Graph orientation with splits

Graph Orientation with Splits[☆]

Yuichi Asahiro[a], Jesper Jansson[b], Eiji Miyano[c], Hesam Nikpey[d], Hirotaka Ono[e]

[a] *Department of Information Science, Kyushu Sangyo University, Fukuoka, Japan*
[b] *Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[c] *Department of Artificial Intelligence, Kyushu Institute of Technology, Iizuka, Japan*
[d] *Department of Computer Engineering, Sharif University of Technology, Tehran, Iran*
[e] *Graduate School of Informatics, Nagoya University, Nagoya, Japan*

Abstract

The *Minimum Maximum Outdegree Problem* (MMO) is to assign a direction to every edge in an input undirected, edge-weighted graph so that the maximum weighted outdegree taken over all vertices becomes as small as possible. In this paper, we introduce a new variant of MMO called the *p-Split Minimum Maximum Outdegree Problem* (p-Split-MMO) in which one is allowed to perform a sequence of p split operations on the vertices before orienting the edges, for some specified non-negative integer p, and study its computational complexity.

Keywords: graph orientation, maximum flow, vertex cover, partition, algorithm, computational complexity.
2010 MSC: 68Q25, 68W40, 05C70, 05C85

1. Introduction

An *orientation* of an undirected graph is an assignment of a direction to each of its edges. The computational complexity of constructing graph orientations that optimize various criteria has been studied, e.g., in [2–5, 7, 8, 10, 12, 15, 17],

[☆]A preliminary version of this paper appeared in Proceedings of ISCO 2018, Vol. 10856 of Lecture Notes in Computer Science, Springer, 2019, pp.52–63.

Email addresses: asahiro@is.kyusan-u.ac.jp (Yuichi Asahiro), jesper.jansson@polyu.edu.hk (Jesper Jansson), miyano@ces.kyutech.ac.jp (Eiji Miyano), hnikpey@ce.sharif.edu (Hesam Nikpey), ono@i.nagoya-u.ac.jp (Hirotaka Ono)

and positive as well as negative results are known for many variants of these problems.

For example, the *Minimum Maximum Outdegree Problem* (MMO) [5, 8–10, 17] takes as input an undirected, edge-weighted graph $G = (V, E, w)$, where V, E, and w denote the set of vertices of G, the set of edges of G, and an edge-weight function $w : E \to \mathbb{Z}^+$, respectively, and asks for an orientation of G that minimizes the resulting maximum weighted outdegree taken over all vertices in the oriented graph. In general, MMO is strongly NP-hard and cannot be approximated within a ratio of $3/2$ unless P $=$ NP [5]. However, in the special case where all edges have weight 1, MMO can be solved exactly in polynomial time [17]. MMO has applications to load balancing, resource allocation, and data structures for fast vertex adjacency queries in sparse graphs [9, 10] based on the technique of placing each edge in the adjacency list of exactly one of its two incident vertices. For example, if G is a planar graph then G admits an orientation in which every vertex has outdegree at most 3 and such an orientation can be found in linear time [10]; this means that for a planar graph, after linear-time preprocessing, any adjacency query can be answered in $O(1)$ time. As an additional example of a graph orientation problem, finding an orientation that maximizes the number of vertices with outdegree 0 is the Maximum Independent Set Problem [3], which cannot be approximated within a ratio of n^{ϵ} for any constant $0 \leq \epsilon < 1$ in polynomial time unless P $=$ NP [18]. Similarly, finding an orientation that minimizes the number of vertices with outdegree at least 1 is the Minimum Vertex Cover Problem and minimizing the number of vertices with outdegree at least 2 is the problem of finding a smallest subset of the vertices in G whose removal leaves a pseudoforest [3], both of which admit polynomial-time 2-approximation algorithms [13].

In this paper, we introduce a new variant of MMO called the *p-Split Minimum Maximum Outdegree Problem* (p-Split-MMO), where p is a specified non-negative integer, and study its computational complexity. Here, one is allowed to perform a sequence of *p split operations* on the vertices before orienting the edges. When thinking of MMO as a load balancing problem, the split operation can be interpreted as a way to alleviate the burden on the existing machines by adding an extra machine.

The paper is organized as follows. Section 2 gives the formal definition of p-Split-MMO. In Section 3, we show the obtained results on unweighted graphs: Section 3.1 presents an $O((n+p)^p \cdot poly(n))$-time algorithm for the unweighted case of the problem, where n is the number of vertices in the input graph, while Section 3.2 proves that if p is unbounded then the problem becomes NP-hard even in the unweighted case. Section 4 shows the intractability on the edge-weighted case: Section 4.1 shows that p-Split-MMO on edge-weighted wheel graphs is weakly NP-hard even if restricted to $p = 1$. As another graph class, we show strong NP-hardness of p-Split-MMO on edge-weighted cactus graphs when $p = \Omega(n)$ in Section 4.2. Finally, Section 4.3 proves that p-Split-MMO on edge-weighted planar bipartite graphs is also strong NP-hard even with $p = 1$. See Table 1 for a summary of the new results.

Table 1: Overview of the computational complexity of p-Split-MMO. Note that in the edge-weighted case, the edge weights are included in the input so it is possible to further classify the NP-hardness results as either weakly NP-hard or strongly NP-hard.

	Unweighted graphs	Edge-weighted graphs
Constant p	$O((n+p)^p \cdot poly(n))$ time (Section 3.1, Theorem 3)	Weakly NP-hard for wheel graphs (Section 4.1, Theorem 8) Strongly NP-hard for planar bipartite graphs (Section 4.3, Theorem 12)
Unbounded p	NP-hard (Section 3.2, Corollary 6)	Strongly NP-hard for cactus graphs (Section 4.2, Theorem 10)

2. Definitions

Let $G = (V, E, w)$ be an undirected, edge-weighted graph with vertex set V, edge set E, and edge weights defined by the function $w : E \to \mathbb{Z}^+$. An *orientation* Λ *of* G is an assignment of a direction to every edge $\{u, v\} \in E$, i.e., $\Lambda(\{u, v\})$ is either (u, v) or (v, u). For any orientation Λ of G, the *weighted outdegree* of a vertex u is

$$d^+_\Lambda(u) = \sum_{\substack{\{u,v\}\in E:\\ \Lambda(\{u,v\})=(u,v)}} w(\{u, v\})$$

and the *cost* of Λ is

$$c(\Lambda) = \max_{u\in V}\{d^+_\Lambda(u)\}.$$

Let MMO be the following optimization problem, previously studied in [5, 8–10, 17].

The Minimum Maximum Outdegree Problem (MMO):

Given an undirected, edge-weighted graph $G = (V, E, w)$, where V, E, and w denote the set of vertices of G, the set of edges of G, and an edge-weight function $w : E \to \mathbb{Z}^+$, output an orientation Λ of G with minimum cost.

Next, for any $v \in V$, the set of vertices in V that are neighbors of v is denoted by $\Gamma[v]$ and the set of edges incident to v is denoted by $E[v]$. A *split operation* on a vertex v_i in G is an operation that transforms: (i) the vertex set of G to $(V \setminus v_i) \cup \{v_{i,1}, v_{i,2}\}$, where $v_{i,1}$ and $v_{i,2}$ are two new vertices; and (ii) the edge set of G to $(E \setminus E[v_i]) \cup \{\{v_{i,1}, s\} : s \in S\} \cup \{\{v_{i,2}, s'\} : s' \in \Gamma[v_i] \setminus S\}$ for some subset $S \subseteq \Gamma[v_i]$. For any non-negative integer p, a *p-split* on G is a sequence of p split operations successively applied to G. Note that in a p-split, a new vertex resulting from a split operation may in turn be the target of a later split operation.

The problem that we study in this paper generalizes MMO above and is defined as follows for any non-negative integer p.

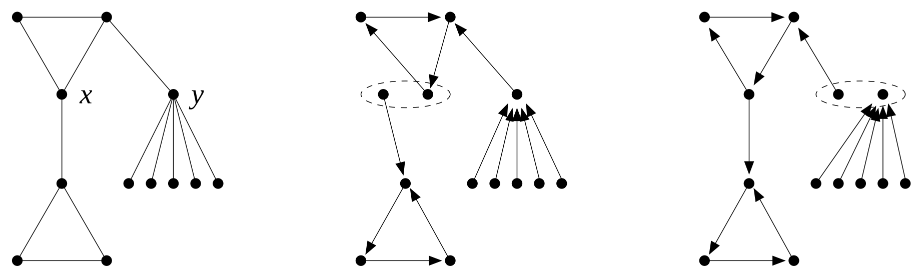

Figure 1: Consider the instance of 1-Split-MMO on the left (here, all edge weights are 1). If the split operation is applied to the vertex x as shown in the middle figure, the resulting instance of MMO can be oriented with maximum outdegree equal to 1, so this is an optimal solution. Observe that if the vertex y had been split instead, the minimum maximum outdegree would have been 2. This shows that greedily applying the split operations to the highest degree nodes will not necessarily yield an optimal solution.

The p-Split Minimum Maximum Outdegree Problem (p-Split-MMO):

Given an undirected, edge-weighted graph $G = (V, E, w)$, where V, E, and w denote the set of vertices of G, the set of edges of G, and an edge-weight function $w : E \to \mathbb{Z}^+$, output a graph G' and an orientation Λ' of G' such that: (i) G' is obtained by a p-split on G; (ii) Λ' has minimum cost among all orientations of all graphs obtainable by a p-split on G.

See Fig. 1 for an example. Throughout the paper, we denote the number of vertices and edges in the input graph G by n and m, respectively. Any orientation of a graph G', where G' can be obtained by applying a p-split to G, will be referred to as a *p-split orientation of G*. The decision version of p-Split-MMO, denoted by p-Split-MMO(W), asks whether or not the input graph G has a p-split orientation Λ' with $c(\Lambda') \leq W$ for a specified integer W. An algorithm `ALG` is called a σ-approximation algorithm if $\frac{ALG(G)}{OPT(G)} \leq \sigma$ for every input graph G, where $ALG(G)$ and $OPT(G)$ are the costs of the orientations obtained by `ALG` and an optimal algorithm. Then, we say that it is NP-hard to approximate within a factor of σ when there is no polynomial-time σ-approximation algorithm unless $P = NP$.

3. Unweighted graphs

3.1. An algorithm for unweighted graphs

This section presents an algorithm for p-Split-MMO on graphs with unweighted edges (equivalently, where all edge weights are equal to 1). Its time complexity is $O((n+p)^p \cdot poly(n))$, which is polynomial when $p = O(1)$.

Our basic strategy is to transform p-Split-MMO to the maximum flow problem on directed networks with edge capacities: (i) We first select an integer W as an upper bound on the cost of a p-split orientation. (ii) Next, we construct a flow network $\mathcal{N}$ based on the input graph G and the integer W. (iii) By computing a maximum network flow in $\mathcal{N}$, we solve p-Split-MMO(W), i.e., determine whether p-Split-MMO(W) admits a feasible solution or not. (iv) By refining W according to a binary search while repeating steps (ii) and (iii), we find the minimum possible value of W and retrieve an optimal p-split orientation of G from the corresponding flow network.

We now describe the details. (Refer to Fig. 2 for an example of the construction.) Let $G = (V, E)$ be the input graph and p any non-negative integer. For any positive integer W and multisubset S of V (i.e., a subset of V in which repetitions are allowed) of cardinality p, define the following flow network $\mathcal{N}_{W,S} = (V_{\mathcal{N}}, E_{\mathcal{N}})$:

$$\begin{aligned} V_{\mathcal{N}} &= V \cup E \cup \{s, t\} \\ E_{\mathcal{N}} &= \bigcup_{e=\{u,v\} \in E} \{(s,e),\, (e,u),\, (e,v)\} \cup \bigcup_{v \in V} \{(v,t)\} \end{aligned}$$

where s and t are newly created vertices. Note that $|V_{\mathcal{N}}| = n + m + 2$ and $|E_{\mathcal{N}}| = n + 3m$. The capacity $cap(u, v)$ of each edge $(u, v) \in E_{\mathcal{N}}$ is set to:

- $cap(s, e) = 1$ for every $e \in E$;
- $cap(e, u) = cap(e, v) = 1$ for every $e = \{u, v\} \in E$; and
- $cap(v, t) = W + W \cdot occ(v)$ for every $v \in V$, where $occ(v)$ is defined as the number of occurrences of v in S.

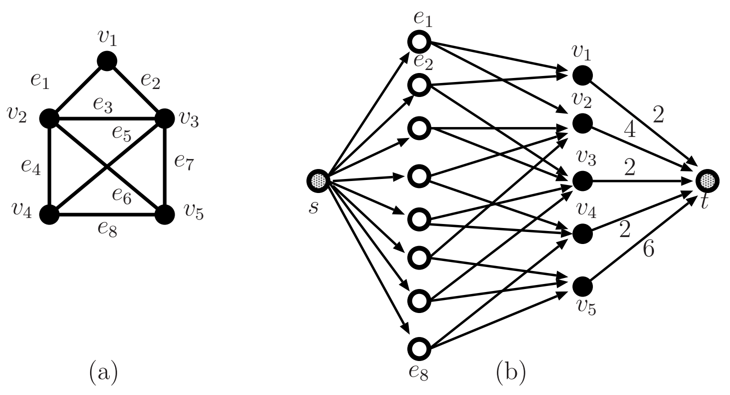

Figure 2: (a) An input graph G and (b) the flow network $\mathcal{N}_{W,S}$ constructed from G when $p = 3$, $W = 2$, and $S = \{v_2, v_5, v_5\}$. For clarity, only edge capacities in $\mathcal{N}_{W,S}$ greater than 1 are displayed.

Consider any maximum flow in $\mathcal{N}_{W,S}$. Since the edge capacities are integers, we can assume that the maximum flow is integral by the integrality theorem (see, e.g., [11]). Then we have:

Lemma 1. *The maximum directed flow from vertex s to vertex t in $\mathcal{N}_{W,S}$ equals $|E|$ if and only if G has a p-split orientation with cost at most W obtained after doing $occ(v)$ split operations on each $v \in V$.*

Proof. ($\Rightarrow$) Let F be a maximum directed flow from s to t with integer values and assume it is equal to $|E|$. Since there are $|E|$ units of flows leaving s in F, exactly one edge among (e, u) and (e, v) for every $e = \{u, v\} \in E$ has one unit of flow in $\mathcal{N}_{W,S}$. We construct a p-split orientation Λ of G by first orienting each edge $e = \{u, v\} \in E$ as (u, v) if (e, u) is using one unit of flow in F and (e, v) is using zero units of flow in F, or as (v, u) otherwise. At this point, each vertex $v \in V$ has outdegree at most $W + W \cdot occ(v)$ because there are at most this many units of flow entering v in $\mathcal{N}_{W,S}$. Next, for each $v \in V$, do $occ(v)$ split operations on v and distribute its outgoing edges evenly among each v and its resulting new vertices so that every vertex has outdegree at most W. Since $\sum_{v \in V} occ(v) = p$, the resulting Λ is a p-split orientation of G with cost at most

W.

($\Leftarrow$) Suppose there is a p-split orientation of G with cost at most W obtained by doing $occ(v)$ split operations on each $v \in V$. Then we can construct a flow in $\mathcal{N}_{W,S}$ that has $|E|$ units of flow by using: (i) all $|E|$ edges of the form (s, e); (ii) $|E|$ edges of the form (e, u) where $e = \{u, v\} \in E$ (either (e, u) or (e, v) depending on if $\{u, v\}$ was oriented as (u, v) or (v, u)); and (iii) at most $|V|$ edges of the form (v, t). Observe that for (iii), each $v \in V$ has at most $W + W \cdot occ(v)$ units of flow entering it in $\mathcal{N}_{W,S}$, which is within the capacity limit of its outgoing edge (v, t), so in total, we have $|E|$ units of flow from s to t. □

The next lemma describes the proposed algorithm.

Lemma 2. p-Split-MMO *can be solved in* $O((n+p)^p \cdot n^2 \cdot T(|V_\mathcal{N}|, |E_\mathcal{N}|) \cdot \log n)$ *time, where* $T(|V_\mathcal{N}|, |E_\mathcal{N}|)$ *is the running time for solving the maximum network flow problem on a directed graph with vertex set* $V_\mathcal{N}$ *and edge set* $E_\mathcal{N}$*.*

Proof. For any candidate value of W, we can identify a p-split orientation of G with cost at most W or determine that none exists, by evaluating every multisubset S of V of cardinality p, constructing $\mathcal{N}_{W,S}$, computing a maximum directed flow in $\mathcal{N}_{W,S}$, and applying Lemma 1. The number of multisubsets is at most $\binom{n-1+p}{p} = O((n+p)^p)$, constructing each $\mathcal{N}_{W,S}$ takes $O(n+m) = O(n^2)$ time, and each maximum network flow instance is solved in $T(|V_\mathcal{N}|, |E_\mathcal{N}|)$ time.

Since the graph G is unweighted, W is upper-bounded by the maximum degree of a vertex. Therefore, applying binary search to obtain the minimum possible value of W (i.e., the smallest W for which the maximum flow is still $|E|$ for some multisubset S of V) increases the running time by a factor of $O(\log n)$. The total time complexity is $O((n+p)^p \cdot n^2 \cdot T(|V_\mathcal{N}|, |E_\mathcal{N}|) \cdot \log n)$. □

Since $|V_\mathcal{N}| = O(m)$ and $|E_\mathcal{N}| = O(m)$, plugging in $T(|V_\mathcal{N}|, |E_\mathcal{N}|) = O(m^2)$ (see [16]) yields:

Theorem 3. p-Split-MMO *for unweighted graphs can be solved in* $O((n+p)^p \cdot n^2 m^2 \log n)$ *time.*

3.2. Intractability for unbounded p

We now prove the NP-hardness of p-Split-MMO for unbounded p, even when restricted to unweighted graphs. Recall that p-Split-MMO(W) is the decision version of p-Split-MMO which asks if G has a p-split orientation of cost at most W. The main result of this section is:

Theorem 4. p-Split-MMO(3) *for unweighted graphs and unbounded p is NP-complete.*

Proof. p-Split-MMO(3) is in NP because a nondeterministic algorithm can guess a p-split of G and an orientation of the resulting graph in polynomial time and check if this orientation has cost at most 3.

To prove the NP-hardness, we give a polynomial-time reduction from the decision version of the Minimum Vertex Cover Problem, VC(k), defined as: Given an undirected graph $G = (V, E)$ and a positive integer k, determine if there is a subset $V' \subseteq V$ with $|V'| \leq k$ such that for each $\{u, v\} \in E$, at least one of u and v belongs to V'. It is known that VC(k) remains NP-complete even if restricted to graphs of degree at most three [14].

The reduction is as follows. (See Fig. 3 for an example.) Suppose we are given an instance $G = (V, E)$ of VC(k), where G has degree at most three. Write $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. We construct an instance G' of p-Split-MMO(3) by defining: (i) a set $U = \{u_1, u_2, \ldots, u_n\}$ of n vertices, where each u_i corresponds to $v_i \in V$; and (ii) a set $W = \{w_1, w_2, \ldots, w_m\}$ of m vertices, where each w_j corresponds to $e_j \in E$. In addition, we prepare: (iii) $n + m$ complete graphs with six vertices each, denoted by G_1^V through G_n^V and G_1^E through G_m^E. Let $V(G_i^V) = \{u_{i,1}, u_{i_2}, \ldots, u_{i,6}\}$ for each $i \in \{1, 2, \ldots, n\}$ and $V(G_j^E) = \{w_{j,1}, w_{j_2}, \ldots, w_{j,6}\}$ for each $j \in \{1, 2, \ldots, m\}$. The vertex set of G' is thus $U \cup W \cup V(G_1^V) \cup V(G_2^V) \cup \cdots \cup V(G_n^V) \cup V(G_1^E) \cup V(G_2^E) \cup \cdots \cup V(G_m^E)$. Next, insert the following edges into the edge set of G' (which already includes the edges of G_1^V through G_n^V and G_1^E through G_m^E): (iv) edges $\{u_h, w_j\}$ and $\{u_i, w_j\}$ if $e_j = \{v_h, v_i\} \in E$ for each $j \in \{1, 2, \ldots, m\}$; (v) an edge $\{u_i, u_{i,h}\}$

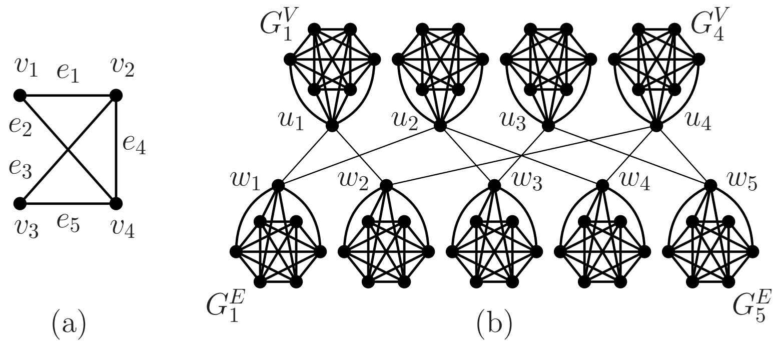

Figure 3: Illustrating the reduction from VC(k) to p-Split-MMO(3). (a) An instance of VC(k) with four vertices and five edges. (b) The instance of p-Split-MMO(3) constructed from (a).

for each $i \in \{1, 2, \dots, n\}$ and each $h \in \{1, 2, \dots, 6\}$; and (vi) an edge $\{w_j, w_{j,h}\}$ for each $j \in \{1, 2, \dots, m\}$ and each $h \in \{1, 2, \dots, 5\}$. Note that each u_i in G' has degree equal to (6 + the degree of v_i in G) and every w_j in G' has degree 7. Finally, we set $p = k$. This completes the reduction.

Next, we show that G has a vertex cover with size at most p if and only if G' has a p-split orientation whose cost is at most three.

($\Rightarrow$) Suppose that G has a vertex cover C of size p. Let $C' \subseteq U$ be the p vertices in G' that correspond to vertices in C. Apply a split operation on each $u_i \in C'$ to transform it into a pair of vertices u_i and u_i^*, the first one (u_i) being adjacent to all six vertices from G_i^V and the second one (u_i^*) being adjacent to the at most three neighbors from W. Let G'' be the resulting graph. By definition, G'' is obtained by applying a p-split to G' and we will now show that G'' admits an orientation of cost three.

First, every G_i^V forms a K_7 (a complete graph with seven vertices) together with u_i in G''. Orient each such K_7 so that all of its vertices have outdegree three, e.g., by applying Proposition 2 in [4]. Secondly, orient the (at most three) edges incident to each u_i^*-vertex away from u_i^*. Since C is a vertex cover, every w_j-vertex in G' will be incident to at most one unoriented edge of the form $\{u_i, w_j\}$ after this step is done. Next, for each w_j, if there is one unoriented edge of the form $\{u_i, w_j\}$ then orient it away from w_j. Finally, every w_j and G_j^E

form a K_7 with one edge incident to w_j missing; orient this subgraph as above, but let w_j have one less outgoing edge than the other vertices so that the outdegree of each such vertex is at most three. This yields an orientation of G'' of cost three.

($\Leftarrow$) Suppose G' has a p-split orientation of cost at most three. If some vertex $u_{i,h}$ in G_i^V was split then we obtain another p-split orientation of cost at most three by not splitting $u_{i,h}$ but splitting u_i instead and orienting the edges of the resulting K_7 as described above, and similarly for vertices in G_j^E. We may therefore assume that every vertex that is split comes from $U \cup W$. Next, if some vertex w_j in W is split and it has an incident u_i-vertex that is not split then we replace the split operation on w_j by a split operation on u_i; by doing so and orienting the edge between u_i and w_j towards w_j, the cost of the orientation will not increase. This produces a p-split orientation of G' in which every vertex from W is incident to at least one vertex from the set of (at most p) vertices from U that were split, which then gives a vertex cover of G of size at most p. □

The above proof also gives an inapproximability result for p-Split-MMO:

Corollary 5. *For any constant $\varepsilon > 0$, it is NP-hard to approximate* p-Split-MMO *to within a factor of $\frac{4}{3} - \varepsilon$, even for unweighted graphs.*

Proof. In the reduction in the proof of Theorem 4, there always exists a p-split orientation Λ' of G' satisfying $c(\Lambda') \leq 4$, as can be seen by ignoring all available split operations and just orienting the at most two edges of the form $\{u_i, w_j\}$ for each w_j away from w_j and all other edges as in the first part of the proof of Theorem 4. Since there exists a p-split orientation Λ' with $c(\Lambda') \leq 3$ if and only if the given instance of VC(k) has a vertex cover with size at most k, the above reduction is a gap-introducing one, i.e., if there existed a polynomial-time $(\frac{4}{3} - \varepsilon)$-approximation algorithm for p-split-MMO(3), then VC(k) could be solved in polynomial time. □

A similar reduction as in the proof of Theorem 4 also gives the following corollary. Instead of attaching complete graphs of size six to the u_i- and w_j-

vertices as in the proof of Theorem 4, we attach complete graphs of size $2W$. Furthermore, in the construction, we prepare $2W$ edges of type (v) edges between each u_i-vertex and its complete graph and $2W - 1$ edges of type (vi) between each w_j-vertex and its complete graph. The edges of type (iv) are defined as before. Based on these, we can show that G has a vertex cover with size at most p if and only if G' has a p-split orientation whose cost is at most W for every fixed integer $W \geq 3$.

Corollary 6. *For every fixed integer $W \geq 3$, p-Split-MMO(W) for unweighted graphs and unbounded p is NP-complete.*

The above theorem shows the NP-completeness for $W \geq 3$. So the remaining question here is about the computational complexity of p-Split-MMO(1) and p-Split-MMO(2) for unweighted graphs and unbounded p. We partially answer this question by explaining how to solve p-Split-MMO(1): First we find a pseudotree in the given input graph, then orient the edges of its cycle in one direction. Then we orient other edges in the pseudotree towards the cycle. After that, the remaining edges in the input graph are oriented arbitrarily. Finally denote the resulting orientation by Λ and apply $d^+_\Lambda(v) - 1$ split operations to each vertex v in such a way that every vertex in the new graph gets exactly one outgoing edge. This gives the minimum number $m - n$ of splits since the numbers of vertices and edges are n and m, respectively, and the target outdegree W is one. In fact, to solve p-Split-MMO(1), it is not necessary to actually construct an orientation; one can just check whether or not $p \geq m - n$.

Theorem 7. p-Split-MMO(1) *for unweighted graphs and unbounded p is solvable in linear time.*

The computational complexity of p-Split-MMO(2) is still unknown.

4. Edge-weighted graphs

4.1. Wheel graphs

In this section, we prove that p-Split-MMO on edge-weighted wheel graphs is weakly NP-hard. To do so, we give a polynomial-time reduction from the Partition Problem, defined as follows: Given a set $S = \{s_1, s_2, \ldots, s_n\}$ of n positive integers, determine if there exists a subset $S' \subseteq S$ such that $\sum_{s_i \in S'} s_i = \sum_{s_j \in S \setminus S'} s_j$. The Partition Problem is weakly NP-hard and admits a pseudopolynomial-time solution [14].

Theorem 8. *For every fixed integer $p \geq 1$,* p-Split-MMO *on edge-weighted graphs is weakly NP-hard even if the input is restricted to wheel graphs.*

Proof. We construct an edge-weighted, undirected wheel graph $G = (V, E, w)$ from any given instance $S = \{s_1, s_2, \ldots, s_n\}$ of the Partition Problem. Define $K = \sum_{i=1}^{n} s_i/2$ and assume without loss of generality that $s_i \leq K$ for all $s_i \in S$. The vertex set V consists of: (i) n vertices representing the integers in S and denoted by $v_1, v_2, \ldots, v_n$; (ii) auxiliary $p-1$ vertices $v_{n+1}, \ldots, v_{n+p-1}$, where there is no vertex of this type in the case $p = 1$; and (iii) one special vertex, denoted by v_c. Let $N = n+p-1$, i.e., $|V| = N+1$. The edge set E consists of: (iv) the N edges $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\ldots$, $\{v_N, v_1\}$ forming a cycle; (v) the n edges $\{v_c, v_1\}$, $\{v_c, v_2\}$, $\ldots$, $\{v_c, v_n\}$; and (vi) the $p-1$ edges $\{v_c, v_{n+1}\}$, $\ldots$, $\{v_c, v_N\}$, where there is no edge of this type in the case of $p = 1$. Hence, $N+1$ vertices $v_c, v_1, \ldots, v_N$ and N edges $\{v_c, v_1\}, \ldots, \{v_c, v_N\}$ forms a star, and so G is a wheel graph. For every edge e of type (iv), assign $w(e) = K$. For every edge of type (v), assign $w(\{v_c, v_i\}) = s_i$ for $1 \leq i \leq n$. For every edge of type (vi), assign $w(\{v_c, v_i\}) = K$ for $n+1 \leq i \leq N$. An example is shown in Figure 4.

Below, we show that the answer to the given instance S of the Partition Problem is yes if and only if G has a p-split orientation whose cost is at most K.

($\Rightarrow$) Assume that there exists an $S' \subseteq S$ such that $\sum_{s_i \in S'} s_i = \sum_{s_j \in S \setminus S'} s_j$. We repeatedly apply a split operation p times on v_c (or any of the newly constructed vertices obtained by these split operations). Let the resulting vertices $v_{c,1}, \ldots, v_{c,p+1}$ be adjacent to the set of vertices of types (i) and (ii) as follows.

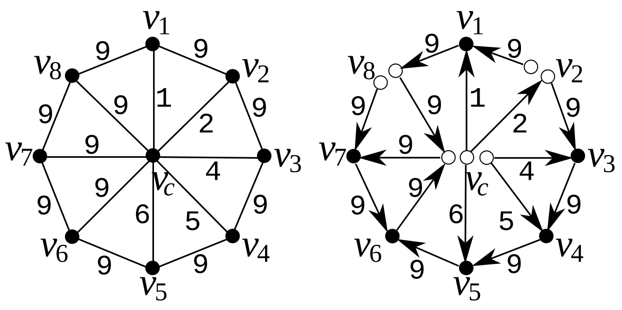

Figure 4: Let $S = \{1, 2, 4, 5, 6\}$ be an instance of the Partition Problem. The reduction in the proof of Theorem 8 sets $K = 9$ and constructs the edge-weighted wheel graph G in the left figure. The right figure shows an optimal 4-split orientation, where the split vertices are represented as unfilled circles.

- $v_{c,1}$ is adjacent to the edges representing S',
- $v_{c,2}$ is adjacent to the edges representing $S \setminus S'$, and
- In the case $p \geq 2$, $v_{c,i}$ is adjacent to v_i for $n + 1 \leq i \leq N$ (remind that in the case $p = 1$, there is no vertex of type (ii)).

For $1 \leq i \leq N$, orient every edge that involves $v_{c,i}$ away from $v_{c,i}$. Orient the remaining N edges so that they form a directed cycle $(v_1, v_2, \ldots, v_N, v_1)$. This way, the weighted outdegree of every vertex is at most K.

($\Leftarrow$) Let Λ be a p-split orientation of G of cost at most K. If S contains a single element equal to K then the answer to the given instance of the Partition Problem is trivially yes. On the other hand, if $s_i < K$ for any $s_i \in S$ then at least one split operation is applied to v_c and all the vertices obtained by applying the split operations to v_c have outdegree K by Λ from Lemma 9 which will be shown later.

Let v_c be replaced with $p - q + 1$ vertices $v_{c,1}, \ldots, v_{c,p-q+1}$. By the above, without loss of generality, q edges $\{v_c, v_{n+1}\}, \ldots, \{v_c, v_{n+q}\}$ of weight K are assumed to be oriented towards v_c. On the other hand, the edges $\{v_c, v_{n+q+1}\}, \ldots,$ $\{v_c, v_{n+p-1}\}$ of weight K and every edge of the form $\{v_c, v_j\}$ of weight s_j for $1 \leq j \leq n$ is oriented away from $v_{c,i}$. Since the cost of Λ is at most K, $p - 1 - q$

vertices, say, $v_{c,3}, \ldots, v_{c,p-q+1}$ are used to orient those $p-1-q$ edges of weight K. Then, since the sum of the remaining weights of edges is $2K$, each of $v_{c,1}$ and $v_{c,2}$ must have weighted outdegree exactly equal to K. Let S' be the set of weights of the edges incident to $v_{c,1}$. Then $\sum_{s_i \in S'} s_i = \sum_{s_j \in S \setminus S'} s_j = K$ and the answer to the given instance of the Partition Problem is yes. □

Finally, we prove Lemma 9, used in the proof of Theorem 8 above.

Lemma 9. *Suppose that Λ is a p-split orientation of G of cost at most K, and $s_i < K$ holds for any $s_i \in S$. At least one split operation is applied to v_c, and all the vertices obtained by applying the split operations to v_c have outdegree K by Λ.*

Proof. First suppose that one split operation was applied to a vertex v_j for $j \in \{1, 2, \ldots, N\}$, thereby replacing v_j by two vertices $v_{j,1}$ and $v_{j,2}$. Each of the N edges not involving v_c has weight K, so at most one of the $N+1$ vertices in $\{v_1, v_2, \ldots, v_N, v_{j,1}, v_{j,2}\} \setminus \{v_j\}$ can orient its edge involving v_c towards v_c. We can consider two cases: (i) the weight of this edge is s_h for some h, and (ii) it is K (note that this case occurs only when $p \geq 2$). For the case (i), the weighted outdegree of v_c is $(p+1)K - s_h > pK$ because $s_i < K$ for all $s_i \in S$. Here the remaining $p-1$ split operations are applied to v_c by which p vertices are newly created from v_c. Then, one of these p vertices must have outdegree greater than $pK/p = K$ according to the pigeonhole principle, contradicting that the cost of Λ is at most K. Namely, if $p = 1$, the split operation must be applied to v_c so that each of the two vertices obtained by the split operation have outdegree K, since the total weights of those edges is $2K$ and Λ has cost K. For the case (ii), the weighted outdegree of v_c is $(p+1)K - K = pK$. Then, the remaining $p-1 > 0$ split operations are applied to v_c and p vertices are constructed. Since the cost of Λ is at most K, all these p vertices must have outdegree K by Λ.

The above discussion can be generalized to the case that $q(\geq 2)$ split operations are applied to other vertices than v_c. Note that in this case, one vertex v_j may be split into three vertices $v_{j,1}$, $v_{j,2}$, and $v_{j,3}$, each of which is adjacent to one another vertex; applying more than three split operations to one vertex in $\{v_1, \ldots, v_N\}$ is useless since a vertex will be independent. As the above, at most q of the N vertices $\{v_1, \ldots, v_N\}$ can orient its edge involving v_c towards v_c. If these q edges include an edge of weight s_h for some h, then the weighted outdegree of v_c is at least $(p+1)K - (q-1)K - s_h > (p-q+1)K$ since $s_h < K$. Then, since $p-q$ split operations will be applied to v_c, at least one vertex of the $p-q+1$ vertices constructed by these split operations must have outdegree greater than $(p-q+1)K/(p-q+1) = K$, again according to the pigeonhole principle. This again contradicts the assumption that the cost of Λ is at most K. Therefore, the q vertices must be chosen from $v_{n+1}, \ldots, v_N (= v_{n+p-1})$, from which we observe that $q \leq p-1$, i.e., at least one split operation is applied to v_c. Then, since the weighted outdegree of v_c is equal to $(p-q+1)K$ after the q split operations, and $p-q+1$ vertices are constructed by $p-q$ split operations to v_c, these vertices must have outdegree K by Λ. $\square$

4.2. Cactus graphs

In the previous section, we showed the weak NP-hardness of p-Split-MMO for wheel graphs with any fixed integer $p \geq 1$. Since the reduction was based on the weak NP-hardness of the Partition problem, we need another reduction to show the strong NP-hardness of the problem. Here, we prove that p-Split-MMO with weighted edges is strongly NP-hard if p is sufficiently large, i.e., $p = \Omega(n)$, and the input is a cactus graph. This result is obtained via a polynomial-time reduction from the 3-Partition Problem: Given a multiset $S = \{s_1, s_2, \ldots, s_{3n}\}$ of positive integers and an integer B such that $B/4 < s_i < B/2$ for every $i \in \{1, 2, \ldots, 3n\}$ and $\sum_{s_i \in S} s_i = n \cdot B$ hold, determine if S can be partitioned into n multisets $S_1, S_2, \ldots, S_n$ so that $|S_j| = 3$ and $\sum_{s_i \in S_j} s_i = B$ for every $j \in \{1, 2, \ldots, n\}$. The 3-Partition Problem is known to be strongly NP-hard [14].

Theorem 10. *For an integer $p = \Omega(n)$, p-*Split-MMO *on edge-weighted graphs is strongly NP-hard even if the input is restricted to cactus graphs.*

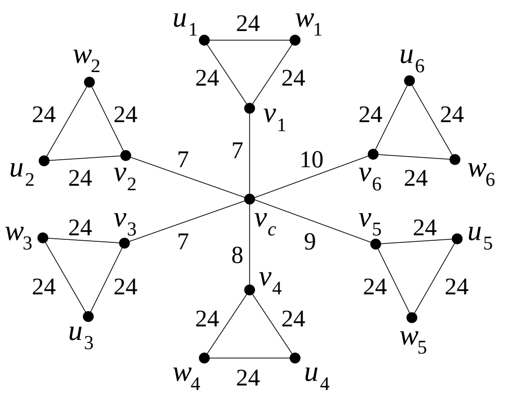

Figure 5: An instance of the 3-Partition Problem with $S = \{7, 7, 7, 8, 9, 10\}$ and $B = 24$ yields the cactus graph G shown above. In the construction, $n = 2$ and $p = 2 - 1 = 1$.

Proof. We construct an edge-weighted, undirected cactus graph $G = (V, E, w)$ from any given instance (S, B) of the 3-Partition Problem, where $S = \{s_1, s_2, \ldots, s_{3n}\}$. Let $p = n - 1$ and recall that $B = \sum_{i=1}^{3n} s_i/n$ by definition. G consists of:

- $3n$ subgraphs, G_1 through G_{3n}, each of which is associated with an element in S. For each $i \in \{1, 2, \ldots, 3n\}$, G_i contains three vertices u_i, v_i, and w_i and three edges $\{u_i, v_i\}$, $\{u_i, w_i\}$, and $\{v_i, w_i\}$ (i.e., G_i is a triangle graph). The weight of every edge in G_i is set to B.
- One special vertex v_c.
- For $i \in \{1, 2, \ldots, 3n\}$, an edge $\{v_c, v_i\}$ of weight s_i that connects G_i to v_c.

The constructed graph is a cactus graph with $9n + 1$ vertices. This completes the description of the reduction. See Fig. 5 for an illustration.

Now we show that the answer to the 3-Partition Problem on input S is yes if and only if the constructed graph G has a p-split orientation of cost B.

($\Rightarrow$) If the answer to the 3-Partition Problem is yes, divide the elements of S into n multisets $S_1, S_2, \ldots, S_n$, where every S_j has the sum B and $|S_j| = 3$. Then, do p split operations on v_c so that each of the resulting $p + 1 = n$ vertices, called *center vertices*, becomes adjacent to exactly three vertices v_x, v_y, and v_z, where $\{s_x, s_y, s_z\}$ is one of the S_j-sets. By orienting all $3n$ edges involving center vertices away from the center vertices, and for each $i \in \{1, 2, \ldots, 3n\}$, orienting the three edges $\{u_i, v_i\}$, $\{u_i, w_i\}$, and $\{v_i, w_i\}$ as (v_i, w_i), (w_i, u_i), and (u_i, v_i), we obtain a p-split orientation of G of cost B.

($\Leftarrow$) Consider any p-split orientation Λ of G with cost B. Let σ be the total number of split operations in this p-split that were done on vertices in the G_i-subgraphs.

First, we show by contradiction that $\sigma = 0$. Suppose $\sigma \geq 1$. If we start from G and apply a sequence of $p - \sigma$ split operations to v_c and the new vertices created by these operations, v_c will be replaced by a set of $p - \sigma + 1 = n - \sigma$ vertices, henceforth denoted by C. Call the $3n$ edges that contain a vertex from C *center edges*. Due to the weights of the edges in each G_i-subgraph, if no split operations are done on u_i, v_i, or w_i then the center edge between v_i and C must be oriented away from C, but each split operation applied to a vertex of the form u_i, v_i, or w_i will allow at most one center edge to become oriented towards C. Let W be the sum of the weights of the center edges that were oriented away from C in Λ. By definition, the weight of every center edge is less than $\frac{B}{2}$, so $W > n \cdot B - \sigma \cdot \frac{B}{2}$. According to the pigeonhole principle, at least one vertex in C must have weighted outdegree at least $W/(n - \sigma)$. However, $W/(n-\sigma) > (n \cdot B - \sigma \cdot \frac{B}{2})/(n-\sigma) > (n \cdot B - \sigma \cdot B)/(n - \sigma) = B$, which is a contradiction because the cost of the p-split orientation Λ was B. Thus, $\sigma = 0$ and $|C| = p + 1 = n$.

Next, note that if a vertex x in C was connected to four or more v_i-vertices then the weighted outdegree of x would be strictly larger than B. Since no vertex in G_i-subgraphs is split and the cost of the p-split orientation Λ is B, the edges in every G_i must be oriented in such a way as to form a directed cycle. This implies that the four or more edges connected to x must be oriented away from x. Each of these edges has weight strictly larger than $\frac{B}{4}$, the weighted outdegree of x would be strictly larger than B, which contradicts the assumption that Λ has cost B.

Finally, since each of the n vertices in C can be connected to at most three v_i-vertices and there are $3n$ v_i-vertices in total, it must be connected to exactly three v_i-vertices and its weighted outdegree is B. Letting the weights of the edges of each such vertex form one S_j-set then gives a partition of S showing that the answer to the 3-Partition Problem is yes. $\square$

We can generalize the above proof to the case p is larger. Let $q = p-(n-1)$. We add q subgraphs G_{3n+1} through G_{3n+q} in addition to the center vertex v_c, the subgraphs G_1 through G_{3n}, and their edges constructed in the above proof. For each $i \in \{1, \ldots, q\}$, G_{3n+i} contains three vertices u_i, v_i, and w_i and three edges $\{u_i, v_i\}$, $\{u_i, w_i\}$, and $\{v_i, w_i\}$ similar to the subgraphs G_1 though G_{3n}. The weight of every edge in G_{3n+i} is also set to B. Then we add an edge $\{v_c, v_{3n+i}\}$ for each $i \in \{1, \ldots, q\}$ of weight B that connects G_{3n+i} to v_c. The resulted graph has $3p+9n-2$ vertices. For this new graph and its optimal p-split, we observe that one split must be applied to G_{3n+i} or v_c for each $i \in \{1, \ldots, q\}$ in order to detach the weight B of the edge $\{v_c, v_{3n+i}\}$ from v_c: If we apply a split to G_{3n+i}, then the edge $\{v_c, v_{3n+i}\}$ is oriented towards v_c, otherwise it is oriented as (v'_c, v_{3n+i}), where v'_c is the vertex newly created from v_c. Then, the remaining $p-q$ splits are applied to the center vertex v_c. Based on this observation, we can show that the answer to the 3-Partition Problem on input S is yes if and only if the constructed graph has a p-split orientation of cost B. Thus we have the following corollary:

Corollary 11. *For an integer $\frac{n}{9} \leq p < \frac{n}{3}$, p-Split-MMO on edge-weighted graphs is strongly NP-hard even if the input is restricted to cactus graphs.*

4.3. Planar bipartite graphs

In the previous section, we showed the strong NP-hardness of p-Split-MMO for cactus graphs when $p = \Omega(n)$. This section gives a simple proof showing that p-Split-MMO on planar bipartite graphs is strongly NP-hard for any fixed integer $p \geq 1$. The next theorem is obtained based on the strong NP-hardness of MMO for planar bipartite graphs [6].

Theorem 12. *For any fixed integer $p \geq 1$, p-Split-MMO on edge-weighted graphs is strongly NP-hard even if the input is restricted to planar and bipartite graphs.*

Proof. As a reduction from the input planar bipartite graph G of MMO to p-Split-MMO, we make $p+1$ copies of G, denoted by $G_1, \ldots, G_{p+1}$. Pick one arbitrary vertex v in the boundary of the outer face of G. Let the vertices corresponding to v in $G_1, \ldots, G_{p+1}$ be $v_1, \ldots, v_{p+1}$. We create new vertices u_i's for $1 \leq i \leq p$, and then insert two edges $\{v_i, u_i\}$ and $\{u_i, v_{i+1}\}$ for $1 \leq i \leq p$. This completes the reduction and the resulting graph G' is clearly planar and bipartite.

The number of split operations we can apply to G' is p. Hence, for a subgraph in G' corresponding to, say, G_1, we cannot apply any split operation. This means that we need to solve MMO for G_1 which has the same structure as G, in order to solve p-Split-MMO. Hence the optimal cost of MMO for G is k if and only if the constructed graph G' has a p-split orientation of cost k. Therefore, p-Split-MMO is also strongly NP-hard even for planar bipartite graphs. $\square$

The proof of Theorem 12 along with the inapproximability bound 1.5 for MMO for edge-weighted planar bipartite graphs in [6] directly gives:

Corollary 13. *For any fixed integer $p \geq 1$, p-Split-MMO cannot be approximated within a ratio of* 1.5 *in polynomial time for planar and bipartite graphs unless P=NP.*

5. Concluding remarks

This paper introduced the p-Split-MMO problem and presented a maximum flow-based algorithm for the unweighted case that runs in polynomial time for any constant p, and proved the NP-hardness of more general problem variants. Future work includes developing polynomial-time approximation algorithms and fixed-parameter tractable algorithms for the NP-hard variants. For example, we only showed a $(4/3 - \varepsilon)$-inapproximability of the unweighted case with unbounded p in this paper. So, one could try to design polynomial-time approximation algorithms for this case.

Another problem is to find the minimum number of split operations that have to be applied to an input unweighted graph so that the resulting graph admits an orientation with maximum outdegree at most W, where W is a fixed integer. As seen in the discussion for Theorem 7, this variant is solvable in polynomial time if $W = 1$. Moreover, for every fixed $W \geq 3$, the problem is APX-hard by Corollary 6 and the APX-hardness of the Minimum Vertex Cover Problem on graphs of degree at most three [1]. However, for $W = 2$, the computational complexity is unknown.

Also, it would be interesting to study how the computational complexity of p-Split-MMO changes if the output orientation is required to be *acyclic* or *strongly connected*. Borradaile *et al.* [8] recently showed that unweighted MMO with either the acyclicity constraint or the strongly connectedness constraint added remains solvable in polynomial time. In contrast, the closely related problem of outputting a *minimum lexicographic orientation* of an input graph, which is solvable in polynomial time for unconstrained orientations, becomes NP-hard for acyclic orientations [8] while its computational complexity for strongly connected orientations is still unknown.

Acknowledgments:

The authors would like to thank an anonymous reviewer of the 5th International Symposium on Combinatorial Optimization (ISCO2018) for providing a simple idea that led to Theorem 12, and also thank Richard Lemence and Edelia Braga for some discussions related to the topic of this paper. This work was partially supported by JSPS KAKENHI Grant Numbers JP17K00016 and JP17K00024, and JST CREST JPMJR1402.

References

[1] Alimonti, P. and Kann, V. (2000). Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1):123 – 134.

[2] Asahiro, Y., Jansson, J., Miyano, E., and Ono, H. (2011a). Graph orientation to maximize the minimum weighted outdegree. *International Journal of Foundations of Computer Science*, 22(3):583–601.

[3] Asahiro, Y., Jansson, J., Miyano, E., and Ono, H. (2015). Graph orientations optimizing the number of light or heavy vertices. *Journal of Graph Algorithms and Applications*, 19(1):441–465.

[4] Asahiro, Y., Jansson, J., Miyano, E., and Ono, H. (2016). Degree-constrained graph orientation: Maximum satisfaction and minimum violation. *Theory of Computing Systems*, 58(1):60–93.

[5] Asahiro, Y., Jansson, J., Miyano, E., Ono, H., and Zenmyo, K. (2011b). Approximation algorithms for the graph orientation minimizing the maximum weighted outdegree. *Journal of Combinatorial Optimization*, 22(1):78–96.

[6] Asahiro, Y., Miyano, E., and Ono, H. (2011c). Graph classes and the complexity of the graph orientation minimizing the maximum weighted outdegree. *Discrete Applied Mathematics*, 159(7):498–508.

[7] Asahiro, Y., Jansson, J., Miyano, E., Ono, H., T.P., Sandhya (2019). Graph orientation with edge modifications. In *Proceedings of FAW 2019*, volume 11458 of *Lecture Notes in Computer Science*, pages 38–50. Springer.

[8] Borradaile, G., Iglesias, J., Migler, T., Ochoa, A., Wilfong, G., and Zhang, L. (2017). Egalitarian graph orientations. *Journal of Graph Algorithms and Applications*, 21(4):687–708.

[9] Brodal, G. S. and Fagerberg, R. (1999). Dynamic representations of sparse graphs. In *Proceedings of WADS 1999*, volume 1663 of *Lecture Notes in Computer Science*, pages 342–351. Springer.

[10] Chrobak, M. and Eppstein, D. (1991). Planar orientations with low outdegree and compaction of adjacency matrices. *Theoretical Computer Science*, 86(2):243–266.

[11] Cormen, T., Leiserson, C., and Rivest, R. (1990). *Introduction to Algorithms*. The MIT Press, Massachusetts.

[12] Deming, R. W. (1979). Acyclic orientations of a graph and chromatic and independence numbers. *Journal of Combinatorial Theory, Series B*, 26:101–110.

[13] Fujito, T. (1998). A unified approximation algorithm for node-deletion problems. *Discrete Applied Mathematics*, 86(2–3):213–231.

[14] Garey, M. and Johnson, D. (1979). *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York.

[15] Khoshkhah, K. (2015). On finding orientations with the fewest number of vertices with small out-degree. *Discrete Applied Mathematics*, 194:163–166.

[16] Orlin, J. B. (2013). Max flows in $O(nm)$ time, or better. In *Proceedings of STOC 2013*, pages 765–774. ACM.

[17] Venkateswaran, V. (2004). Minimizing maximum indegree. *Discrete Applied Mathematics*, 143(1–3):374–378.

[18] Zuckerman, D. (2007). Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(1):103–128.