Graph Orientations Optimizing the Number of Light or Heavy Vertices*

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. This paper introduces four graph orientation problems named Maximize W-Light, Minimize W-Light, Maximize W-Heavy, and Minimize W-Heavy, where W can be any fixed non-negative integer. In each of these problems, the input is an undirected graph G and the objective is to assign a direction to each edge in G so that the number of vertices with outdegree at most W or at least W in the resulting directed graph is maximized or minimized. We derive a number of results on the computational complexity and polynomial-time approximability of these problems for different values of W and various special classes of graphs. In particular, we show that Maximize 0-Light and Minimize 1-Heavy are equivalent to Maximum Independent Set and Minimum Vertex Cover, respectively, so by allowing the value of W to vary, we obtain a new, natural generalization of the two latter problems.

1 Introduction

Two well-studied computational problems in theoretical computer science are Maximum Independent Set and Minimum Vertex Cover. Here, the input is an undirected graph $G = (V, E)$ and the objective is to find a largest possible subset V' of V such that no two vertices in V' are adjacent in G (Maximum Independent Set) and a smallest possible subset V' of V such that every edge in E is incident to at least one vertex in V' (Minimum Vertex Cover). They

* Funded by KAKENHI grant numbers 21680001, 22650004, 22700019, and 23500020 and The Hakubi Project at Kyoto University.

A.R. Mahjoub et al. (Eds.): ISCO 2012, LNCS 7422, pp. 332–343, 2012.

were among the first problems ever to be shown to be NP-hard[1], and have been used to prove the NP-hardness of countless other problems during the 1970's [10] and onwards. In recent years, they have been central to the development of three important subfields of computational complexity: polynomial-time approximation algorithms, hardness of approximation, and parameterized complexity.

A relatively less researched area is the computational complexity of graph orientation problems.[2] By an *orientation* of an undirected graph G, we mean an assignment of a direction to each one of its edges. *Graph orientation problems* take an undirected graph G as input and ask for an orientation of G that optimizes some well-defined criterion on the resulting directed graph, e.g., involving connectivity between vertices, the diameter, acyclicity, or constraints on the vertices' indegrees and/or outdegrees. One typical application of graph orientation problems is *load balancing* for parallel machine scheduling where some "jobs" (corresponding to edges in G) have to be distributed among "machines" (corresponding to vertices in G) in a fair way. For example, a graph orientation of G that minimizes the maximum outdegree [2] can be used to support fast (in the worst case) vertex adjacency queries in G when using adjacency lists [6]. As another example, a graph orientation of G that maximizes the minimum outdegree [1] solves a special case of the Santa Claus problem [4] in which Santa Claus has a set of gifts (corresponding to edges in G) to distribute among a set of children (corresponding to vertices in G), each gift is of value to exactly two children, and the objective is to make the least lucky child as happy as possible.

In this paper, we connect the concepts of Maximum Independent Set / Minimum Vertex Cover and graph orientation. We first introduce four new, closely related graph orientation problems that we call Maximize W-Light, Minimize W-Light, Maximize W-Heavy, and Minimize W-Heavy, where W can be any fixed non-negative integer. We study their computational complexity and polynomial-time approximability for different values of W and different graph classes, and derive a number of simple results. Significantly, we demonstrate that Maximum Independent Set and Minimum Vertex Cover can be viewed as a special case of these graph orientation problems. Thus, by varying the parameter W, we obtain a new, natural generalization of Maximum Independent Set and Minimum Vertex Cover. We also investigate the connections to other graph theoretical concepts such as maximum flows, edge packings, and bipartite matchings, which we exploit to obtain efficient algorithms.

1.1 Problem Definitions

Let $G = (V, E)$ be an undirected graph with a vertex set V and an edge set E. An *orientation* Λ *of* G is a function that maps each undirected edge $\{u, v\}$ in E

[1] Karp's influential paper [17] established the NP-hardness of Maximum Clique (which is computationally equivalent to Maximum Independent Set), Minimum Vertex Cover, and several other fundamental problems.

[2] Some other aspects of graph orientations not related to computational complexity have been studied in graph theory and combinatorial optimization; see chapter 61 in [18] for a survey.

to one of the two possible directed edges (u, v) and (v, u). Applying Λ to all edges in E transforms G into a directed graph, which we denote by $\Lambda(G)$. For convenience, we write $\Lambda(E) = \bigcup_{e \in E}\{\Lambda(e)\}$ to represent the set of directed edges in $\Lambda(G)$. Next, for any vertex $u \in V$, define *the outdegree of u under Λ* as $d^+_\Lambda(u) = |\{v : (u, v) \in \Lambda(E)\}|$, i.e., the number of outgoing edges from u in the directed graph $\Lambda(G)$. For any non-negative integer W, a vertex $u \in V$ is said to be *W-light in $\Lambda(G)$* if $d^+_\Lambda(u) \leq W$, and *W-heavy in $\Lambda(G)$* if $d^+_\Lambda(u) \geq W$.

We are now ready to define four graph orientation problems called Maximize W-Light, Minimize W-Light, Maximize W-Heavy, and Minimize W-Heavy. In each problem, W is a fixed non-negative integer, the input is an undirected graph $G = (V, E)$ and the output is an orientation Λ of G such that:

- Maximize W-Light: the number of W-light vertices in $\Lambda(G)$ is maximized.
- Minimize W-Light: the number of W-light vertices in $\Lambda(G)$ is minimized.
- Maximize W-Heavy: the number of W-heavy vertices in $\Lambda(G)$ is maximized.
- Minimize W-Heavy: the number of W-heavy vertices in $\Lambda(G)$ is minimized.

Throughout the paper, we define $n = |V|$ and $m = |E|$. Without loss of generality, we assume that the input graph G is connected. For any instance of Maximize W-Light or Minimize W-Light, let $OPT(G)$ denote the number of W-light vertices in an optimal solution, and for any instance of Maximize W-Heavy or Minimize W-Heavy, let $OPT(G)$ denote the number of W-heavy vertices in an optimal solution. Consider an algorithm $\mathcal{A}$ that takes as input an undirected graph G and outputs an orientation of G. We say that $\mathcal{A}$ is a *σ-approximation algorithm* for Maximize W-Light (resp. Maximize W-Heavy), or that *$\mathcal{A}$'s approximation ratio is at most σ*, if $\mathcal{A}(G) \geq \frac{OPT(G)}{\sigma}$ holds for every G, where $\mathcal{A}(G)$ is the number of W-light (resp. W-heavy) vertices in the solution returned by $\mathcal{A}$. Similarly, we say that $\mathcal{A}$ is a *σ-approximation algorithm* for Minimize W-Light (resp. Minimize W-Heavy), or that *$\mathcal{A}$'s approximation ratio is at most σ*, if $\mathcal{A}(G) \leq \sigma \cdot OPT(G)$ holds for every G, where $\mathcal{A}(G)$ is the number of W-light (resp. W-heavy) vertices in the solution returned by $\mathcal{A}$.

1.2 Preliminaries

For any given graph $G = (V, E)$, any orientation Λ of G, and any fixed non-negative integer W, the vertex set V is partitioned into two disjoint subsets: the set of W-light vertices in $\Lambda(G)$ and the set of $(W + 1)$-heavy vertices in $\Lambda(G)$. Therefore, if $\mathcal{A}$ is an algorithm that solves Maximize W-Light exactly then $\mathcal{A}$ solves Minimize $(W + 1)$-Heavy exactly as well, and we say that Maximize W-Light and Minimize $(W + 1)$-Heavy are *supplementary problems*. The relationship between the two problems Minimize W-Light and Maximize $(W + 1)$-Heavy is analogous. In the same way, Maximum Independent Set and Minimum Vertex Cover are supplementary problems. However, when $\mathcal{A}$ is a good approximation algorithm for Maximize W-Light, it does not automatically follow that $\mathcal{A}$ yields a good approximation algorithm for Minimize $(W + 1)$-Heavy. Indeed, many pairs of supplementary problems whose

polynomial-time approximability properties differ greatly can be found in the literature; for example, Maximum Independent Set is NP-hard to approximate within a ratio of n^ϵ for any constant $0 \leq \epsilon < 1$ [19], while Minimum Vertex Cover is easy to approximate within a ratio of 2 by finding a maximal matching in the graph and outputting the set of matched vertices [10].

1.3 Related Work

We note that the algorithm in Section 4.2 of [3] computes an orientation Λ of G that minimizes $\max\{d^+_\Lambda(u) : u \in V\}$, taken over all possible orientations of G, in $O(m^{3/2} \cdot \log \Delta)$ time, where Δ is the maximum (unweighted) degree among all vertices in G. Using this algorithm, we can find an orientation of G in which *all* vertices are W-light, for any W, when such an orientation exists. However, when such an orientation does not exist, the algorithm does not help us to find a suitable solution for Maximize W-Light.[3] Similarly, Algorithm `Exact-1-MaxMin0` in Section 3 of [1] computes an orientation Λ of G maximizing $\min\{d^+_\Lambda(u) : u \in V\}$, taken over all possible orientations of G, in $O(m^{3/2} \cdot \log m \cdot (\log \Delta)^2)$ time. By running `Exact-1-MaxMin0`, it is trivial to construct an orientation of G in which all vertices are W-heavy, for any W, when one exists.

2 Maximize W-Light & Minimize $(W+1)$-Heavy

This section investigates the supplementary problems Maximize W-Light and Minimize $(W+1)$-Heavy for different values of W. (Minimize 0-Heavy is not interesting because all vertices have outdegree ≥ 0 under every orientation of G, i.e., $OPT(G) = n$ and any orientation of the edges gives an optimal solution.)

2.1 $W = 0$

We first prove the following lemma:

Lemma 1. *Let $G = (V, E)$ be an undirected graph. For any orientation Λ of G, the set of 0-light vertices in $\Lambda(G)$ forms an independent set in G. Conversely, given any independent set I in G, there exists an orientation of G in which the vertices from I are 0-light.*

Proof. $\Longrightarrow$) For any pair u, v of 0-light vertices in $\Lambda(G)$, no edges in $\Lambda(G)$ are oriented away from u or v by the definition of 0-light, so G cannot contain the edge $\{u, v\}$. Thus, the set of 0-light vertices forms an independent set in G.

$\Longleftarrow$) Define an orientation Λ of G as follows. First, for each $u \in I$, orient all edges involving u towards u. Next, orient all remaining edges arbitrarily. Obviously, every vertex from I will be 0-light in $\Lambda(G)$. □

[3] Intuitively, in some instances of Maximize W-Light, it is better to "sacrifice" one vertex by giving it a high outdegree. As an example, let G be a star graph and $W = 0$. Orienting every edge towards the center vertex minimizes $\max\{d^+_\Lambda(u) : u \in V\}$, but gives a very poor solution for Maximize 0-Light.

Recall that for any orientation Λ of G, the set of 0-light vertices in $\Lambda(G)$ and the set of 1-heavy vertices in $\Lambda(G)$ form a partition of V. Also, any $V' \subseteq V$ is an independent set in G if and only if $V \setminus V'$ is a vertex cover of G. Together with Lemma 1, this yields:

Lemma 2. *Let $G = (V, E)$ be an undirected graph. For any orientation Λ of G, the set of 1-heavy vertices in $\Lambda(G)$ forms a vertex cover of G. Conversely, given any vertex cover C of G, there exists an orientation of G in which the vertices from C are 1-heavy.*

Theorem 1. Maximize 0-Light *and* Maximum Independent Set *are equivalent, and* Minimize 1-Heavy *and* Minimum Vertex Cover *are equivalent.*

Consequently, the known hardness results for Maximum Independent Set [19] and Minimum Vertex Cover [8] immediately carry over to Maximize 0-Light and Minimize 1-Heavy. On the positive side, we can apply existing approximation algorithms for Maximum Independent Set [9] and Minimum Vertex Cover [16]. Furthermore, Maximum Independent Set and Minimum Vertex Cover can be solved in polynomial time for some classes of graphs such as bipartite graphs [12], and even in linear time for certain important special classes of graphs such as chordal graphs[4] [11]. In summary, we have:

Corollary 1.
- *([19])* Maximize 0-Light *cannot be approximated within a ratio of n^ϵ for any constant $0 \leq \epsilon < 1$ in polynomial time, unless $P = NP$.*
- *([9])* Maximize 0-Light *can be approximated within a ratio of $O(n(\log\log n)^2/(\log n)^3)$ in polynomial time.*
- *([8])* Minimize 1-Heavy *cannot be approximated within a ratio of* 1.3606 *in polynomial time, unless $P = NP$.*
- *([16])* Minimize 1-Heavy *can be approximated within a ratio of $2-\Theta(\frac{1}{\sqrt{\log n}})$ in polynomial time.*
- *([12])* Maximize 0-Light *and* Minimize 1-Heavy *restricted to bipartite graphs can be solved in polynomial time.*
- *([11])* Maximize 0-Light *and* Minimize 1-Heavy *restricted to chordal graphs can be solved in linear time.*

2.2 $W \geq 1$, Restriction to Trees

When G is a tree, we apply an algorithm named `Up-To-Roots` in [2] that works as follows: Select any node r in G, root G in r, and orient every edge towards r. Clearly, `Up-To-Roots` produces an orientation with exactly $n-1$ vertices having outdegree 1 and one vertex having outdegree 0, which means that for any $W \geq 1$, trivially, all n vertices are W-light and none are $(W+1)$-heavy. This gives:

Theorem 2. *For any $W \geq 1$,* Maximize W-Light *and* Minimize $(W+1)$-Heavy *restricted to trees can be solved in $O(n)$ time.*

[4] The class of chordal graphs includes many useful types of graphs such as interval graphs, split graphs, threshold graphs, and trees [5].

3 Minimize W-Light & Maximize $(W+1)$-Heavy

We now consider the pair of supplementary problems Minimize W-Light and Maximize $(W+1)$-Heavy. (The variant Maximize 0-Heavy is trivial with $OPT(G) = n$ because every vertex has outdegree ≥ 0 in any orientation of G.)

3.1 $W = 0$

For $W = 0$, the following theorem holds.

Theorem 3. Minimize 0-Light *and* Maximize 1-Heavy *can be solved in* $O(m^{3/2})$ *time.*

Proof. Given $G = (V, E)$, build a bipartite graph $\mathcal{B}_G = (V \cup E, F)$ whose vertices correspond to the vertices and edges of G, and whose edge set F is defined as $F = \big\{\{u, e\}, \{v, e\} \mid e = \{u, v\} \in E\big\}$, i.e., every edge $e = \{u, v\}$ in G is represented by the two edges $\{u, e\}$ and $\{v, e\}$ in $\mathcal{B}_G$. Note that $\mathcal{B}_G$ consists of $m+n = O(m)$ vertices (since G is connected) and $2m$ edges. Run the algorithm of Hopcroft-Karp [15] to find a maximum cardinality matching M in $\mathcal{B}_G$ in $O((|F|+|V|+|E|) \cdot \sqrt{|V|+|E|}) = O(m^{3/2})$ time. For every edge $\{u, e\}$ of $\mathcal{B}_G$ belonging to M, orient e away from u in G. Orient any remaining edges arbitrarily. This gives an orientation of G in which exactly $|M|$ vertices have outdegree at least 1, and this is an optimal solution for Minimize 0-Light and Maximize 1-Heavy on input G by the optimality of the maximum cardinality matching. □

3.2 Linear Time Solutions for Special Graphs

In this subsection, we consider unbounded W but restrict the input graph G to certain graph classes (planar and outerplanar graphs).

We first give a reduction from Maximize W-Heavy to a problem named Maximum H'-Edge Packing, studied in, e.g., [14]. The latter is defined as follows. Let H' be an undirected graph. The Maximum H'-Edge Packing problem takes as input an undirected graph H and asks for the maximum number of edge-disjoint isomorphic copies of H' in H. Now, given an instance $G = (V, E)$ of Maximize W-Heavy with n vertices, construct an instance of Maximum $K_{1,(W+n)}$-Edge Packing, where $K_{1,(W+n)}$ denotes the star graph consisting of one center vertex with $W + n$ neighboring leaves. To do this, let H be a copy of G, and for each $v \in V$, create n new vertices and attach them to v in H. Thus, H contains $n + n^2$ vertices and $O(n^2)$ edges. Then, we have the following:

Lemma 3. *For any positive integer x, the graph G has an orientation Λ such that $\Lambda(G)$ contains at least x W-heavy vertices if and only if the graph H contains at least x edge-disjoint copies of $K_{1,(W+n)}$.*

By transforming G to H as explained above and then applying the algorithms from [14] for Maximum H'-Edge Packing, where H' is a star graph, we obtain:

Theorem 4. • *For any* $W \geq 0$, MINIMIZE W-LIGHT *and* MAXIMIZE $(W+1)$-HEAVY *restricted to outerplanar graphs can be solved in* $O(n)$ *time.*
• *For any* $W \geq 0$, MAXIMIZE $(W+1)$-HEAVY *restricted to planar graphs has an* $O(n^2)$*-time 2-approximation algorithm.*

Proof. If the given instance G of MINIMIZE W-LIGHT / MAXIMIZE $(W+1)$-HEAVY is an outerplanar graph then the constructed graph H is also outerplanar. According to Theorem 4.2 in [14], MAXIMUM H'-EDGE PACKING restricted to outerplanar graphs can be solved in $O(|V_H|)$ time by dynamic programming, where $|V_H|$ is the number of vertices in H, when H' is any star graph with at least 3 leaves. By Lemma 3 above, setting $H' = K_{1,(W+1+n)}$ and running the algorithm in Theorem 4.2 in [14] on H solves MINIMIZE W-LIGHT / MAXIMIZE $(W+1)$-HEAVY in $O(n^2)$ time. The running time can be improved to $O(n)$ by bypassing the dynamic programming computations on the n^2 dummy vertices.

In the slightly more general case where G is a planar graph, H becomes planar, and we apply Theorem 5.3 in [14] instead, which says that MAXIMUM H'-EDGE PACKING restricted to planar graphs admits an $O(|E_H|)$-time 2-approximation algorithm when H' is a star graph with at least 3 leaves and where $|E_H|$ is the number of edges in H. □

3.3 $W \geq 2$, NP-Hardness

Theorem 3.1 in [14] proves that for any fixed $W \geq 2$, MAXIMUM $K_{1,(W+1)}$-EDGE PACKING is NP-hard, even if restricted to planar graphs. (Recall that $K_{1,(W+1)}$ denotes the star graph with one center vertex and $W+1$ leaves.) The reduction is from PLANAR 3-SAT. We observe that in the reduction, every vertex in the constructed graph H has degree strictly less than $2(W+1)$. Therefore, any two copies of $K_{1,W+1}$ in H must use different center vertices, and it follows that any set of x edge-disjoint copies of the star graph $K_{1,(W+1)}$ in H induces an orientation of H in which x vertices are $(W+1)$-heavy, and vice versa. Therefore, optimal solutions to the two problems MAXIMIZE $(W+1)$-HEAVY and MAXIMUM $K_{1,(W+1)}$-EDGE PACKING for the constructed graph H are equivalent.

Theorem 5. *For any fixed* $W \geq 2$, MINIMIZE W-LIGHT *and* MAXIMIZE $(W+1)$-HEAVY *are NP-hard, even if restricted to planar graphs.*

3.4 $W \geq 1$, An Approximation Algorithm for MAXIMIZE $(W+1)$-HEAVY

The following is a greedy approximation algorithm for MAXIMIZE $(W+1)$-HEAVY for general graphs which we call `Greedy_Graph_Orientation`:

1. Repeat until all vertices have been considered exactly once:
1.1 Select any previously unconsidered vertex u.
1.2 If u has $\geq W+1$ incident unoriented edges then orient any $W+1$ of them away from u.
2. Orient any remaining unoriented edges arbitrarily.

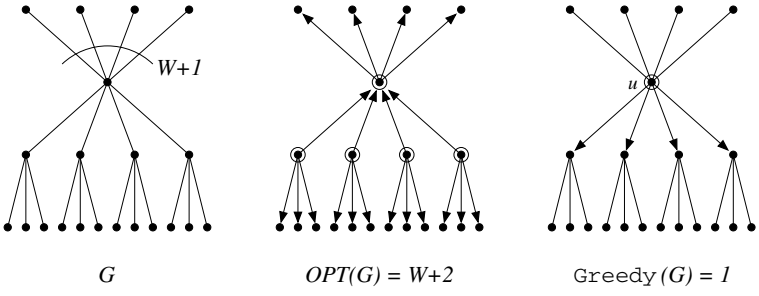

Fig. 1. A bad example for `Greedy_Graph_Orientation`. (In the figure, $W = 3$.) $OPT(G) = W + 2$, but if the algorithm first selects u and orients $(W + 1)$ edges away from u as on the right, then only u can be $(W + 1)$-heavy in any resulting orientation.

Theorem 6. *For any* $W \geq 1$, `Greedy_Graph_Orientation` *is a* $(W+2)$*-approximation algorithm for* MAXIMIZE $(W + 1)$-HEAVY *and runs in linear time.*

Proof. Let Λ be the partial orientation on E defined after Step 1 of the algorithm, where "partial" means that we allow some edges in E to remain unoriented. Let Λ^* be any optimal solution. Denote the set of $(W + 1)$-heavy vertices in $\Lambda(G)$ by S and the set of $(W + 1)$-heavy vertices in $\Lambda^*(G)$ by S^*. For any u in S, u takes an edge from at most $(W + 1)$ vertices that belong to S^* and therefore prevents at most $(W+1)$ of the vertices in S^* from being $(W+1)$-heavy in $\Lambda(G)$. Denote the set of these vertices by S^*_u. Then, S^* can be partitioned into $\bigcup_{u\in S} S^*_u$ and $S \cap S^*$ (this is because if there is a vertex in $S^* \setminus (\bigcup_{u\in S} S^*_u \cup (S \cap S^*))$, it must be selected in Step 1.2 and thus be included in $S \cap S^*$, which would give a contradiction). Hence, the approximation ratio is:

$$\frac{|S^*|}{|S|} = \frac{|\bigcup_{u\in S} S^*_u| + |S \cap S^*|}{|S|} \leq \frac{\sum_{u\in S} |S^*_u| + |S|}{|S|} \leq \frac{\sum_{u\in S}(W + 1) + |S|}{|S|} = W + 2$$

Step 1 is performed exactly n times, and each edge is considered at most two times in total, so the algorithm can be implemented to run in $O(n+m)$ time. □

The approximation ratio $(W+2)$ in Theorem 6 is tight. See Fig. 1 for an example.

We remark that Theorem 5.1 in [14] describes an approximation algorithm for MAXIMUM H'-EDGE PACKING which is based on a similar idea. However, we cannot apply it here directly because it assumes that H' is fixed; it identifies all copies of H' in H, so its running time depends exponentially on the size of H'.

3.5 $W \geq 1$, An Approximation Algorithm for MINIMIZE W-LIGHT

Finally, we give a polynomial-time approximation algorithm for MINIMIZE W-LIGHT for any fixed $W \geq 1$. It is based on computing maximum flows in a family of flow networks $\{\mathcal{N}_G(0), \mathcal{N}_G(1), \ldots, \mathcal{N}_G(n)\}$ with positive edge capacities.

Let $G = (V, E)$ be any input undirected graph to MINIMIZE W-LIGHT. Define a directed graph $\mathcal{N}_G = (V_G, E_G)$ as illustrated in Fig. 2 by setting:

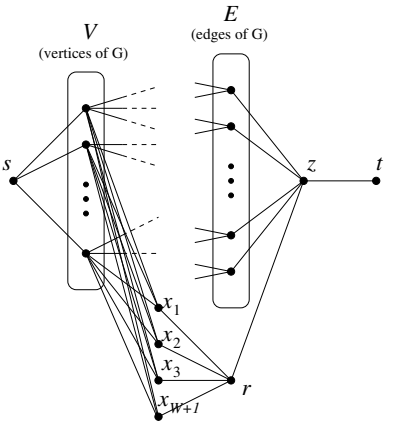

Fig. 2. The directed graph $\mathcal{N}_G$. All edges are directed from left to right.

$V_G = V \cup E \cup \{r, s, t, z\} \cup \{x_1, x_2, \ldots, x_W, x_{W+1}\}$,
$E_G = \big\{(s, v) \mid v \in V\big\} \cup \big\{(v_i, e), (v_j, e) \mid e = \{v_i, v_j\} \in E\big\} \cup \big\{(e, z) \mid e \in E\big\} \cup$
$\big\{(v, x_i) \mid v \in V, 1 \le i \le W + 1\big\} \cup \big\{(x_i, r) \mid 1 \le i \le W + 1\big\} \cup \big\{(r, z), (z, t)\big\}$

(Each vertex in $\mathcal{N}_G$ that corresponds to a vertex v in G has outdegree $\deg(v) + W + 1$, and each vertex in $\mathcal{N}_G$ that corresponds to an edge in G has indegree 2.)

Next, for any integer $q \in \{0, 1, \ldots, n\}$, let $\mathcal{N}_G(q) = (V_G, E_G, cap_q)$ be the flow network obtained by augmenting $\mathcal{N}_G$ with edge capacities cap_q, where:

$$cap_q(a) = \begin{cases} W + 1, & \text{if } a = (s, v) \text{ with } v \in V; \\ 1, & \text{if } a = (v, e) \text{ with } v \in V,\ e \in E; \\ 1, & \text{if } a = (v, x_i) \text{ with } v \in V,\ 1 \le i \le W + 1; \\ 1, & \text{if } a = (e, z) \text{ with } e \in E; \\ q, & \text{if } a = (x_i, r) \text{ with } 1 \le i \le W + 1; \\ (W + 1) \cdot n, & \text{if } a = (r, z) \text{ or } a = (z, t). \end{cases}$$

Lemma 4. *For any $q \in \{0, 1, \ldots, n\}$, if the maximum directed flow from vertex s to vertex t in $\mathcal{N}_G(q)$ equals $(W + 1) \cdot n$ then there exists an orientation of G in which the number of W-light vertices is at most $(W + 1) \cdot q$.*

Proof. Fix q and let $\mathcal{F}$ be any maximum directed flow from s to t in $\mathcal{N}_G(q)$ with integer values. Suppose $\mathcal{F}$ has value $(W+1) \cdot n$. Construct an orientation Λ of G as follows. Every vertex in $\mathcal{N}_G$ that corresponds to an edge $e = \{u, v\}$ can receive at most one unit of flow in $\mathcal{F}$, and this will arrive at e either along (u, e), in which case we orient e in Λ as (u, v), or along (v, e), in which case we orient e in Λ as (v, u). Next, orient all remaining unoriented edges of G arbitrarily. Observe that for any $v \in V$, if the corresponding vertex in $\mathcal{N}_G$ does not send any of its $(W+1)$ units of flow to $\{x_1, x_2, \ldots, x_W, x_{W+1}\}$ then v is $(W+1)$-heavy in $\Lambda(G)$.

By the construction of $\mathcal{N}_G(q)$, at most $(W + 1) \cdot q$ units of flow can pass through r. Each of the n vertices in $\mathcal{N}_G$ corresponding to a vertex in V receives

precisely $W+1$ units of flow from s in $\mathcal{F}$, and at most $(W+1)\cdot q$ of them send at least one unit of flow to $\{x_1, x_2, \ldots, x_W, x_{W+1}\}$ and may not be $(W+1)$-heavy in $\Lambda(G)$. In other words, at most $(W+1)\cdot q$ vertices are W-light in $\Lambda(G)$. □

Lemma 5. *For any $q \in \{0, 1, \ldots, n\}$, if the maximum directed flow from vertex s to vertex t in $\mathcal{N}_G(q)$ is strictly less than $(W+1)\cdot n$ then the number of W-light vertices in every orientation of G is strictly larger than q.*

Proof. We prove the contrapositive. Suppose Λ is an orientation of G with at most q W-light vertices. Construct a flow from s to t in $\mathcal{N}_G(q)$ with value $(W+1)\cdot n$ as follows. Use all of the capacity of edges of the form (s, v), (e, z), (z, t), where $v \in V$ and $e \in E$. For each $e = \{u, v\} \in E$, if $\Lambda(e) = (u, v)$ then send one unit of flow along (u, e), and if $\Lambda(e) = (v, u)$ then send one unit of flow along (v, e). Since at most q vertices are W-light, the total flow between vertices in $\mathcal{N}_G$ corresponding to vertices in V and vertices in $\mathcal{N}_G$ corresponding to edges in E is at least $(W+1)\cdot n-(W+1)\cdot q$. Next, for each W-light vertex v, distribute whatever remains of its at most $(W+1)$ units of flow among $\{x_1, x_2, \ldots, x_W, x_{W+1}\}$ arbitrarily by capacity-1 edges of the form (v, x_i). Finally, let the flow along each edge (x_i, r) be the sum of all incoming flows to x_i and let the flow along (r, z) be the sum of all incoming flows to r. In total, the flow from s to t is $(W+1)\cdot n$. □

We now describe the algorithm. For any $q \in \{0, 1, \ldots, n\}$, let $F(q)$ be an integral maximum directed flow from vertex s to vertex t in $\mathcal{N}_G(q)$, as computed by the algorithm of Goldberg and Rao [13].[5]

1. Construct $\mathcal{N}_G$.
2. Let $q = 0$.
3. Repeat until the value of the flow $F(q)$ equals $(W+1)\cdot n$:
   $q = q + 1$.
4. For every $e \in E$, if an edge of the form (v_i, e) in $\mathcal{N}_G(q)$ has one unit of flow in $F(q)$ then orient e away from v_i in G. Orient all remaining unoriented edges of G arbitrarily.

Let $\mathcal{A}(G)$ be the number of W-light vertices in the orientation constructed by the algorithm. During the execution of the algorithm, at some point p, the following situation occurs: (a) $q = p-1$ implies that the maximum flow in $\mathcal{N}_G(q) < (W+1)\cdot n$, and (b) $q = p$ implies that the maximum flow in $\mathcal{N}_G(q) = (W+1)\cdot n$. By Lemma 5, (a) means that $OPT(G) > p-1$, i.e., $OPT(G) \geq p$. By Lemma 4, (b) gives $\mathcal{A}(G) \leq (W+1)\cdot p$. It follows that the approximation ratio is at most $\frac{(W+1)\cdot p}{p} = W+1$. We have just shown:

Theorem 7. *For any $W \geq 1$,* Minimize W-Light *can be approximated within a ratio of $(W+1)$ in polynomial time.*

For greater efficiency, use a binary search on q in Step 3 instead of checking all candidate values of q incrementally. This gives an $O(n^3 \log^3 n)$-time algorithm.

[5] Since all edge capacities are integers, we may assume by the integrality theorem (see [7]) that the flow along each edge in $F(q)$ found by the algorithm in [13] is an integer.

4 Concluding Remarks

Our results are summarized in the following two tables:

Sect. 2	MAXIMIZE W-LIGHT	MINIMIZE $(W+1)$-HEAVY
$W = 0$	Equivalent to MAXIMUM INDEPENDENT SET (Theorem 1)	Equivalent to MINIMUM VERTEX COVER (Theorem 1)
$W \geq 1$	Solvable in $O(n)$ time for trees (Theorem 2)	Solvable in $O(n)$ time for trees (Theorem 2)

Sect. 3	MINIMIZE W-LIGHT	MAXIMIZE $(W+1)$-HEAVY
$W = 0$	Solvable in $O(m^{3/2})$ time (Theorem 3)	Solvable in $O(m^{3/2})$ time (Theorem 3)
$W \geq 0$	Solvable in $O(n)$ time for outerplanar graphs (Theorem 4)	Solvable in $O(n)$ time for outerplanar graphs (Theorem 4)
$W \geq 1$	Polynomial-time $(W+1)$-approx. (Theorem 7)	Polynomial-time $(W+2)$-approx. (Theorem 6) and $O(n^2)$-time 2-approx. for planar graphs (Theorem 4)
$W \geq 2$	NP-hard even for planar graphs (Theorem 5)	NP-hard even for planar graphs (Theorem 5)

Letting the parameter W vary yields a new generalization of MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER. One interpretation of MINIMIZE $(W+1)$-HEAVY as a MINIMUM VERTEX COVER problem is that every vertex in V is allowed to cover W or less of its incident edges in G "for free", without having to be placed in the output vertex cover V'.

The results derived in this paper did not rely on any advanced techniques. The main contribution of this paper has been to introduce the four new graph orientation problems and to show how they extend the fundamental problems MAXIMUM INDEPENDENT SET and MINIMUM VERTEX COVER in a novel way. We hope that this paper will inspire more sophisticated approximation algorithms and hardness results in the near future. Indeed, the two tables above expose several open problems:

- What is the computational complexity of MAXIMIZE W-LIGHT and MINIMIZE $(W+1)$-HEAVY when $W \geq 1$?
- In particular, MAXIMIZE 0-LIGHT is equivalent to MAXIMUM INDEPENDENT SET and therefore already extremely hard, so does MAXIMIZE W-LIGHT become easier when W gets larger?
- What is the computational complexity of MINIMIZE W-LIGHT and MAXIMIZE $(W+1)$-HEAVY for the special case $W = 1$?
- Can Theorem 5 be strengthened to give non-trivial polynomial-time inapproximability bounds for MINIMIZE W-LIGHT and MAXIMIZE $(W+1)$-HEAVY?

- Is it possible to refine our polynomial-time approximation algorithms for MINIMIZE W-LIGHT and MAXIMIZE $(W+1)$-HEAVY to improve the approximation ratios in Theorems 6 and 7?

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., 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)
4. Bansal, N., Sviridenko, M.: The Santa Claus problem. In: Proc. of STOC 2006, pp. 31–40. ACM (2006)
5. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications (1999)
6. Brodal, G.S., Fagerberg, R.: Dynamic Representations of Sparse Graphs. In: Dehne, F., Gupta, A., Sack, J.-R., Tamassia, R. (eds.) WADS 1999. LNCS, vol. 1663, pp. 342–351. Springer, Heidelberg (1999)
7. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT Press, Massachusetts (1990)
8. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. Annals of Mathematics 162(1), 439–485 (2005)
9. Feige, U.: Approximating maximum clique by removing subgraphs. SIAM Journal on Discrete Mathematics 18(2), 219–225 (2004)
10. Garey, M., Johnson, D.: Computers and Intractability – A Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
11. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM Journal on Computing 1(2), 180–187 (1972)
12. Gavril, F.: Testing for equality between maximum matching and minimum node covering. Information Processing Letters 6(6), 199–202 (1977)
13. Goldberg, A.V., Rao, S.: Beyond the flow decomposition barrier. Journal of the ACM 45(5), 783–797 (1998)
14. Heath, L.S., Vergara, J.P.C.: Edge-packing in planar graphs. Theory of Computing Systems 31(6), 629–662 (1998)
15. Hopcroft, J.E., Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing 2(4), 225–231 (1973)
16. Karakostas, G.: A better approximation ratio for the vertex cover problem. ACM Transactions on Algorithms 5(4), Article 41 (2009)
17. Karp, R.M.: Reducibility among combinatorial problems. In: Proc. of Complexity of Computer Computations. The IBM Research Symposia Series, pp. 85–103. Plenum Press (1972)
18. Schrijver, A.: Combinatorial Optimization. Springer (2003)
19. Zuckerman, D.: Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. Theory of Computing 3(1), 103–128 (2007)