda'_{S,S'}$ maximizes $\sum_{v \in S'} \max\{0, d^+_\Lambda(v) - W\}$, so there is no directed path from a $(W+1)$-heavy vertex in $V \setminus (S \cup S')$ to a vertex in S'. We therefore see that $\Lambda'_{S,S'}$ is also an optimal orientation of $P_2(G, W, S \cup S')$ and:

$$\begin{aligned} g_2(S) \ &= \ g'(\emptyset) - \sum_{v \in V \setminus S} \max\{0, d^+_{\Lambda'_{S,S'}}(v) - W\} \\ &= \ g'(\emptyset) - \sum_{v \in V \setminus (S \cup S')} \max\{0, d^+_{\Lambda_{S,S'}}(v) - W\} \\ &\quad - \sum_{v \in S'} \max\{0, d^+_{\Lambda_{S,S'}}(v) - W\} \\ &= \ g_2(S \cup S') - \sum_{v \in S'} \max\{0, d^+_{\Lambda_{S,S'}}(v) - W\}, \end{aligned}$$

which is equivalent to (5). By (4) and (5), $g_2(S \cup \{i\}) - g_2(S) = \beta(S, \{i\}) \geq 0$ holds for any $i \in V \setminus S$, guaranteeing the non-decreasing property of g_2.

Finally, to prove the submodularity of g_2, use the following equivalent condition of the submodularity of g_2 (analogous to (3)):

$$g_2(S \cup \{i\}) - g_2(S) \ \geq \ g_2(S \cup \{i, j\}) - g_2(S \cup \{j\}) \quad (6)$$

for every $S \subseteq V$ and every $i, j \in V \setminus S$. It follows from (5) that:

$$\begin{aligned} g_2(S \cup \{i\}) - g_2(S) \ &= \ \beta(S, \{i\}), \\ g_2(S \cup \{j\}) - g_2(S) \ &= \ \beta(S, \{j\}), \\ g_2(S \cup \{i, j\}) - g_2(S) \ &= \ \beta(S, \{i, j\}), \end{aligned}$$

and hence:

$$\begin{aligned} & g_2(S \cup \{i\}) - g_2(S) + g_2(S \cup \{j\}) - g_2(S \cup \{i, j\}) \\ = \; & \beta(S, \{i\}) + \beta(S, \{j\}) - \beta(S, \{i, j\}). \end{aligned}$$

Since $\beta(S, \{i\}) \geq \beta(S, \{i, j\}) - \beta(S, \{j\})$ by the definition of β, this gives (6). We conclude that g_2 is submodular. □

5 Inapproximability results

Here, we present new inapproximability results for MAX W-LIGHT, MIN W-HEAVY, MIN W-LIGHT, and MAX W-HEAVY (in this order). The problems are treated in Sections 5.1, 5.2, 5.3, and 5.4, respectively. For more information about the MAXIMUM INDEPENDENT SET, MINIMUM VERTEX COVER, and SET COVER problems referred to below, see, e.g., [16].

5.1 Inapproximability of MAX W-LIGHT

Our first inapproximability result concerns MAX W-LIGHT:

Theorem 4 *For every fixed $W \geq 1$,* MAX W-LIGHT *cannot be approximated within a ratio of $(n/W)^{1-\varepsilon}$ for any constant $\varepsilon > 0$ in polynomial time, unless $\mathcal{P} = \mathcal{NP}$.*

Proof The result is obtained via a gap-preserving reduction from MAXIMUM INDEPENDENT SET, which is known to be hard to approximate [30].

Given any instance G of MAXIMUM INDEPENDENT SET, an instance H of MAX W-LIGHT is constructed in polynomial time as follows. (See Fig. 3 for an example of the construction with $W = 1$.) For each vertex $v_i \in V(G)$, we prepare a set of $2(2W + 1) = 4W + 2$ vertices $U_i = U_{i,1} \cup U_{i,2}$, where $U_{i,j} =$

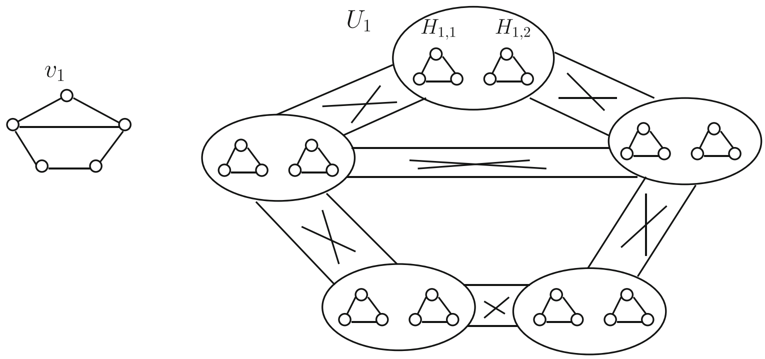

Fig. 3 Reduction from MAXIMUM INDEPENDENT SET to MAX 1-LIGHT. (Left) An input graph G; (Right) The constructed graph H.

$\{u_{i,j,1}, u_{i,j,2}, \ldots, u_{i,j,2W+1}\}$ for $j \in \{1, 2\}$, and let $V(H) = \bigcup_{v_i \in V(G)} U_i$. Thus, $|V(H)| = (4W + 2)n_G = \Theta(Wn_G)$, where $n_G = |V(G)|$. We define the edges of H so that each of the induced subgraphs $H[U_{i,1}]$ and $H[U_{i,2}]$, denoted by $H_{i,1}$ and $H_{i,2}$ respectively, forms a complete graph K_{2W+1}. (There are no edges between $U_{i,1}$ and $U_{i,2}$.) Then the number of edges in $H[U_i]$ is $W(4W + 2)$. For each edge $\{v_i, v_j\} \in E(G)$, we connect every pair of vertices $u_{i,k,x} \in U_i$ and $u_{j,l,y} \in U_j$ for $k, l \in \{1, 2\}$ and $1 \le x, y \le 2W + 1$ in H.

Let $OPT(G)$ and $OPT'(H)$ denote the value of an optimal solution for G of Maximum Independent Set and for H of Max W-Light, respectively. We need to prove that the above reduction is a gap-preserving one, i.e., that:

(i) If $OPT(G) \ge k$ then $OPT'(H) \ge (4W + 2)k$; and
(ii) If $OPT(G) < k/n_G^{1-\varepsilon}$ then $OPT'(H) < (4W + 2)k/n_G^{1-\varepsilon}$ for any positive constant ε.

Lemmas 8 and 9 below prove that the two conditions (i) and (ii) are indeed satisfied. Since $n_H = \Theta(W \cdot n_G)$, where $n_H = |V(H)|$, the approximation gap is $O((n_H/W)^{1-\varepsilon})$. The theorem then follows from the fact that Maximum Independent Set cannot be approximated within a ratio of $n_G^{1-\varepsilon}$ for any constant $\varepsilon > 0$ in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ [30]. □

Lemma 8 *If $OPT(G) \ge k$, then $OPT'(H) \ge (4W + 2)k$.*

Proof Without loss of generality, assume that there is an independent set $\{v_1, v_2, \ldots, v_k\}$ of size k in G. Consider a subset of vertices $U = \bigcup_{1 \le i \le k} U_i$ of H such that $|U| = (4W + 2)k$. The subgraph $H[U]$ contains $2k$ K_{2W+1}'s, $H_{1,1}, H_{1,2}, H_{2,1}, H_{2,2}, \ldots, H_{k,1}$, and $H_{k,2}$. Note that each $H_{i,j}$ $(1 \le i \le j$ and $j \in \{1, 2\})$ is not adjacent to the other K_{2W+1}'s, i.e., $H_{i,j'}$ $(j' \ne j)$ and $H_{i',j}$'s $(i' \ne i)$ in $H[U]$, because of the independence of $v_1, \ldots, v_k$. Construct an orientation of H as follows. First, orient the edges of K_{2W+1}'s in $H[U]$ in accordance with Proposition 2. Then, orient every edge $\{s, t\}$ as (s, t) for $s \in V(H) \setminus U$ and $t \in U$. The remaining edges in H, whose orientations are not decided yet, are oriented arbitrarily. In this orientation, all vertices in U have outdegree W in $H[U]$ by Proposition 2, so $OPT'(H) \ge (4W + 2)k$. □

Before proceeding to Lemma 9, we make an observation regarding the number of W-light vertices in two adjacent U_i's. We introduce the following concept: A *canonical orientation of H* is an orientation of H in which each U_i satisfies either of the following two conditions: (A) U_i is W-light; or (B) U_i is $(W + 1)$-heavy.[4]

Proposition 5 *If U_i and U_j are adjacent and the number of W-light vertices in U_i is at least as large as the number of W-light vertices in U_j, then at most $4W + 1$ vertices in $U_i \cup U_j$ are W-light in any non-canonical orientation.*

[4] It is not necessary for all U_i's to satisfy the same condition; i.e., it is possible that U_i is W-light while U_j is $(W + 1)$-heavy for some $i \ne j$.

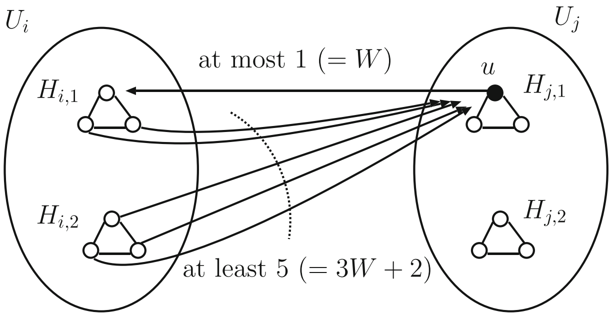

Fig. 4 Illustrating the proof of Proposition 5

Proof The subgraph $H_{i,1}$ ($H_{i,2}$, resp.) can be W-light only if all the edges $\{v, w\}$ connecting a vertex $v \in U_{i,1}$ with $w \notin U_{i,1}$ ($v \in U_{i,2}$ and $w \notin U_{i,2}$, resp.) are oriented as (w, v) by Proposition 4, because the density of $H_{i,1}$ ($H_{i,2}$, resp.) is W with $2W+1$ vertices and $W(2W+1)$ edges. Hence, if a vertex $u \in U_j$ is W-light, i.e., has at most W outgoing arcs, at least $2(2W+1) - W = 3W+2$ edges are oriented outwards from U_i, since the number of edges between u and U_i is $2(2W+1)$. For an example with $W = 1$, see Fig. 4.

As mentioned above, $H[U_i]$ has $4W+2$ vertices, $W(4W+2)$ edges, and at least $3W+2$ outgoing arcs. The density of U_i is thus at least $(W(4W+2)+3W+2)/(4W+2) > W$, so according to Proposition 4, we cannot make U_i W-light. Suppose we make one vertex, say, $u_{i,1,1} \in U_{i,1}$, $(W+1)$-heavy. Since $d(u_{i,1,1}) = 2W$ in $H[U_i]$ and only one of the $3W+2$ outgoing arcs is connected to $u_{i,1,1}$, the density of the rest of the vertices in U_i is at least $(W(4W+2)+3W+2-2W-1)/(4W+1) = (W(4W+2)+W+1)/(4W+1) > W$; still, it is greater than W and thus we cannot make them W-light. Then at least two vertices in U_i must be $(W+1)$-heavy, and hence the number of W-light vertices in $U_i \cup U_j$ is at most $4W+2-2+1 = 4W+1$.

We can generalize the above discussion to the case where the number of W-light vertices in U_j is larger than one. Assuming that there are k W-light vertices in U_j, and that (at least) $4W+2-k$ vertices in U_i are W-light for $1 \le k \le 2W+1$ leads to a contradiction as follows: The number of outgoing arcs from U_i is at least $k(3W+2)$. Moreover, k of the $k(3W+2)$ outgoing arcs are connected to one vertex in U_i. The average degree of the $4W+2-k$ vertices excluding k $(W+1)$-heavy vertices in U_i is:

$$\frac{W(4W+2)+k(3W+2)-k(2W+k)}{4W+2-k} = W + \frac{k(2W+2-k)}{4W+2-k} > W,$$

where the last inequality comes from the assumption that the number of W-light vertices in U_i is at least as large as that in U_j, i.e., $1 \le k \le 2W+1$. Therefore, if there are k W-light vertices in U_j, at least $k+1$ vertices in U_i must be $(W+1)$-heavy. Consequently, the total number of W-light vertices in $U_i \cup U_j$ is at most $4W+1$. □

Now we are ready to prove Lemma 9.

Lemma 9 *If $OPT(G) < k/n_G^{1-\varepsilon}$, then $OPT'(H) < (4W+2)k/n_G^{1-\varepsilon}$.*

Proof We prove the contrapositive, i.e., if $OPT'(H) \geq (4W+2)k/n_G^{1-\varepsilon}$ then $OPT(G) \geq k/n_G^{1-\varepsilon}$. First, observe that a non-canonical orientation is worse than a canonical one with respect to the number of W-light vertices from Proposition 5: Consider a pair of U_i and U_j that are adjacent. If we orient all the edges of $H_{i,1}$ and $H_{i,2}$ in $H[U_i]$ according to Proposition 2 and orient the edges between U_i and U_j towards U_i, then all the $4W+2$ vertices in U_i are W-light. Note that this orientation makes U_j $(W+1)$-heavy (more precisely, $(4W+2)$-heavy) whatever orientation inside $H[U_i]$ is selected because each vertex in U_j is adjacent to $4W+2$ vertices in U_i. Therefore, for two adjacent U_i and U_j, the way to maximize the number of W-light vertices is by orienting the edges so that U_i (or U_j) is W-light as described above, and U_j (or U_i) is $(W+1)$-heavy. This implies that only canonical orientations need to be considered as candidates of an optimal orientation.

Under a canonical orientation of H, if $(4W+2)k'$ vertices are W-light, then there exist k' W-light U_i's. One can see that this set of k' U_i's corresponds to an independent set as in the proof of Lemma 8. Thus, if $OPT'(H) \geq (4W+2)k/n_G^{1-\varepsilon}$ then $OPT(G) \geq k/n_G^{1-\varepsilon}$; equivalently, if $OPT(G) < k/n_G^{1-\varepsilon}$ then $OPT'(H) < (4W+2)k/n_G^{1-\varepsilon}$. □

5.2 Inapproximability of Min W-Heavy

We now show that Min W-Heavy is hard to approximate.

Theorem 5 *For every fixed $W \geq 1$, if* Min W-Heavy *can be approximated within a ratio of f in polynomial time, then so can* Minimum Vertex Cover.

Proof For the case $W = 1$, it was proved in [5] that Min W-Heavy and Minimum Vertex Cover are the same problem. We therefore assume $W \geq 2$ and describe a reduction from Minimum Vertex Cover to Min W-Heavy for any fixed $W \geq 2$.

Let G be any given instance of Minimum Vertex Cover (i.e., an undirected graph) with n vertices. Based on G, construct a graph H for Min W-Heavy in polynomial time as follows. (See Fig. 5 for an illustration.) Write $V(G) = \{v_1, v_2, \ldots, v_n\}$. The constructed graph H has n subgraphs denoted $H_1, H_2, \ldots, H_n$. Each subgraph H_i consists of one vertex $u_{i,0}$ and a complete graph K^i_{2W-1} of $2W-1$ vertices, named $u_{i,1}$ through $u_{i,2W-1}$. The vertex $u_{i,0}$ is connected to $W-1$ vertices $u_{i,1}$ through $u_{i,W-1}$. That is, the number of edges in H_i is $(2W-1)(2W-2)/2 + (W-1) = 2W(W-1)$. For every edge $\{v_i, v_j\}$ in G, the subgraphs H_i and H_j are connected by an edge $\{u_{i,0}, u_{j,0}\}$ in H.

Let $OPT(G)$ and $OPT'(H)$ denote the values of optimal solutions of Minimum Vertex Cover for G and Min W-Heavy for H. Lemma 10 below shows that $OPT(G) \leq k$ if and only if $OPT'(H) \leq k$, which then yields the theorem. □

Lemma 10 $OPT(G) \leq k$ *if and only if* $OPT'(H) \leq k$.

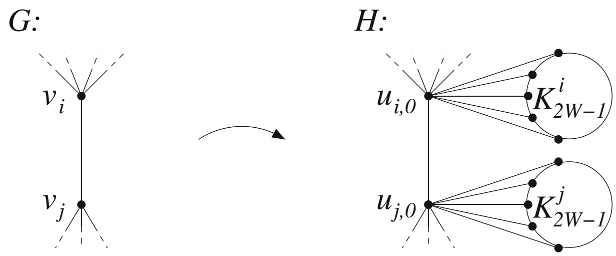

Fig. 5 Reduction from MINIMUM VERTEX COVER to MIN W-HEAVY

Proof (Only-if part) Suppose that $OPT(G) \le k$. Let $S \subseteq V(G)$ be a vertex cover of G with size at most k, and construct an orientation Λ of H as follows. For each $i = 1, 2, \ldots, n$, the internal edges of K^i_{2W-1} in the subgraph H_i are assigned any orientation under which every vertex has outdegree $W - 1$ by Proposition 2. The number of edges between $u_{i,0}$ and the K^i_{2W-1} in H_i is $W - 1$, and those edges are oriented from $u_{i,0}$ towards K^i_{2W-1}. So far, the outdegree of every vertex in H is exactly $W - 1$. Next, for any edge $\{u_{i,0}, u_{j,0}\}$ between two subgraphs H_i and H_j such that $v_i \in S$ and $v_j \in V(G) \setminus S$, orient it from $u_{i,0}$ to $u_{j,0}$; if both v_i and v_j are in S, then the edge $\{u_{i,0}, u_{j,0}\}$ is oriented arbitrarily. By the definition of a vertex cover, at least one end vertex of each edge $\{v_i, v_j\}$ belongs to S, so the above determines the direction of all edges in H. In this orientation Λ, the outdegree of any vertex $u_{i,0}$ in H corresponding to a vertex v_i in G that belongs to $V(G) \setminus S$ is $W - 1$. On the other hand, the outdegree of a vertex in H corresponding to a vertex in S is at least W. Therefore, the number of W-heavy vertices in $\Lambda(H)$ is at most k, i.e., $OPT'(H) \le k$.

(If part) Suppose $OPT'(H) \le k$. Let Λ be an orientation such that the number of W-heavy vertices in $\Lambda(H)$ is at most k. The following simple observation plays a key role in the proof. Suppose that $\{v_i, v_j\} \in E(G)$. Consider the subgraph $H[V(H_i) \cup V(H_j)]$ induced by $V(H_i)$ and $V(H_j)$, which includes the edge $\{u_{i,0}, u_{j,0}\}$. It consists of $|V(H_i)|+|V(H_j)| = 4W$ vertices and $|E(H_i)|+|E(H_j)|+1 = 4W(W-1)+1$ edges, so the density of $H[V(H_i) \cup V(H_j)]$ is strictly larger than $W - 1$. By Proposition 1, *at least one vertex in* $H[V(H_i) \cup V(H_j)]$ *must be* W*-heavy in* $\Lambda(H)$.

Let S be a set of vertices from G, where for each $i = 1, 2, \ldots, n$, if H_i contains a W-heavy vertex then the vertex v_i is included in S. As observed above, for each edge $\{v_i, v_j\}$ in $E(G)$, at least one vertex in the two subgraphs H_i and H_j must be W-heavy, so at least one endpoint of every edge in $E(G)$ belongs to S. Thus, S is a vertex cover of G. Finally, $OPT(G) \le |S| \le k$. □

It is known that MINIMUM VERTEX COVER cannot be approximated within a ratio of 1.3606 in polynomial time, unless $\mathcal{P} = \mathcal{NP}$ [10]. Theorem 5 thus implies:

Corollary 1 *For every fixed* $W \ge 1$, MIN W-HEAVY *cannot be approximated within a ratio of 1.3606 in polynomial time, unless* $\mathcal{P} = \mathcal{NP}$.

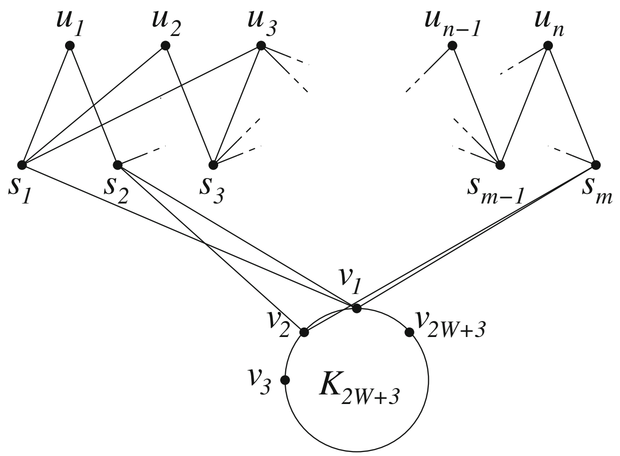

Fig. 6 Reduction from $(W + 1)$-SET COVER to MIN W-LIGHT. For clarity, edges of the form $\{u_i, v_h\}$ are not shown

5.3 Inapproximability of MIN W-LIGHT

Let B-SET COVER be the SET COVER problem restricted to instances where every subset contains at most B elements [7, 16, 27]. For MIN W-LIGHT, we derive the following inapproximability result.

Theorem 6 *For every fixed $W \geq 1$, if* MIN W-LIGHT *can be approximated within a ratio of f in polynomial time, then so can* $(W + 1)$-SET COVER.

Proof We reduce $(W + 1)$-SET COVER to MIN W-LIGHT.

Given any instance I of the decision version of $(W + 1)$-SET COVER, consisting of a universe set $U = \{1, 2, \dots, n\}$, a family $\mathcal{F}$ of subsets $S_1, S_2, \dots, S_m \subseteq U$ each of size at most $W+1$, and a positive integer k (and where the objective is to determine whether there exists a subfamily $\mathcal{F}'$ of $\mathcal{F}$ with $|\mathcal{F}'| \leq k$ such that $\bigcup_{S_i \in \mathcal{F}'} S_i = U$), construct an instance G of the decision version of MIN W-LIGHT, asking if G can be oriented so that at most k vertices become W-light, in polynomial time as follows. (See Fig. 6 for an illustration.) Create two sets of vertices $\{u_1, u_2, \dots, u_n\}$ and $\{s_1, s_2, \dots, s_m\}$, the former corresponding to U and the latter to $\mathcal{F}$, and a complete graph K_{2W+3} whose vertices are named $v_1, v_2, \dots, v_{2W+3}$. To connect these vertices to each other, insert an edge $\{u_i, s_j\}$ for every $i \in U$ and every j with $i \in S_j$. Also, for every $i \in \{1, 2, \dots, n\}$, insert W edges $\{u_i, v_h\}$ where $h \in \{1, 2, \dots, W\}$. Finally, for every $j \in \{1, 2, \dots, m\}$, if $|S_j| \leq W$ then insert all edges of the form $\{s_j, v_h\}$ with $h \in \{1, \dots, W + 1 - |S_j|\}$. Let G be the resulting graph. Note that every vertex s_j has degree $W + 1$.

The correctness of the theorem then follows by Lemma 11 below. □

Lemma 11 *There exists an orientation of G with at most k W-light vertices if and only if the answer to the instance I of* $(W + 1)$-SET COVER *is yes.*

Proof (Only-if part) Let Λ be an orientation of G in which the set of W-light vertices is L and where $|L| \leq k$. We can assume that L does not contain any vertex from K_{2W+3} (otherwise, we can obtain an orientation where all vertices in K_{2W+3} have outdegree at least $W+1$ by replacing the orientations of the internal edges in K_{2W+3} with the one in Proposition 2). Without loss of generality, also assume that every edge connected to a vertex in K_{2W+3} from the outside is oriented towards K_{2W+3} in Λ.

Now suppose that $\{u_1, u_2, \ldots, u_n\} \cap L \neq \emptyset$. Then there exists some $u_i \in \{u_1, u_2, \ldots, u_n\} \cap L$. Since all W edges between this u_i and K_{2W+3} are oriented towards K_{2W+3} and the outdegree of u_i is W, every edge between u_i and a vertex in $\{s_1, s_2, \ldots, s_m\}$ must be oriented towards u_i in Λ. This means that by flipping the direction of one edge for each such u_i, we can force all vertices in $\{u_1, u_2, \ldots, u_n\}$ to become $(W+1)$-heavy. We therefore assume that $\{u_1, u_2, \ldots, u_n\} \cap L = \emptyset$ and $L \subseteq \{s_1, s_2, \ldots, s_m\}$. One can see that, for every u_i in $\{u_1, u_2, \ldots, u_n\}$, there is at least one edge of the form $\{u_i, s_j\}$ oriented as (u_i, s_j) in Λ, which implies that s_j cannot be $(W+1)$-heavy and hence s_j must belong to L; this corresponds the element $i \in U$ being covered by the subset S_j from $\mathcal{F}$. In summary, L induces a set cover of size at most k for the original $(W+1)$-SET COVER instance I.

(If part) Suppose that the answer to the instance I is yes. Let $\mathcal{F}'$ be a subfamily of $\mathcal{F}$ such that $|\mathcal{F}'| \leq k$ and $\bigcup_{S_i \in \mathcal{F}'} S_i = U$. From $\mathcal{F}'$, construct an orientation Λ of G as follows. First, orient the internal edges of K_{2W+3} as in Proposition 2 so that the outdegree of every vertex in K_{2W+3} is $W+1$. This ensures that $V(K_{2W+3})$ will be $(W+1)$-heavy. Then, for each $i \in \{1, 2, \ldots, n\}$, orient all W edges between u_i and K_{2W+3} away from u_i. Also, for each $j \in \{1, 2, \ldots, m\}$, orient all $W+1-|S_j|$ edges between s_j and K_{2W+3} away from s_j. All remaining edges are between the u_i- and s_j-vertices. For each such u_i, select any one $S_j \in \mathcal{F}'$ such that $i \in S_j$ (there must exist at least one because $\bigcup_{S_i \in \mathcal{F}'} S_i = U$) and orient the edge $\{u_i, s_j\}$ as (u_i, s_j). Finally, any unoriented edge $\{u_i, s_j\}$ is oriented as (s_j, u_i).

Under Λ, the outdegree of every vertex in K_{2W+3}, in $\{u_1, u_2, \ldots, u_n\}$, and in $\{s_j : S_j \notin \mathcal{F}'\}$ is $W+1$. The outdegree of any vertex in $\{s_j : S_j \in \mathcal{F}'\}$ is at most W. Thus, the number of W-light vertices is at most k in $\Lambda(G)$. □

For every fixed $B \geq 3$, B-SET COVER is $\mathcal{NP}$-hard to approximate [7, 27]. Plugging in the known inapproximability bounds for 3-SET COVER (100/99, from [7]), for 4-SET COVER (53/52, also from [7]), and for B-SET COVER ($\ln B - O(\log\log B)$, from [27]) into Theorem 6 yields:

Corollary 2 MIN 2-LIGHT *and* MIN 3-LIGHT *cannot be approximated within a ratio of* 100/99 *and* 53/52*, respectively, in polynomial time, unless* $\mathcal{P} = \mathcal{NP}$*. Furthermore, for sufficiently large values of* W*,* MIN W-LIGHT *cannot be approximated within a ratio of* $\ln(W+1) - O(\log\log W)$ *in polynomial time, unless* $\mathcal{P} = \mathcal{NP}$*.*

5.4 Inapproximability of MAX W-HEAVY

Lastly, we consider the polynomial-time inapproximability of MAX W-HEAVY.

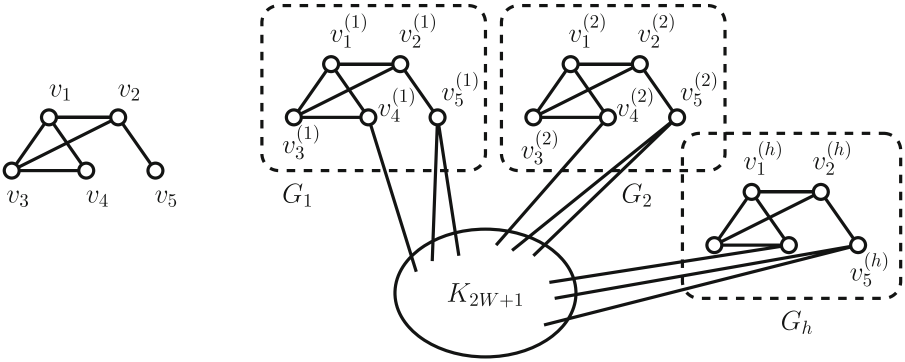

Fig. 7 Reduction from MAXIMUM INDEPENDENT SET to MAX W-HEAVY. The given graph G is on the left and the constructed graph G' is on the right

Theorem 7 *For every fixed $W = \Omega(n^{1/3})$,* MAX W-HEAVY *cannot be approximated within a ratio of $(n/W)^{1/2-\varepsilon}$ for any constant $\varepsilon > 0$ in polynomial time, unless $\mathcal{P} = \mathcal{NP}$.*

Proof We give a reduction from MAXIMUM INDEPENDENT SET (different from the one in Section 5.1) to MAX W-HEAVY.

Let $G = (V, E)$ be any given instance of MAXIMUM INDEPENDENT SET such that $\Delta \le W$. To guarantee that this condition $\Delta \le W$ always holds, we assume that $W \ge |V|$ in the following. Construct an instance $G' = (V', E')$ of MAX W-HEAVY as follows. (See Fig. 7 for a sketch of the construction.) For some positive integer h, prepare h copies $G_i = (V_i, E_i)$ of the graph G, where $i = 1, 2, \dots, h$. The value of h will be determined later. For each vertex $v \in V$, denote the copy of v in V_i by $v^{(i)}$. Also, create a complete graph K_{2W+1} whose vertices are named $\{u_1, u_2, \dots, u_{2W+1}\}$. Let the set V' of vertices in G' be $\{u_1, \dots, u_{2W+1}\} \cup \bigcup_{i=1}^{h} V_i$. Thus, the total number of vertices in V' is $|V'| = h \cdot |V| + 2W + 1$. Next, for each $i \in \{1, \dots, h\}$ and $v \in V$, define $E^{(i,v)}$ to be a set of $W - d(v)$ edges between $v^{(i)}$ and u_j for $j = 1, 2, \dots, W - d(v)$. Let the edge set $E(G')$ be the union of $\bigcup_{1 \le i \le h} E_i$, $E(K_{2W+1})$, and $\bigcup_{1 \le i \le h, v \in V} E^{(i,v)}$. Thus, every vertex in $G_1, G_2, \dots, G_h$ has degree W. If h is bounded by a polynomial in the size of G then the time needed to construct G' is also polynomially bounded.

Denote the value of an optimal solution of MAXIMUM INDEPENDENT SET for G by $OPT(G)$, and the value of an optimal solution of MAX W-HEAVY for G' by $OPT'(G')$. Let $n = |V(G)|$. It is known that MAXIMUM INDEPENDENT SET is $\mathcal{NP}$-hard to approximate within a ratio of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [30]. From Lemma 12 below, we obtain:

(i) If $OPT(G) \ge k$ then $OPT'(G') \ge hk + 2W + 1$; and
(ii) If $OPT(G) < k/n^{1-\varepsilon}$ then $OPT'(G') < hk/n^{1-\varepsilon} + 2W + 1$ for any positive constant ε.

Note that in (ii),

$$OPT'(G') < \frac{hk}{n^{1-\varepsilon}} + 2W + 1 = \frac{hk + 2W + 1}{n^{1-\varepsilon}} \left(1 + \frac{(n^{1-\varepsilon} - 1)(2W + 1)}{hk + 2W + 1}\right).$$

We select $h = Wn$, which means that

$$\frac{(n^{1-\varepsilon} - 1)(2W + 1)}{hk + 2W + 1} < \frac{3Wn^{1-\varepsilon}}{Wnk} < 3.$$

Hence $OPT'(G') < (hk+2W+1)/(n^{1-\varepsilon}/4)$. Since the number $|V'|$ of vertices in G' is $hn+2W+1 = Wn^2+2W+1$, we have $n^{1-\varepsilon}/4 = (|V'|/W)^{1/2-\varepsilon'}$ by an appropriate choice of ε', that is, we obtain the inapproximability bound $(|V'|/W)^{1/2-\varepsilon'}$. □

Based on the above theorem, we have the following corollary which gives an almost tight inapproximability bound since a $(W + 1)$-approximation algorithm is known [5].

Corollary 3 *For $W = \Theta(n^{1/3})$,* Max W-Heavy *cannot be approximated within a ratio of $W^{1-\varepsilon}$ for any constant $\varepsilon > 0$ in polynomial time, unless $\mathcal{P} = \mathcal{NP}$.*

Proof By taking $W = cn$ for some constant $c \geq 1$ in the reduction in the proof of Theorem 7, the number $|V'|$ of the vertices in G' becomes $cn^3 + 2cn + 1$, by which $W = \Theta(|V'|^{1/3})$. Then, the approximation gap becomes $n^{1-\varepsilon}/4 = W^{1-\varepsilon'}$ by choosing an appropriate ε'. □

Lemma 12 *$OPT(G) \geq k$ if and only if $OPT'(G') \geq hk + 2W + 1$.*

Proof (Only-if part) Suppose G has an independent set $I \subseteq V$ of size at least k. Construct an orientation Λ of G' by orienting the internal edges of K_{2W+1} according to Proposition 2 so that all vertices in K_{2W+1} have outdegree W. Furthermore, for $i = 1, 2, \ldots, h$, any internal edge $\{v^{(i)}, v'^{(i)}\}$ in G_i such that $v \in I$ and $v' \notin I$ is oriented from $v^{(i)}$ to $v'^{(i)}$ (this is always possible due to the independence of I), and all other internal edges in G_i are oriented arbitrarily. Finally, every edge between G_i (for $i = 1, 2, \ldots, h$) and K_{2W+1} is oriented outwards from G_i.

Now, if $v \in I$ then $v^{(i)}$ for all $i = 1, 2, \ldots, h$ have outdegree W in $\Lambda(G')$. Every vertex of K_{2W+1} has outdegree W, so the total number of W-heavy vertices in G' under Λ is at least $|I| \cdot h + (2W + 1) \geq hk + (2W + 1)$, which gives $OPT(G') \geq hk + (2W + 1)$.

(If part) Let Λ be an orientation of G' in which the set S of W-heavy vertices is of size at least $hk + 2W + 1$. We may assume that $V(K_{2W+1}) \subseteq S$, because otherwise we can replace the orientation of the internal edges in K_{2W+1} with the one in Proposition 2 without decreasing the number of W-heavy vertices. By this assumption, $|S \setminus V(K_{2W+1})| \geq hk$ holds, and thus at least one G_i contains at least k W-heavy vertices, i.e., $|V_i \cap S| \geq k$. By the construction of G', each vertex in V_i is incident to exactly W edges, so all edges incident to a vertex v belonging to $V_i \cap S$ must be oriented away from v in Λ. This implies that no two vertices in $V_i \cap S$ are adjacent, so $V_i \cap S$ forms an independent set in G_i. The set of vertices in G corresponding to $V_i \cap S$ is also an independent set, and we have $OPT(G) \geq |V_i \cap S| \geq k$. □

Remark 2 The proof of Theorem 7 depends on the hardness of MAXIMUM INDEPENDENT SET. A crucial condition in the proof is that W is at least Δ, the maximum degree in the given instance G of MAXIMUM INDEPENDENT SET. Since MAXIMUM INDEPENDENT SET is $\mathcal{NP}$-hard when $\Delta \geq 3$, the above reduction also gives an alternative proof that MAX W-HEAVY is $\mathcal{NP}$-hard for any fixed $W \geq 3$, different from the one in [5]. However, for the case $W = 2$, the reduction does not provide any new insights and the computational complexity of MAX 2-HEAVY is still unknown.

6 Degree-bounded graphs

In this section, we obtain several small positive results for graphs whose degrees satisfy certain bounds. Corollaries 8 and 9 below are a first step towards determining the unknown computational complexity of MIN 1-LIGHT and its supplementary problem MAX 2-HEAVY.

6.1 Exact solutions in special cases

This subsection investigates under what circumstances (in terms of W and given bounds on the input graph's degree) some known graph orientation algorithms solve MAX W-LIGHT, MIN W-HEAVY, MIN W-LIGHT, and MAX W-HEAVY exactly.

Lemma 1 in Section 3 above showed that Algorithm EulerO (described in Section 3) outputs an orientation in which every vertex has outdegree at most $\lceil \Delta/2 \rceil$, where Δ is the maximum degree of G. A simple corollary is:

Corollary 4 *If the input graph G satisfies $W \geq \lceil \Delta/2 \rceil$ then Algorithm* EulerO *finds an optimal orientation for* MAX W-LIGHT *and* MIN $(W + 1)$-HEAVY *in which all vertices are W-light.*

A polynomial-time algorithm named Reverse [6] computes an orientation of G that minimizes the maximum outdegree of the input graph (or, equivalently, minimizes the maximum indegree [28]).[5] Write $\gamma = \min_\Lambda \max_v d^+_\Lambda(v)$. Then every vertex has outdegree at most γ in the orientation output by Algorithm Reverse, and if $\gamma \leq W$ then all vertices are W-light. Since Algorithm EulerO outputs an orientation in which the maximum outdegree is at most $\lceil \Delta/2 \rceil$, it always holds that $\gamma \leq \lceil \Delta/2 \rceil$. It follows that:

[5] Algorithm Reverse works as follows: Start with any orientation Λ of G. Select a vertex v_0 with the highest outdegree in $\Lambda(G)$, find a directed path $P = (v_0, v_1, \ldots, v_k)$ satisfying $d^+_\Lambda(v_i) \leq d^+_\Lambda(v_0)$ for $1 \leq i \leq k-1$ and $d^+_\Lambda(v_k) \leq d^+_\Lambda(v_0) - 2$, flip the direction of every edge along P, and repeat the above process (select a v_0, etc.) until no such path P exists. Output Λ.

Corollary 5 *If the input graph G satisfies $W \geq \lceil \Delta/2 \rceil$ then Algorithm* Reverse *finds an optimal orientation for* MAX W-LIGHT *and* MIN $(W+1)$-HEAVY.

Next, using arguments analogous to those in the proof of Lemma 1, we deduce the following lower bound for EulerO:

Lemma 13 *In the orientation output by Algorithm* EulerO*, every vertex has outdegree at least $\lfloor \delta/2 \rfloor$, where δ is the minimum degree of G.*

Proof By the same reasoning as in the proof of Lemma 1, the outdegree of any vertex v in $\Lambda^+(G^+)$ is at least $\lceil \delta/2 \rceil$. Now, however, we note that removing E^+ in Step 4 decreases the outdegree of v in the resulting $\Lambda(G)$ by one in case the degree of v in G is odd, and has no effect in case the degree of v in G is even. Thus, the outdegree of v in $\Lambda(G)$ is at least $\lfloor \delta/2 \rfloor$. □

Corollary 6 *If the input graph G satisfies $W+1 \leq \lfloor \delta/2 \rfloor$ then Algorithm* EulerO *finds an optimal orientation for* MIN W-LIGHT *and* MAX $(W+1)$-HEAVY *in which all vertices are $(W+1)$-heavy.*

Define $\psi = \max_\Lambda \min_v d^+_\Lambda(v)$. Algorithm EulerO outputs an orientation in which the minimum outdegree is at least $\lfloor \delta/2 \rfloor$, so $\psi \geq \lfloor \delta/2 \rfloor$ always holds. The existence of such an orientation has previously been discussed in [18]; see also [25, Theorem 61.1] and [13, Theorem 2.3.5]. These are essentially the same theorem, with the last one containing a constructive proof showing how to find an optimal orientation in polynomial time. An algorithm named Exact-1-MaxMinO, based on similar ideas but using a maximum-flow technique, was presented in [2] by the authors of the current paper. In summary, Exact-1-MaxMinO runs in polynomial time and outputs an orientation in which every vertex has outdegree at least ψ. Similar to the above, this gives the following corollary:

Corollary 7 *If the input graph G satisfies $W+1 \leq \lfloor \delta/2 \rfloor$ then Algorithm* Exact-1-MaxMinO *finds an optimal orientation for* MIN W-LIGHT *and* MAX $(W+1)$-HEAVY.

Also note that Corollary 6 directly implies the following:

Corollary 8 MIN 1-LIGHT *and* MAX 2-HEAVY *can be solved in linear time for $\delta \geq 4$.*

6.2 A polynomial-time $\lfloor \Delta/2 \rfloor$-approximation algorithm for MAX 2-HEAVY

The goal of this subsection is to develop a polynomial-time approximation algorithm for MAX 2-HEAVY whose approximation ratio depends on the maximum degree Δ of the input graph, and to determine if it yields an exact algorithm for MAX 2-HEAVY when Δ is small.

Before presenting the algorithm, we need some additional tools. The *line graph* $L(G)$ (see, e.g., [25]) of an undirected graph $G = (V, E)$ is defined as $L(G) = (E, E')$, where $\{e_1, e_2\} \in E'$ if and only if $e_1 \cap e_2 \neq \emptyset$ for any $e_1, e_2 \in E$. Note that each vertex v in G corresponds to a clique of size $d(v)$ in $L(G)$. We introduce two procedures that convert a given orientation of G to a matching in $L(G)$, and vice versa.

Procedure MtoO

Input: An undirected graph $G = (V, E)$ and a matching M in $L(G)$
Output: An orientation Λ_M of G

1. For each edge $\{\{u, v\}, \{v, w\}\} \in M$, orient the two corresponding edges in G as (v, u) and (v, w).
2. Orient all remaining edges arbitrarily.
3. Output the resulting orientation of G.

Procedure OtoM

Input: An undirected graph $G = (V, E)$ and an orientation Λ of G
Output: A matching M_Λ in $L(G)$

1. Let $U := \{v \in V : d^+_\Lambda(v) \geq 2\}$ and $M_\Lambda := \emptyset$.
2. Select any vertex $v \in U$ and denote edges oriented outwards from v in Λ by $(v, u_1), \ldots, (v, u_d)$, where $d = d^+_\Lambda(v)$.
3. For $i := 1, \ldots, \lfloor d/2 \rfloor$, insert the edge $\{\{v, u_{2i-1}\}, \{v, u_{2i}\}\}$ of $L(G)$ into M_Λ.
4. Let $U := U \setminus \{v\}$. If $U \neq \emptyset$, goto Step 2; otherwise, output M_Λ and halt.

The next two lemmas relate the size of a matching in the line graph to the number of 2-heavy vertices:

Lemma 14 *Let M be any matching in $L(G)$ and let Λ_M be the orientation output by Procedure* MtoO*. The number of 2-heavy vertices in $\Lambda_M(G)$ is at least $|M|/\lfloor \Delta/2 \rfloor$.*

Proof For any vertex v in G, let C_v be its corresponding clique in $L(G)$. Procedure MtoO makes v a 2-heavy vertex in Step 1 if and only if M contains one or more edges belonging to C_v. By definition, the number of vertices in C_v is $d(v) \leq \Delta$, so M contains at most $\lfloor \Delta/2 \rfloor$ edges from C_v. This means that M must contain an edge from at least $|M|/\lfloor \Delta/2 \rfloor$ different cliques in $L(G)$, so the number of 2-heavy vertices in $\Lambda_M(G)$ is at least $|M|/\lfloor \Delta/2 \rfloor$. □

Lemma 15 *Let* Λ *be any orientation of* G *and let* M_Λ *be the matching in* $L(G)$ *output by Procedure* OtoM. *The number of 2-heavy vertices in* $\Lambda(G)$ *is at most* $|M_\Lambda|$.

Proof Let U be the set of 2-heavy vertices in $\Lambda(G)$. Consider any $v \in U$. After it is selected in Step 2 of OtoM, Step 3 inserts at least one edge $\{\{v, x\}, \{v, y\}\}$ into M_Λ since $d^+_\Lambda(v) \geq 2$. Furthermore, the same edge $\{\{v, x\}, \{v, y\}\}$ will never be inserted again for some $x \in U$ or $y \in U$. Therefore, $|U| \leq |M_\Lambda|$. □

We now describe the approximation algorithm for MAX 2-HEAVY:

Algorithm Max-2-H
Input: An undirected graph $G = (V, E)$
Output: An orientation Λ of G

1. Construct the line graph $L(G)$ of G.
2. Compute a maximum matching M^* in $L(G)$.
3. Apply Procedure MtoO to (G, M^*), and output the obtained orientation.

Theorem 8 *Algorithm* Max-2-H *is a polynomial-time* $\lfloor\Delta/2\rfloor$*-approximation algorithm for* MAX 2-HEAVY.

Proof To analyze the approximation ratio, let Λ^* be an optimal orientation of G for MAX 2-HEAVY, let $OPT(G)$ be the number of 2-heavy vertices in $\Lambda^*(G)$, and let $ALG(G)$ be the number of 2-heavy vertices in the solution returned by Max-2-H. Also, let M_{Λ^*} denote the matching in $L(G)$ obtained by applying Procedure OtoM to (G, Λ^*). By Lemma 14, we know that $ALG(G) \geq |M^*|/\lfloor\Delta/2\rfloor$. Moreover, $OPT(G) \leq |M_{\Lambda^*}|$ according to Lemma 15. Since the matching M^* computed in Step 2 of Algorithm Max-2-H is a maximum matching, $|M_{\Lambda^*}| \leq |M^*|$ always holds. By combining the three inequalities, we get:

$$OPT(G) \leq |M_{\Lambda^*}| \leq |M^*| \leq \lfloor\Delta/2\rfloor \cdot ALG(G).$$

We conclude that the approximation ratio of Max-2-H is $\lfloor\Delta/2\rfloor$.

In Step 1, the line graph can be constructed in polynomial time, and a maximum matching can be found in polynomial time (see, e.g., [1]) in Step 2. Step 3 also takes polynomial time, so the total running time of Max-2-H is clearly polynomial. □

The following example shows that the approximation ratio $\lfloor\Delta/2\rfloor$ for Algorithm Max-2-H stated in Theorem 8 is tight. (See Fig. 8 for an illustration of the case $\Delta = 4$.)

First, suppose that Δ is even. Consider the graph $G = (V, E)$ with $V = \{v_1, v_2, u_1, \ldots, u_\Delta\}$ and $E = \{\{v_1, u_i\}, \{v_2, u_i\} : 1 \leq i \leq \Delta\}$. The line graph $L(G)$

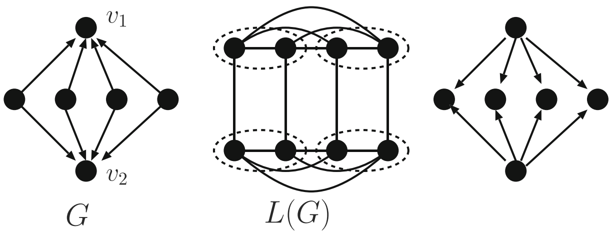

Fig. 8 A tight example for Max-2-H when $\Delta = 4$. (Left) A graph G and an optimal orientation of G; (Middle) The line graph $L(G)$ and one possible maximum matching M^* in $L(G)$; (Right) The non-optimal orientation of G output by Max-2-H if this M^* is used

consists of two complete graphs of size Δ, corresponding to v_1 and v_2 in G, with Δ disjoint edges between them, corresponding to paths of length two of the form (v_1, u_i, v_2) in G. Step 2 of Algorithm Max-2-H may select a matching M^* in $L(G)$ containing $\Delta/2$ edges from each of the two complete graphs (each vertex of $L(G)$ is included in some edge in M^*, so M^* is a matching of maximum size). In Step 3, Max-2-H makes the two vertices v_1 and v_2 2-heavy (in fact, Δ-heavy) in $\Lambda(G)$. No other vertices are 2-heavy in $\Lambda(G)$. In contrast, an optimal solution has Δ vertices that are 2-heavy, namely $\{u_1, \ldots, u_\Delta\}$, as can be seen by orienting the edges in E as $\{(u_i, v_1), (u_i, v_2) : 1 \le i \le \Delta\}$. Here, the optimal number of 2-heavy vertices is $\Delta/2$ times greater than the number of 2-heavy vertices obtained by Max-2-H.

Next, suppose that Δ is odd. Use the same example as for even Δ, but remove the vertex u_Δ from V and replace the two edges $\{v_1, u_\Delta\}$ and $\{v_2, u_\Delta\}$ in E by a single edge $\{v_1, v_2\}$. After this modification, the maximum degree in G is still Δ, and a matching in $L(G)$ consisting of $(\Delta - 1)/2$ edges from each of the two complete graphs is a maximum matching which yields two 2-heavy vertices. The number of 2-heavy vertices in an optimal solution is $\Delta - 1$, so the approximation ratio is $(\Delta - 1)/2$ when Δ is odd.

From Theorem 8, we also have the following corollary, which reveals another side of the computational complexity of MIN 1-LIGHT and MAX 2-HEAVY:

Corollary 9 MIN 1-LIGHT *and* MAX 2-HEAVY *can be solved in polynomial time for* $\Delta \le 3$.

6.3 A linear-time 2-approximation algorithm for MAX W-LIGHT when $\Delta = 2W + 1$

In this subsection, we develop a linear-time 2-approximation algorithm for MAX W-LIGHT when $\Delta = 2W + 1$ by modifying Algorithm EulerO in Section 3. (Recall that if $\Delta = 2W$, MAX W-LIGHT can be solved in linear time by Corollary 4.)

We change the way that Algorithm EulerO chooses pairs of vertices having odd degrees when inserting edges to make all degrees even. The new algorithm is:

Algorithm EulerO+

Input: An undirected graph $G = (V, E)$ of maximum degree $2W + 1$

Output: An orientation Λ of G

1. If there exist at least two vertices of degree $2W + 1$ then select any pair of two such vertices and insert an edge between them. Otherwise, goto Step 3.
2. Repeat Step 1 until at most one vertex has degree $2W + 1$.
3. If there remains a vertex v of degree $2W + 1$ then insert an edge $e = \{v, u\}$, where u is an arbitrary vertex of odd degree. Otherwise, let e be any existing edge and let its endpoints be v and u.
4. While there exist vertices having odd degrees, select any pair of two such vertices and insert an edge between them.
5. Let G^+ be the obtained multigraph and let $E^+ := E(G^+) \setminus E(G)$.
6. Find an Eulerian orientation Λ^+ of G^+ such that the edge e is oriented as (v, u).
7. Output $\Lambda := \Lambda^+ \setminus \Lambda^+(E^+)$.

Theorem 9 *If* $\Delta = 2W+1$ *then Algorithm* EulerO+ *is a linear-time 2-approximation algorithm for* MAX W-LIGHT.

Proof As in the proof of Lemma 1, all vertices having degree at most $2W$ in the original input graph G have outdegree at most W in $\Lambda(G)$, i.e., are W-light in $\Lambda(G)$. Let $V' \subseteq V$ be the set of vertices having degree exactly equal to $2W + 1$ in G, and let $n' = |V'|$. By the construction of G^+ and the definition of an Eulerian orientation, every vertex in V' has indegree $W + 1$ and outdegree $W + 1$ in $\Lambda^+(G^+)$. Furthermore, for any pair of vertices $u, v \in V'$ that are connected by an edge in E^+, one of u and v will have indegree W and outdegree $W + 1$ in $\Lambda(G)$, and the other one will have indegree $W + 1$ and outdegree W.

If n' is even, the number of W-light vertices in $\Lambda(G)$ is at least $(n - n') + n'/2 = n - n'/2 \geq n/2$ since $n' \leq n$. If n' is odd, one vertex $v \in V'$ may be connected to a vertex $u \notin V'$ by an edge $e = \{u, v\} \in E^+$ in Step 3. In Step 6, this edge is oriented as (v, u) in Λ^+, which makes the outdegree of v in $\Lambda(G)$ equal to W while u is still W-light in $\Lambda(G)$. Thus, in case n' is odd, the number of W-light vertices in $\Lambda(G)$ is also $(n - n') + ((n' - 1)/2) + 1 \geq n/2$. Finally, since the number of W-light vertices in any optimal orientation of G is at most n, the approximation ratio of EulerO+ is 2.

To achieve a time complexity of $O(m)$, apply the algorithm from Proposition 3 to implement Step 6; all other steps are straightforward to implement in linear time. □

To demonstrate that the approximation ratio 2 for Algorithm EulerO+ mentioned in Theorem 9 is asymptotically tight, consider the complete graph K_{2W+2}, which has $\Delta = 2W + 1$. As shown in Fig. 9 (for the case $W = 2$), Algorithm EulerO+ outputs an orientation of K_{2W+2} with $W + 1$ vertices that are W-light (outdegree W, indegree $W + 1$) and $W + 1$ vertices that are $(W + 1)$-heavy (outdegree $W + 1$,

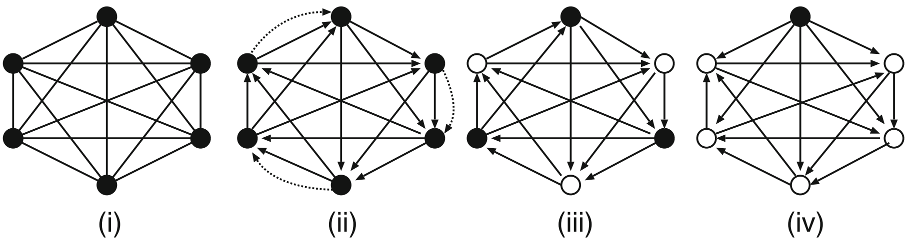

Fig. 9 A tight example for EulerO+ when $W = 2$. (i) The complete graph K_6; (ii) The graph K_6^+ and an Eulerian orientation of K_6^+; (iii) The orientation of K_6 produced by EulerO+; (iv) An optimal orientation of K_6. White vertices in (iii) and (iv) are 2-light

indegree W). On the other hand, an optimal solution has $2W + 1$ vertices that are W-light (outdegree W, indegree $W + 1$) and only one $(W + 1)$-heavy vertex (outdegree $2W + 1$, indegree 0), obtained by selecting any vertex v in K_{2W+2}, orienting all its incident edges outwards, and taking an Eulerian orientation as in Proposition 2 of $K_{2W+2} \setminus \{v\}$. Here, the optimal number of W-light vertices is $2 - O(1/W)$ times larger than the number of W-light vertices obtained by EulerO+.

7 Concluding remarks

In this paper, we have derived several new results on the complexity of the graph orientation problems MAX W-LIGHT, MIN W-LIGHT, MAX W-HEAVY, and MIN W-HEAVY. See Table 1 in Section 1 for a summary.

A simple $(n/(2W + 1))$-approximation algorithm for MAX W-LIGHT was given in Theorem 1, which almost matches the $(n/W)^{1-\varepsilon}$-inapproximability bound in Theorem 4. By combining it with Reverse from [6] (see Section 6.1 above), a slightly better approximation ratio can be achieved: (i) Apply the $(n/(2W + 1))$-approximation algorithm to G; (ii) Apply Reverse to each induced subgraph of G of size $n - i$, where $i \in \{0, 1, \ldots, c\}$ for some constant c, and orient any remaining edges of G appropriately; and (iii) Output the best orientation found. If an optimal solution has at least $n - c$ vertices that are W-light, it will be found in Step (ii). By an argument similar to the one in Theorem 1, the approximation ratio is decreased to $(n - c)/(2W + 1)$. The running time will increase, but is still polynomial if c is constant.

The problems were defined for unweighted graphs. A natural generalization is to allow the vertices to be weighted and try to minimize (or maximize) the total weight of the heavy (or light) vertices in the output orientation. Under this generalization, designing algorithms becomes harder in general, but some of the presented approximation algorithms (e.g., the ones in Section 4) can easily be adjusted while keeping the same approximation guarantees. Alternatively, the problems can be generalized by allowing the edges to be weighted, in which case the outdegree of a vertex is defined as the total weight of its outgoing edges.

An interesting open question is whether or not Min 1-Light and Max 2-Heavy are $\mathcal{NP}$-hard for general graphs. Currently, all we know is that they can be solved in polynomial time in case $\Delta \leq 3$ (Corollary 9) or $\delta \geq 4$ (Corollary 8). In comparison, Min 0-Light and Max 1-Heavy are in $\mathcal{P}$, while Min W-Light and Max $(W+1)$-Heavy are $\mathcal{NP}$-hard for every fixed $W \geq 2$ (see [5]). Another topic for future research is to close the gaps between the known polynomial-time approximability and inapproximability bounds of the problems.

A Appendix: An alternative proof of the submodularity of g_1

Suppose $E = \{e_1, e_2, \ldots, e_m\}$. Let $U = \bigcup_{i=1}^{W+1}\{v^{(i)} : v \in V\}$ be a set of $W+1$ copies of the vertices of V. To represent adjacency between the endpoints of e_i in G, define $E_{i,j} = \{\{e_i, u^{(j)}\}, \{e_i, v^{(j)}\} : e_i = \{u, v\} \in E\}$. For each edge e_i, let E_i denote $\bigcup_{j=1}^{W+1} E_{i,j}$.

Consider a bipartite graph $H = (E \cup U, E')$ with $(W+1)n + m$ vertices and $2(W+1)m$ edges, where $E' = \bigcup_{i=1}^{m} E_i$. Every matching M in H can be defined as a subset of E'. Then, the family of matchings is denoted by $\mathcal{M} = \{M : M \text{ is a matching in } H\}$. Here it is easy to see that a pair $(E', \mathcal{M})$ is a transversal matroid induced by $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$.

We now show the correspondence between the matroid $(E', \mathcal{M})$ and the objective value of $P_1(G, W, S)$. The key observation is that a matching edge $\{e, u\}$ in H corresponds to orienting the edge e away from the vertex u in G. We can make an orientation of G based on a matching in H: If an edge e is an endpoint of the matching edge $\{e, u\}$ (called *type I*), then orient e away from u in G. After that, orient the remaining edges (*type II*) arbitrarily in G.

For each v, let C_v denote $\bigcup_{i=1}^{W+1}\{v^{(i)}\}$. Consider the induced subgraph $H' = H[E \cup \bigcup_{v \in V \setminus S} C_v]$ of H and a maximum matching M^* in H'. For each $v \in \bigcup_{u \in V \setminus S} C_u$ in H', the number n_v of vertices in C_v covered by M^* is equal to $\min\{W+1, d^+_\Lambda(v)\}$ under the above constructed orientation Λ of G: Since H has only $W+1$ copies of each vertex u, n_v is clearly at most $W+1$. To obtain a contradiction, assume $d^+_\Lambda(v) < W+1$ and $n_v \neq d^+_\Lambda(v)$. If $n_v < d^+_\Lambda(v)$, there is an edge $e = \{v, u\}$ of type II for some u in G. The existence of such an edge in G implies that the edges of the form $\{e, v^{(i)}\}$ and $\{e, u^{(i)}\}$ exist in H but none of them are included in M^*, which contradicts the optimality of M^*. In addition, the above procedure to construct an orientation guarantees that $n_v \leq d^+_\Lambda(v)$. Hence, the optimal value of $P_1(G, W, S)$, i.e., $\max_{\Lambda \in \mathcal{O}(G)} \sum_{v \in V \setminus S} \min\{W+1, d^+_\Lambda(v)\}$ equals the size of a maximum matching in H'.

Finally, we verify the submodularity of g_1. For $T \subseteq E'$, consider the subgraph $H[T]$ of H. The size of a maximum matching in $H[T]$ is equal to the rank function of the matroid $(E', \mathcal{M})$:

$$r_{\mathcal{M}}(T) = \max\{|Z| : Z \subseteq T, Z \in \mathcal{M}\}$$

For a subset $S \subseteq V$ of G, we can define $T_S \subseteq E'$ as $\{\{e, v\} : \{e, v\} \in E', v \notin S\}$. Hence, the optimal value of $P_1(G, W, S)$ can be rewritten as $r_{\mathcal{M}}(T_S)$. It is well known that the rank function of a (transversal) matroid is submodular and that the sum of two submodular functions is submodular (e.g., [25]). It follows that $g_1(S) = r(T_S) + |S|(W + 1)$ is submodular since $|S|(W + 1)$ is also submodular.

Acknowledgments This work was supported by KAKENHI grant numbers 23500020, 25104521, 25330018, 26330017, and 26540005 and The Hakubi Project at Kyoto University. The authors would like to thank the anonymous reviewers for their detailed comments and suggestions that helped to improve the presentation of the paper, and Peter Floderus for pointing out an error in one of the figures.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. Prentice Hall (1993)
2. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Graph orientation to maximize the minimum weighted outdegree. Int. J. Found. Comput. Sci. **22**(3), 583–601 (2011)
3. Asahiro, Y., Jansson, J., Miyano, E., Ono, H., Zenmyo, K.: Approximation algorithms for the graph orientation minimizing the maximum weighted outdegree. J. Comb. Optim. **22**(1), 78–96 (2011)
4. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Upper and lower degree bounded graph orientation with minimum penalty. In: Proceedings of CATS 2012. CRPIT Series **128**, 139–146 (2012)
5. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Graph orientations optimizing the number of light or heavy vertices. In: Proceedings of ISCO 2012. LNCS **7422**, 332–343 (2012)
6. Asahiro, Y., Miyano, E., Ono, H., Zenmyo, K.: Graph orientation algorithms to minimize the maximum outdegree. Int. J. Found. Comput. Sci. **18**(2), 197–215 (2007)
7. Chlebik, M., Chlebikova, J.: Complexity of approximating bounded variants of optimization problems. Theor. Comput. Sci. **354**(3), 320–338 (2006)
8. Chrobak, M., Eppstein, D.: Planar orientations with low out-degree and compaction of adjacency matrices. Theor. Comput. Sci. **86**(2), 243–266 (1991)
9. Chung, F.R.K., Garey, M.R., Tarjan, R.E.: Strongly connected orientations of mixed multigraphs. Networks **15**(4), 477–484 (1985)
10. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. Ann. Math. **162**(1), 439–485 (2005)
11. Ebenlendr, T., Krčál, M., Sgall, J.: Graph balancing: A special case of scheduling unrelated parallel machines. Algorithmica **68**(1), 62–80 (2014)
12. Feige, U.: Approximating maximum clique by removing subgraphs. SIAM J. Discret. Math. **18**(2), 219–225 (2004)
13. Frank, A.: Connections in Combinatorial Optimization. Oxford University Press (2011)
14. Frank, A., Gyárfás, A.: How to orient the edges of a graph? Combinatorics Volume I, North-Holland, 353–364 (1978)
15. Gabow, H.N.: Upper degree-constrained partial orientations. In: Proceedings of SODA 2006, 554–563 (2006)
16. Garey, M., Johnson, D.: Computers and Intractability – A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
17. Goldberg, A.V., Rao, S.: Beyond the flow decomposition barrier. J. ACM **45**(5), 783–797 (1998)
18. Hakimi, S.L.: On the degrees of the vertices of a directed graph. J. Frankl. Inst. **279**(4), 290–308 (1965)
19. Kowalik, L.: Approximation scheme for lowest outdegree orientation and graph density measures. In: Proceedings of ISAAC 2006. LNCS **4288**, 557–566 (2006)
20. Landau, H.G.: On dominance relations and the structure of animal societies: III The condition for a score structure. Bull. Math. Biophys. **15**(2), 143–148 (1953)
21. Nash-Williams, C.St.J.A.: On orientations, connectivity and odd-vertex-pairings in finite graphs. Can. J. Math. **12**(4), 555–567 (1960)

22. Orlin, J.B.: A polynomial time primal network simplex algorithm for minimum cost flows. Math. Program. **97**, 109–129 (1997)
23. Orlin, J.B.: Max flows in $O(nm)$ time, or better. In: Proceedings of STOC 2013, 765–774 (2013)
24. Robbins, H.E.: A theorem on graphs, with an application to a problem of traffic control. The American Mathematical Monthly **46**(5), 281–283 (1939)
25. Schrijver, A.: Combinatorial Optimization. Springer (2003)
26. Stanley, R.P.: Acyclic orientations of graphs. Discret. Math. **5**(2), 171–178 (1973)
27. Trevisan, L.: Non-approximability results for optimization problems on bounded degree instances. In: Proceedings of STOC 2001, 453–461(2001)
28. Venkateswaran, V.: Minimizing maximum indegree. Discret. Appl. Math. **143**(1-3), 374–378 (2004)
29. Wolsey, L.A.: An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2**(4), 385–393 (1982)
30. Zuckerman, D.: Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. Theory of Computing **3**(1), 103–128 (2007)