Degree-Constrained Graph Orientation: Maximum Satisfaction and Minimum Violation*

Yuichi Asahiro[1], Jesper Jansson[2], Eiji Miyano[3], and Hirotaka Ono[4]

[1] Department of Information Science, Kyushu Sangyo University, Higashi-ku, Fukuoka 813-8503, Japan
asahiro@is.kyusan-u.ac.jp

[2] Laboratory of Mathematical Bioinformatics, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan
jj@kuicr.kyoto-u.ac.jp

[3] Department of Systems Design and Informatics, Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502, Japan
miyano@ces.kyutech.ac.jp

[4] Department of Economic Engineering, Kyushu University, Higashi-ku, Fukuoka 812-8581, Japan
hirotaka@en.kyushu-u.ac.jp

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 [4]: 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. The problems' computational complexities vary with W. Here, we resolve several open questions related to their polynomial-time approximability and present a number of positive and negative results.

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 $d^+_\Lambda(u) \leq W$, and W*-heavy in* $\Lambda(G)$ if

* Supported by KAKENHI Grant Numbers 21680001, 23500020, 25104521, and 25330018 and The Hakubi Project at Kyoto University.

C. Kaklamanis and K. Pruhs (Eds.): WAOA 2013, LNCS 8447, pp. 24–36, 2014.

$d_\Lambda^+(u) \geq W$. 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 [4]. 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: $\left|\{u \in V : d_\Lambda^+(u) \leq W\}\right|$ is maximized
- MIN W-LIGHT: $\left|\{u \in V : d_\Lambda^+(u) \leq W\}\right|$ is minimized
- MAX W-HEAVY: $\left|\{u \in V : d_\Lambda^+(u) \geq W\}\right|$ is maximized
- MIN W-HEAVY: $\left|\{u \in V : d_\Lambda^+(u) \geq W\}\right|$ is minimized

We write $n = |V|$ and $m = |E|$ for the input graph G.

The *degree of* u *in* G is denoted by $d(u)$. We define $\delta = \min\{d(u) \mid u \in V\}$ and $\Delta = \max\{d(u) \mid u \in V\}$. For any $U \subseteq V$, the subgraph induced by U is denoted by $G[U]$.

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 but their polynomial-time approximability properties may differ. The same observation holds for the pair MIN W-LIGHT and MAX $(W+1)$-HEAVY.

The computational complexities of MAX W-LIGHT, MIN W-LIGHT, MAX W-HEAVY, and MIN W-HEAVY were studied for different values of W in [4]. As observed in [4], the special case of MAX 0-LIGHT is equivalent to the well-known MAXIMUM INDEPENDENT SET problem (and the supplementary problem MIN 1-HEAVY is equivalent to MINIMUM VERTEX COVER). Thus, allowing the value of W to vary yields a natural generalization of MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER. In many cases, however, the (in)tractability and the (in)approximability remained unknown. In this paper, we establish several new results on the polynomial-time approximability of these problems.

New Results: Below is a summary of previous results from [4] and the new results presented in this paper (see Table 1 for a summary). Due to space limitations, many technical details will be deferred to the full version of the paper.

– 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 $\mathcal{P} = \mathcal{NP}$ [4,31]. Theorem 6 of Sect. 4 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 7 of Sect. 4 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 $\mathcal{P} = \mathcal{NP}$ [4,9]. Theorem 5 of Sect. 4 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 $\log(\Delta - W + 1)$ in polynomial time for every fixed $W \geq 2$ in Theorem 2 of Sect. 3.

Table 1. Summary of the results from [4] and the new results in this paper

W	Max W-Light	Min $(W+1)$-Heavy
$=0$	Identical to Maximum Independent Set [4]	Identical to Minimum Vertex Cover [4]
≥ 1	Solvable in $O(n)$ time for trees [4] $(n/(2W+1))$-approx. (Theorem 7) $(n/W)^{1-\varepsilon}$-inapprox. (Theorem 6)	Solvable in $O(n)$ time for trees [4] $\log(\Delta - W)$-approx. (Theorem 2) 1.3606-inapprox. (Theorem 5)

W	Min W-Light	Max $(W+1)$-Heavy
$=0$	Solvable in $O(m^{3/2})$ time [4]	Solvable in $O(m^{3/2})$ time [4]
≥ 0	Solvable in $O(n)$ time for trees [4] Solvable in $O(n^2)$ time for outerplanar graphs [4]	Solvable in $O(n)$ time for trees [4] Solvable in $O(n^2)$ time for outerplanar graphs [4]
≥ 1	$(W+1)$-approx. [4] $\log(W+1)$-approx. (Theorem 1)	$O(n^2)$-time 2-approx. for planar graphs [4] $O(m)$-time $(W+2)$-approx. [4]
≥ 2	NP-hard for planar graphs [4]	NP-hard for planar graphs [4]
large	$(\log(W+1) - O(\log\log(W+1)))$ -inapprox. (Theorem 4)	$W^{1-\varepsilon}$-inapprox. (Theorem 3)

– Min W-Light: A polynomial-time $(W+1)$-approximation algorithm was given in [4]. Theorem 1 of Sect. 3 improves the approximation ratio to $\log(W+1)$ for any $W \geq 1$. Moreover, Theorem 4 in Sect. 4 shows that for sufficiently large W, Min W-Light is $\mathcal{NP}$-hard to approximate within $\log(W+1) - O(\log\log(W+1))$, implying that our $\log(W+1)$-approximation is almost tight.
– Max W-Heavy: It was shown in [4] that Max 1-Heavy and Min 0-Light are in $\mathcal{P}$, but Max W-Heavy and Min $(W-1)$-Light are $\mathcal{NP}$-hard for every fixed $W \geq 3$. An open problem from [4] was to determine the computational complexity of Max 2-Heavy and Min 1-Light. Now, consider two special cases: (i) $\Delta \leq 3$ and (ii) $\delta \geq 4$. Corollary 3 of Sect. 5 and Proposition 5 of Sect. 2 demonstrate that Max 2-Heavy and Min 1-Light can be solved in polynomial time for (i) and (ii), respectively. Also, Theorem 3 in Sect. 4 proves that for sufficiently large W, Max W-Heavy is $\mathcal{NP}$-hard to approximate within $W^{1-\varepsilon}$ for any $\varepsilon > 0$. The best previously known polynomial-time approximation ratio was $W+1$ [4].

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 [8,21,24] 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 [2,7,10,29] 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 [7]. There are many optimization criteria for graph orientation other than these. See [3] or chapter 61 in [27] for more details and additional references.

On the other hand, *degree-constrained* graph orientations [12,13,15,19] arise when a lower degree bound $W^l(v)$ and an upper degree bound $W^u(v)$ for each vertex v in the graph are specified, and the outdegree of v in any valid graph orientation is required to lie in the interval $W^l(v)..W^u(v)$. 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 [3,4]. 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 $W^l(v) = W$ and $W^u(v) = \infty$ for every vertex v in the input graph, then we obtain Max W-Heavy and Min $(W-1)$-Light.

2 Preliminaries

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

It is known [12] that finding the maximum density of any graph is equivalent to finding the smallest integer W such that the graph is W-orientable:

Proposition 1 ([12]). *Any graph G is W-orientable if and only if $D_{G'} \leq W$ for all induced subgraphs G' in G.*

The following immediate consequence plays an important role in the paper. Note that the orientation referred to in Proposition 2 is an Eulerian orientation.

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

[1] The ceiling function gives the maximum degree of the vertices in the subgraph induced by S, where the maximum degree is an integer here.

Proposition 3 (p. 91 of [27]). *Given a graph G with all degrees even, an Eulerian orientation of G can be found in $O(m)$ time.*

The following proposition extends the notion of density D_G for our problems:

Proposition 4. *Consider a 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 outward 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 $v \in U$ such that $d^+_\Lambda(v) \geq \lceil (|E(U)| + m')/|U| \rceil$.*

For restricted instances, Max $(W+1)$-Heavy and Min W-Light can be solved in polynomial time. The fundamental idea of the algorithm is (i) first insert matching edges between odd degree vertices, and then (ii) orient the edges along with an Eulerian tour.

Proposition 5. *If the minimum degree δ of the input graph G satisfies $W+1 \leq \lfloor \delta/2 \rfloor$, an $O(m)$-time algorithm finds an optimal orientation for* Max $(W+1)$-Heavy *and* Min W-Light*, under which no vertex is W-light.*

Let $\delta^* = \max_\Lambda \min_v d^+_\Lambda(v)$. Since the algorithm in Proposition 5 outputs an orientation under which the minimum outdegree is at least $\lfloor \delta/2 \rfloor$, it always holds that $\delta^* \geq \lfloor \delta/2 \rfloor$. A known polynomial-time algorithm from [1] named Exact-1-MaxMinO outputs an orientation under which the minimum outdegree is δ^*, which gives the following corollary:

Corollary 1. *The algorithm* Exact-1-MaxMinO *outputs an optimal orientation for* Max $(W+1)$-Heavy *and* Min W-Light *when $W+1 \leq \lfloor \delta/2 \rfloor$.*

An analogous discussion gives the following proposition and corollary, utilizing the polynomial-time algorithm Reverse from [5]:

Proposition 6. *If Δ satisfies $W \geq \lceil \Delta/2 \rceil$, then* Min $(W+1)$-Heavy *and* Max W-Light *can be solved in $O(m)$ time.*

Corollary 2. *The algorithm* Reverse *outputs an optimal orientation for* Min $(W+1)$-Heavy *and* Max W-Light *when $W \geq \lceil \Delta/2 \rceil$.*

3 Greedy Algorithms for Min *W*-Light and Min (*W* + 1)-Heavy

In this section, for general W, we present greedy algorithms for Min W-Light and for Min $(W+1)$-Heavy, which use the same framework, but different criterion functions are adopted.

Here we explain the main idea of the greedy algorithm for Min W-Light. Our algorithm sequentially chooses vertices to be removed as violating vertices (W-light vertices). We refer by S the temporary vertices to be removed in Min W-Light, that is, S starts from $\emptyset$ and the size of S increases one-by-one by a greedy manner until $V(G) \setminus S$ becomes $(W+1)$-heavy. The criterion of the greedy algorithm is defined by the following problem and its polynomial time solvability:

Problem Attainment of $(W+1)$-Heavy Orientation $(P_1(G,W,S))$

$$\begin{aligned} \max \quad & \sum_{v \in V \setminus S} \min\{W+1, d^+_\Lambda(v)\} \\ \text{subject to} \quad & \Lambda \in \mathcal{A}(G) \ , \end{aligned}$$

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

Since $P_1(G,W,S)$ can be solved via the maximum flow problem, we obtain the following lemma.

Lemma 1. 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 following maximum flow problem which can be solved in $O(m^{1.5}\min\{m^{0.5}, \log m \log W\})$ time [14,18,22]: For graph G, we construct network $\mathcal{N}(G,W,S)$, where the set of vertices is $\{s,t\} \cup E(G) \cup V(G)$ and the set of arcs is $\{(s,e) \mid e \in E\} \cup \{(e,u),(e,v) \mid e = \{u,v\} \in E(G)\} \cup \{(u,t) \mid u \in V(G)\}$. The capacities of the arcs are defined by

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

We can see that the objective value of $P_1(G,W,S)$ corresponds to the flow value of the network. In fact, flow with size one from s to $e(=\{u,v\})$ goes through either u or v (exactly one of u and v) by the flow integrality. This is interpreted as follows: $e = \{u,v\}$ is oriented as (u,v) if the flow via e goes through u, and (v,u) otherwise; the value of flow via u is considered the minimum of $W+1$ and the outdegree of u of the corresponding orientation. Thus, the optimal value of $P_1(G,W,S)$ is obtained by solving the maximum flow problem on $\mathcal{N}(G,W,S)$. □

By the optimality of the maximum flow, there is a simple characterization of an optimal orientation.

Lemma 2. *Λ is an optimal orientation of $P_1(G,W,S)$ if and only if there is not a directed path on Λ of $G \setminus S$ from any $(W+2)$-heavy vertex in $V \setminus S$ or a vertex in S to W-light vertex in $V \setminus S$.*

As mentioned above, we design a greedy algorithm that uses the optimal value of $P_1(G,W,S)$ as a criterion. Let $g_1(S)$ be the optimal value of $P_1(G,W,S)$ plus $|S|(W+1)$. It is easy to see that $g_1(S) = g_1(V)$ if $G \setminus S$ is $(W+1)$-heavy. Thus, by using this $g_1(S)$, Min W-Light can be formulated as $\min_{S \subseteq V}\{|S| \mid g_1(S) = g_1(V)\}$. We can show the following lemma.

Lemma 3. *$g_1(S)$ is a non-decreasing submodular function, that is, it satisfies that (non-decreasingness) $g_1(S \cup \{i\}) - g_1(S) \geq 0$ for any $i \in V \setminus S$, 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 two disjoint subsets $S, S' \subseteq V$ of vertices, let us denote

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

where $\mathit{OptO}(P_1(G, W, S))$ is the set of all optimal orientations of $P_1(G, W, S)$. To prove this lemma, we first show that

$$g_1(S \cup S') - g_1(S) = |S'|(W + 1) - \alpha(S, S') \tag{1}$$

holds for any disjoint $S, S' \subseteq V$. Let $\Lambda_{S,S'}$ be an orientation that achieves $\alpha(S, S')$. We can see that $\Lambda_{S,S'}$ is also an optimal orientation of $P_1(G, W, S \cup S')$. In fact, by Lemma 2 and the optimality of $\Lambda_{S,S'}$ for $P_1(G, W, S)$, there is no directed path from $(W + 2)$-heavy vertex in $V \setminus S$ or a vertex in S to W-light vertex in $\Lambda_{S,S'}$. Also, there exists no directed path from a vertex in S' to a W-light vertex in $V \setminus (S \cup S')$, otherwise it contradicts that $\Lambda_{S,S'}$ minimizes $\sum_{v \in S'} \min\{W + 1, d^+_\Lambda(v)\}$. These imply the optimality of $\Lambda_{S,S'}$ for $P_1(G, W, S \cup S')$. Thus, we have

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

which is equivalent to (1). Note that the second equality in the above is based on the fact that S and S' are disjoint. By (1), $g_1(S \cup \{i\}) - g_1(S) = (W + 1) - \alpha(S, \{i\}) \geq 0$ holds for any $i \in V \setminus S$, which implies the non-decreasing property of g_1.

We are ready to prove the submodularity of g_1. An equivalent condition of the submodularity of g_1 is that

$$g_1(S \cup \{i\}) - g_1(S) \geq g_1(S \cup \{i, j\}) - g_1(S \cup \{j\}) \tag{2}$$

for any $S \subseteq V$ and any $i, j \in V \setminus S$. By (1), we have

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

Here, it is easy to see that $\alpha(S, \{i\}) + \alpha(S, \{j\}) \leq \alpha(S, \{i, j\})$ holds. This implies (2), the submodularity of g_1. □

It is known that optimization problems that form $\min_{S \subseteq V}\{|S| \mid g(S) = g(V)\}$ can be approximated within a $\log(\max_{i\in V}\{g(\{i\})-g(\emptyset)\})$ factor by the following greedy algorithm, if g is a non-decreasing submodular function [30].

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

In our case, g_1 is a non-decreasing submodular function from Lemma 3, so it can be approximated within a $\log(\max_{i\in V}\{g_1(\{i\}) - g_1(\{\emptyset\})\}) \leq \log(W + 1)$ factor by the greedy algorithm. In our case, this algorithm can be executed by n iterations of Step 2. Step 2 is done in $O(m^{1.5} \min\{m^{0.5}, \log m \log W\}) + O(mn)$ time, where $O(m^{1.5}\{m^{0.5}, \log m \log W\})$ time is for computing the maximum flow for $g_1(\emptyset)$ based on Lemma 1, and $O(mn)$ is n-times finding an augmenting path to compute $g_1(S \cup \{i\})$ from $g_1(S)$. We obtain the following theorem:

Theorem 1. Min W-Light *can be approximated within a factor of* $\log(W+1)$ *in* $O((mn + m^{1.5} \min\{m^{0.5}, \log m \log W\})n)$ *time.*

As for Min $(W+1)$-Heavy, we can obtain the similar theorem as follows, though we need to be a little more careful because we use the minimum cost flow for the proof, and it is not as simple as the maximum flow.

Theorem 2. Min $(W + 1)$-Heavy *can be approximated within a factor of* $\log(\Delta - W)$ *in polynomial time.*

4 (In)approximability of the Problems

In this section, we give several results on the (in)approximability of the four problems, Max W-Heavy, Min W-Light, Min W-Heavy, and Max W-Light in this order.

In [4], the $\mathcal{NP}$-hardness of Max W-Heavy is shown for $W \geq 3$, however, no inapproximability results are known. The next theorem gives an inapproximability of Max W-Heavy for a sufficiently large W:

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

It should be noted that the proof of this theorem is based on the hardness of Max Independent Set. An important condition here is $W \geq \Delta$ of an instance of Max Independent Set. Since Max Independent Set is $\mathcal{NP}$-hard when $\Delta \geq 3$, the proof implies that Max W-Heavy is $\mathcal{NP}$-hard also when $W \geq 3$, i.e., we cannot show the hardness of Max W-Heavy for the case $W = 2$.

Next we give an inapproximability of Min W-Light here:

Theorem 4. Min 2-Light *and* Min 3-Light *cannot be approximated within a constant factor 100/99 and 53/52, respectively, in polynomial time unless* $\mathcal{P} = \mathcal{NP}$. *Furthermore, for sufficiently large* W, Min W-Light *cannot be approximated within a factor of* $\log(W + 1) - O(\log \log W)$ *in polynomial time unless* $\mathcal{P} = \mathcal{NP}$.

Since Min 1-Heavy is equivalent to Min Vertex Cover [4], it can be approximated within a ratio of $2 - \Theta(1/\sqrt{\log n})$ [16]. Also, in this paper, we designed $O(\log(\Delta - W))$-approximation algorithm for Min W-Heavy in Theorem 2. On the other hand, the following inapproximability of Min W-Heavy can be also shown.

Theorem 5. *For every fixed* $W \geq 1$, Min W-Heavy *cannot be approximated within a ratio of* 1.3606 *in polynomial time unless* $\mathcal{P} = \mathcal{NP}$.

Proof. Since Min 1-Heavy is equivalent to Minimum Vertex Cover[4], Min 1-Heavy cannot be approximated within a ratio of 1.3606 in polynomial time unless $\mathcal{P} = \mathcal{NP}$ [9]. The hardness of approximating Min W-Heavy for every fixed $W \geq 2$ is shown by a gap-reserving reduction from Minimum Vertex Cover. Let $G = (V(G), E(G))$ be an input graph of Minimum Vertex Cover with n vertices. Then, we construct a graph $H = (V(H), E(H))$ of Min W-Heavy from G. Let $OPT(G)$ and $OPT'(H)$ denote the values of optimal solutions for G of Minimum Vertex Cover and for H of Min W-Heavy, respectively. Let $V(G) = \{v_1, v_2, \cdots, v_n\}$ of n vertices in G. The constructed graph H has n subgraphs H_1 through 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, $u_{i,1}$ 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 the subgraph H_i is $(2W-1)(2W-2)/2 + (W-1) = 2W(W-1)$. If $\{v_i, v_j\}$ in G of Minimum Vertex Cover, then H_i and H_j are connected by an edge $\{u_{i,0}, u_{j,0}\}$. This reduction can be done in polynomial time. In the following we show that this reduction can completely preserve the approximation gap of $\alpha = 1.3606$ in Minimum Vertex Cover, i.e., $OPT(G) \leq k$ if and only if $OPT'(H) \leq k$ holds.

The following simple observation plays a key role in this proof: Now suppose that $\{v_i, v_j\} \in E(G)$. Then, consider the subgraph $G[V(H_i) \cup V(H_j)]$ induced by $V(H_i)$ and $V(H_j)$ connected by the edge $\{u_{i,0}, u_{j,0}\}$. One can see that $G[V(H_i) \cup V(H_j)]$ contains $|V(H_i)| + |V(H_j)| = 4W$ vertices and $|E(H_i)| + |E(H_j)| + 1 = 4W(W-1) + 1$ edges; the density of $G[V(H_i) \cup V(H_j)]$ is larger than $W-1$. This means that the maximum density is at least W so that at least one vertex in $G[V(H_i) \cup V(H_j)]$ must be W-heavy.

(Only-if part) Consider a vertex cover $S \subseteq V(G)$ with size at most k of G. Then we can give the following orientation of H: For the internal edges of K^i_{2W-1} in the ith subgraph H_i for every $i = 1, 2, \cdots, n$, we give an arbitrary orientation in which every vertex has outdegree $W-1$ by Proposition 2. The number of edges between $u_{i,0}$ and the complete graph K^i_{2W-1} in H_i is $W-1$, and those are oriented from $u_{i,0}$ to $W-1$ vertices in K^i_{2W-1}. At this moment, the outdegree of $u_{i,0}$ is exactly $W-1$. For an edge $\{u_{i,0}, u_{j,0}\}$ between H_i and H_j where $v_i \in S$ and $v_j \in V \setminus S$, we orient it from $u_{i,0}$ to $u_{j,0}$. If both vertices v_i and v_j are in S, then the edge $\{u_{i,0}, u_{j,0}\}$ is oriented arbitrarily. Since at least one vertex in $\{u_{i,0}, u_{j,0}\}$ between H_i and H_j is in S, the outdegree of a vertex in $V(G) \setminus S$ is $W-1$. The number of W-heavy vertices is at most k.

(If part) Consider an orientation Λ such that the number of W-heavy vertices in H is at most k. As observed above, at least one vertex in the subgraph induced by two subgraphs H_i and H_j corresponding to two vertices in an edge $\{v_i, v_j\}$ in G is W-heavy. If the W-heavy vertex is in H_i, then we select the vertex v_i into the subset S of vertices. Otherwise, the vertex v_j is selected into S. Then, at least one endpoint of every edge in $E(G)$ must be in S. Thus, S is a vertex cover of G and $|S| \leq k$ holds by the assumption. □

Since MAX 0-LIGHT is equivalent to MAX INDEPENDENT SET [4], it cannot be approximated within a factor of $n^{1-\varepsilon}$ [31] while it can be approximated within a factor of $n(\log\log n)^2/(\log n)^3$ [11]. In the following we give the inapproximability and the approximability of MAX W-LIGHT for $W \geq 1$:

Theorem 6. *For every fixed $W \geq 1$,* MAX W-LIGHT *cannot be approximated within a factor of $(n/W)^{1-\varepsilon}$ in polynomial time unless $\mathcal{P} = \mathcal{NP}$.*

The following algorithm runs in linear time. Although it is quite simple, its approximation ratio is almost tight due to the inapproximability ratio of $\Omega((n/W)^{1-\varepsilon})$ above.

1. Pick any $\min\{2W + 1, n\}$ vertices in the input G. Let the set of the chosen vertices be U.
2. Apply the algorithm in Prop. 6 to $G[U]$.
3. Orient the edges in $E \setminus E(U)$ connecting to any vertex in U towards U.
4. Orient the remaining edges arbitrarily.

Theorem 7. *There is a linear time $n/(2W + 1)$-approximation algorithm for* MAX W-LIGHT.

5 Degree-Bounded Graphs

In this section, the obtained results for input graphs with bounded degrees are briefly summarized.

First we can obtain a polynomial time 2-approximation algorithm by a slight modification to the one in Prop. 5; the main idea of the modification is to choose appropriate pairs of vertices having odd degrees, when inserting matching edges. Recall that if $\Delta = 2W$, then the problem MAX W-LIGHT can be solved in polynomial time by Corollary 2.

Theorem 8. *If $\Delta = 2W + 1$, there is a polynomial time 2-approximation algorithm for* MAX W-LIGHT.

Next theorem shows the $\lfloor \frac{\Delta}{2} \rfloor$-approximability for MAX 2-HEAVY. The algorithm roughly works as follows: (i) first it obtains a line graph $L(G)$ of the input graph G, (ii) finds a maximum matching in $L(G)$, then (iii) converts the obtained matching to an orientation of G. Here an important property is that the size of the maximum matching in $L(G)$ guarantees the number of 2-heavy vertices in the resulted directed graph.

Theorem 9. *There is a polynomial time* $\lfloor \Delta/2 \rfloor$*-approximation algorithm for* MAX 2-HEAVY.

Based on this theorem, the following corollary holds, which shows one side of the complexity of MAX 2-HEAVY and MIN 1-LIGHT; it is unknown whether MAX 2-HEAVY and MIN 1-LIGHT are $\mathcal{NP}$-hard or not for general.

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

6 Concluding Remarks

In this paper, we have derived several new results on the complexity of MAX W-LIGHT, MIN W-LIGHT, MAX W-HEAVY, and MIN W-HEAVY. As for one technical aspect, we remark that the proof of the submodularity in Sect. 3 might be simplified using matroid theory. We would also like to note here that the 2-approximation algorithm for FEEDBACK VERTEX SET [6] gives a fundamental idea for a polynomial-time 2-approximation algorithm for MIN 2-HEAVY.

An interesting open question is whether MAX 2-HEAVY (or MIN 1-LIGHT) is $\mathcal{NP}$-hard for general graphs. Furthermore, there are still many gaps between the known polynomial-time approximability and inapproximability bounds for the problems; investigating stricter thresholds is a further research topic.

The problems were defined on unweighted graphs. A natural generalization is to let the vertices be weighted and try to minimize (or maximize) the total weights of heavy (or light) vertices. Under this generalization, designing algorithms becomes harder in general, but some of the presented approximation algorithms (e.g., the ones in Sect. 3) can easily be adjusted to the weighted version with the same approximation guarantees. Alternatively, the problems can be generalized by allowing the edges to be weighted, in which the outdegree of a vertex is defined by the total weights of outgoing edges.

References

1. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Graph orientation to maximize the minimum weighted outdegree. International Journal of Foundations of Computer Science 22(3), 583–601 (2011)
2. Asahiro, Y., Jansson, J., Miyano, E., Ono, H., Zenmyo, K.: Approximation algorithms for the graph orientation minimizing the maximum weighted outdegree. Journal of Combinatorial Optimization 22(1), 78–96 (2011)
3. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Upper and lower degree bounded graph orientation with minimum penalty. In: Proc. of CATS 2012. CRPIT Series, vol. 128, pp. 139–146 (2012)
4. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Graph orientations optimizing the number of light or heavy vertices. In: Mahjoub, A.R., Markakis, V., Milis, I., Paschos, V.T. (eds.) ISCO 2012. LNCS, vol. 7422, pp. 332–343. Springer, Heidelberg (2012)

5. Asahiro, Y., Miyano, E., Ono, H., Zenmyo, K.: Graph orientation algorithms to minimize the maximum outdegree. International Journal of Foundations of Computer Science 18(2), 197–215 (2007)
6. Bafna, V., Berman, P., Fujito, T.: Constant ratio approximations of the weighted feedback vertex set problem for undirected graphs. In: Staples, J., Katoh, N., Eades, P., Moffat, A. (eds.) ISAAC 1995. LNCS, vol. 1004, pp. 142–151. Springer, Heidelberg (1995)
7. Chrobak, M., Eppstein, D.: Planar orientations with low out-degree and compaction of adjacency matrices. Theoretical Computer Science 86(2), 243–266 (1991)
8. Chung, F.R.K., Garey, M.R., Tarjan, R.E.: Strongly connected orientations of mixed multigraphs. Networks 15(4), 477–484 (1985)
9. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. Annals of Mathematics 162(1), 439–485 (2005)
10. Ebenlendr, T., Krčál, M., Sgall, J.: Graph balancing: A special case of scheduling unrelated parallel machines. In: Proc. of SODA 2008, pp. 483–490 (2008), Journal version: Graph balancing: A special case of scheduling unrelated parallel machines. Algorithmica (June 2012) published online doi:10.1007/s00453-012-9668-9
11. Feige, U.: Approximating maximum clique by removing subgraphs. SIAM Journal on Discrete Mathematics 18(2), 219–225 (2004)
12. Frank, A., Gyárfás, A.: How to orient the edges of a graph? In: Combinatorics, vol. I, pp. 353–364. North-Holland (1978)
13. Gabow, H.N.: Upper degree-constrained partial orientations. In: Proc. of SODA 2006, 554–563 (2006)
14. Goldberg, A.V., Rao, S.: Beyond the flow decomposition barrier. Journal of the ACM 45(5), 783–797 (1998)
15. Hakimi, S.L.: On the degrees of the vertices of a directed graph. Journal of the Franklin Institute 279(4), 290–308 (1965)
16. Karakostas, G.: A better approximation ratio for the vertex cover problem. ACM Transactions on Algorithms 5(4), Article 41(2009)
17. Kowalik, Ł.: Approximation scheme for lowest outdegree orientation and graph density measures. In: Asano, T. (ed.) ISAAC 2006. LNCS, vol. 4288, pp. 557–566. Springer, Heidelberg (2006)
18. King, V., Rao, S., Tarjan, R.E.: A faster deterministic maximum flow algorithm. J. Algorithms 23, 447–474 (1994)
19. Landau, H.G.: On dominance relations and the structure of animal societies: III The condition for a score structure. Bulletin of Mathematical Biophysics 15(2), 143–148 (1953)
20. Lovász, L.: Graph minor theory. Bulletin of the American Mathematical Society 43, 75–86 (2005)
21. Nash-Williams, C., St, J.A.: On orientations, connectivity and odd-vertex-pairings in finite graphs. Canadian Journal of Mathematics 12(4), 555–567 (1960)
22. Orlin, J.B.: Max flows in $O(nm)$ time, or better. In: Proc. of STOC 2013, pp. 765–774 (2013)
23. Picard, J.-C., Queyranne, M.: A network flow solution to some nonlinear 0-1 programming problems with application to graph theory. Networks 12, 141–159 (1982)
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. Robertson, N., Seymour, P.D.: Graph minors. XX. Wagner's conjecture. Journal of Combinatorial Theory Ser.B 92(2), 325–357 (2004)
26. Stanley, R.P.: Acyclic orientations of graphs. Discrete Mathematics 5(2), 171–178 (1973)

27. Schrijver, A.: Combinatorial Optimization. Springer (2003)
28. Vazirani, V.V.: Approximation Algorithms. Springer (2001)
29. Venkateswaran, V.: Minimizing maximum indegree. Discrete Applied Mathematics 143(1-3), 374–378 (2004)
30. Wolsey, L.A.: An analysis of the greedy algorithm for the submodular set covering Problem. Combinatorica 2(4), 385–393 (1982)
31. Zuckerman, D.: Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. Theory of Computing 3(1), 103–128 (2007)