Graph Orientation with Splits

Yuichi Asahiro[1], Jesper Jansson[2(✉)], Eiji Miyano[3], Hesam Nikpey[4], and Hirotaka Ono[5]

[1] Department of Information Science, Kyushu Sangyo University, Fukuoka, Japan
asahiro@is.kyusan-u.ac.jp
[2] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
jesper.jansson@polyu.edu.hk
[3] Department of Systems Design and Informatics, Kyushu Institute of Technology, Iizuka, Japan
miyano@ces.kyutech.ac.jp
[4] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
hnikpey@ce.sharif.edu
[5] Graduate School of Informatics, Nagoya University, Nagoya, Japan
ono@i.nagoya-u.ac.jp

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

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 [1–5,7,9,12,14], and positive as well as negative results are known for many variants of these problems.

For example, the *Minimum Maximum Outdegree Problem* (MMO) [4–7,14] 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 $w : E \rightarrow \mathbb{Z}^+$, 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 [4]. However, in the special

J. Lee et al. (Eds.): ISCO 2018, LNCS 10856, pp. 52–63, 2018.
https://doi.org/10.1007/978-3-319-96151-4_5

case where all edges have weight 1, MMO can be solved exactly in polynomial time [14]. MMO has applications to load balancing, resource allocation, and data structures for fast vertex adjacency queries in sparse graphs [6,7] based on the technique of placing each edge in the adjacency list of exactly one of its two incident vertices. E.g., 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 [7], which means that for a planar graph, any adjacency query can be answered in $O(1)$ time after linear-time preprocessing. 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 [2], which cannot be approximated within a ratio of n^{ϵ} for any constant $0 \leq \epsilon < 1$ in polynomial time unless P = NP [15]. 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 [2], both of which admit polynomial-time 2-approximation algorithms [10].

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. Section 3 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 Sect. 4 proves that if p is unbounded then the problem becomes NP-hard even in the unweighted case. On the other hand, for the edge-weighted case, Sect. 5 shows that p-Split MMO with weighted edges is weakly NP-hard even if restricted to $p = 1$. Finally, Sect. 6 proves that the most general case of the problem, i.e., with weighted edges as well as unbounded p, is strongly NP-hard. 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 (Sect. 3, Theorem 1)	Weakly NP-hard (Sect. 5, Theorem 3)
Unbounded p	NP-hard (Sect. 4, Theorem 2)	Strongly NP-hard (Sect. 6, Theorem 4)

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 $w : E \rightarrow \mathbb{Z}^+$. 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 [4–7,14].

> 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 $w : E \rightarrow \mathbb{Z}^+$, 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.

> 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 $w : E \rightarrow \mathbb{Z}^+$, 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.

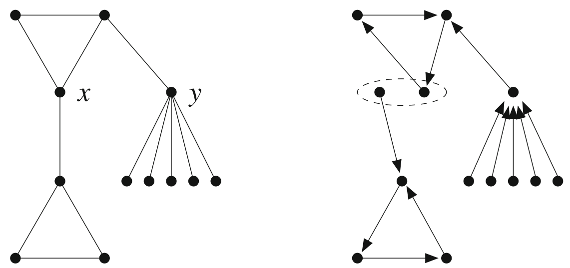

Fig. 1. Consider the instance of 1-Split-MMO on the left (here, all edge weights are 1). If the split operation is applied to the vertex x as shown in the middle figure, the resulting instance of MMO can be oriented with maximum outdegree equal to 1, so this is an optimal solution. Observe that if the 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.

3 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 $\mathcal{N}_{W,S} = (V_{\mathcal{N}}, E_{\mathcal{N}})$:

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

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

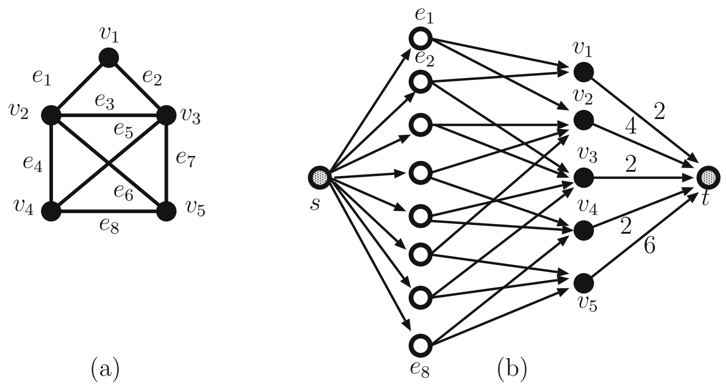

Fig. 2. (a) An input graph G and (b) the flow network $\mathcal{N}_{W,S}$ constructed from G when $p = 3$, $W = 2$, and $S = \{v_2, v_5, v_5\}$. For clarity, only edge capacities in $\mathcal{N}_{W,S}$ greater than 1 are displayed.

- $cap(v,t) = W + W \cdot occ(v)$ for every $v \in V$, where $occ(v)$ is defined as the number of occurrences of v in S.

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., [8]). 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.

($\Leftarrow$) 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. □

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 [13]) yields:

Theorem 1. p-Split-MMO *for unweighted graphs can be solved in* $O((n+p)^p \cdot n^2 m^2 \log n)$ *time.*

4 Unweighted Graphs, 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 2. 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 [11].

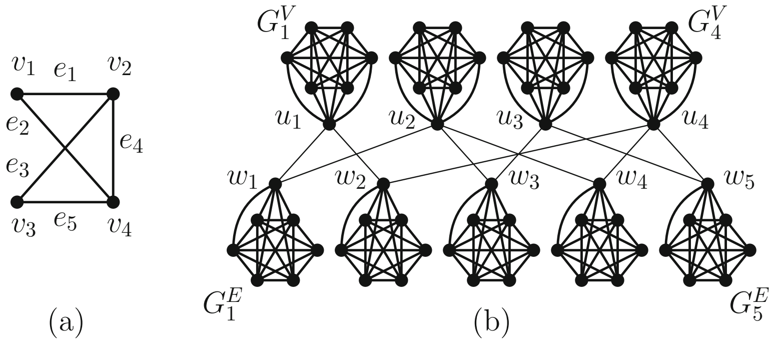

Fig. 3. Illustrating the reduction from VC(k) to p-Split-MMO(3). (a) An instance of VC(k) with four vertices and five edges. (b) The instance of p-Split-MMO(3) constructed from (a).

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 $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$. We construct an instance G' of p-Split-MMO(3) by defining: (i) a set $U = \{u_1, u_2, \dots, u_n\}$ of n vertices, where each u_i corresponds to $v_i \in V$; and (ii) a set $W = \{w_1, w_2, \dots, w_m\}$ 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 $V(G_i^V) = \{u_{i,1}, u_{i_2}, \dots, u_{i,6}\}$ for each $i \in \{1, 2, \dots, n\}$ and $V(G_j^E) = \{w_{j,1}, w_{j_2}, \dots, w_{j,6}\}$ for each $j \in \{1, 2, \dots, m\}$. The vertex set of G' is thus $U \cup W \cup V(G_1^V) \cup V(G_2^V) \cup \dots \cup V(G_n^V) \cup V(G_1^E) \cup V(G_2^E) \cup \dots \cup V(G_m^E)$. 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 $e_j = \{u_h, u_i\} \in E$ for each $j \in \{1, 2, \dots, m\}$; (v) an edge $\{u_i, u_{i,h}\}$ 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 [3]. 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. □

Corollary 1. *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 2, 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 2. 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. □

5 Edge-Weighted Graphs, Bounded p

In this section, we prove that p-Split-MMO on edge-weighted graphs is weakly NP-hard even if restricted to $p = 1$. 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 [11].

Theorem 3. 1-Split-MMO *is weakly NP-hard even if the input is restricted to edge-weighted wheel graphs.*

Proof. We construct an edge-weighted, undirected graph $G = (V, E, w)$ from any given instance $S = \{s_1, s_2, \ldots, s_n\}$ of the Partition Problem. Define $K = \frac{\sum_{i=1}^{n} s_i}{2}$

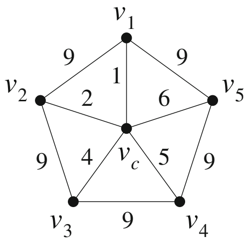

Fig. 4. Let $S = \{1, 2, 4, 5, 6\}$ be an instance of the Partition Problem. The reduction in the proof of Theorem 3 sets $K = 9$ and constructs the edge-weighted graph G above.

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 $v_1, v_2, \ldots, v_n$; and (ii) one special vertex, denoted by v_c. The edge set E consists of: (iii) the n edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_n, v_1\}$ forming a cycle; and (iv) the n edges $\{v_c, v_1\}, \{v_c, v_2\}, \ldots, \{v_c, v_n\}$ forming a star. (Hence, G is a wheel graph.) For every edge e of type (iii), assign $w(e) = K$. For every edge $\{v_c, v_i\}$ of type (iv), assign $w(\{v_c, v_i\}) = s_i$. An example is shown in Fig. 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 1-split orientation whose cost is at most K.

($\Rightarrow$) If there exists an $S' \subseteq S$ such that $\sum_{s_i \in S'} s_i = \sum_{s_j \in S \setminus S'} s_j$ then apply a split operation on the vertex v_c and let the two resulting vertices $v_{c,1}$ and $v_{c,2}$ be adjacent to the set of vertices of type (i) representing S' and $S \setminus S'$, respectively. For $i \in \{1, 2\}$, 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 $v_1 \to v_2 \to \cdots \to v_n \to v_1$. This way, the weighted outdegree of every vertex is at most K.

($\Leftarrow$) Let Λ' be a 1-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 $s_i \neq K$ for all $s_i \in S$ then we claim that the vertex in G to which the split operation was applied is v_c. To prove the claim, suppose the split operation was applied to some other vertex v_j, where $j \in \{1, 2, \ldots, n\}$, 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. Let the weight of this edge be s_k. Then the weighted outdegree of v_c is $2K - s_k > K$ because $s_i < K$ for all $s_i \in S$, contradicting that the cost of Λ' is at most K. This proves the claim. Now, since the split operation was applied to v_c (thus replacing v_c by two vertices $v_{c,1}$ and $v_{c,2}$) and the cost of Λ' is at most K, each of the n vertices in $\{v_1, v_2, \ldots, v_n\}$ has one of the n edges of weight K oriented away from it. This means that every edge of the form $\{v_{c,i}, v_j\}$ is oriented away from $v_{c,i}$, and since the sum of these edges' weights 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. □

Corollary 2. *For every fixed integer $p \geq 1$,* p-Split-MMO *on edge-weighted graphs is weakly NP-hard.*

6 Edge-Weighted Graphs, Unbounded p

Here, we prove that p-Split-MMO with weighted edges is strongly NP-hard if p is sufficiently large, i.e., $p = \Omega(n)$. 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 [11].

Theorem 4. p-Split-MMO *is strongly NP-hard even if the input is restricted to edge-weighted cactus graphs.*

Proof. We construct an edge-weighted, undirected 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 $B = \frac{\sum_{i=1}^{3n} s_i}{n}$ 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. 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.

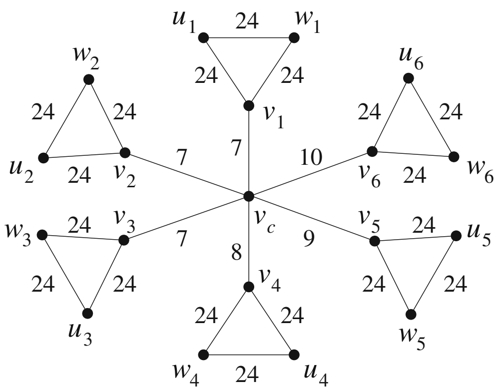

Fig. 5. An instance of the 3-Partition Problem with $S = \{7, 7, 7, 8, 9, 10\}$ and $B = 24$ yields the graph G shown above. In the construction, $n = 2$ and $p = 2 - 1 = 1$.

($\Rightarrow$) 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 (u_i, v_i), (w_i, u_i), and (v_i, w_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 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 since these edges must be oriented away from C and each of them has weight strictly larger than $\frac{B}{4}$, the weighted outdegree of x would be strictly larger than B, which is impossible. 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. $\square$

7 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. E.g., one could try to approximate the minimum maximum weighted outdegree for a value of p specified as part of the input, or approximate the smallest p for which some specified upper bound on the maximum weighted outdegree is attainable.

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.* [5] 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 [5] while its computational complexity for strongly connected orientations is still unknown.

Acknowledgments. The authors would like to 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. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Graph orientation to maximize the minimum weighted outdegree. Int. J. Found. Comput. Sci. **22**(3), 583–601 (2011)
2. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Graph orientations optimizing the number of light or heavy vertices. J. Graph Algorithms Appl. **19**(1), 441–465 (2015)
3. Asahiro, Y., Jansson, J., Miyano, E., Ono, H.: Degree-constrained graph orientation: maximum satisfaction and minimum violation. Theory Comput. Syst. **58**(1), 60–93 (2016)
4. Asahiro, Y., Jansson, J., Miyano, E., Ono, H., Zenmyo, K.: Approximation algorithms for the graph orientation minimizing the maximum weighted outdegree. J. Comb. Optim. **22**(1), 78–96 (2011)
5. Borradaile, G., Iglesias, J., Migler, T., Ochoa, A., Wilfong, G., Zhang, L.: Egalitarian graph orientations. J. Graph Algorithms Appl. **21**(4), 687–708 (2017)
6. Brodal, G.S., Fagerberg, R.: Dynamic representations of sparse graphs. In: Dehne, F., Sack, J.-R., Gupta, A., Tamassia, R. (eds.) WADS 1999. LNCS, vol. 1663, pp. 342–351. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48447-7_34
7. Chrobak, M., Eppstein, D.: Planar orientations with low out-degree and compaction of adjacency matrices. Theor. Comput. Sci. **86**(2), 243–266 (1991)
8. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. The MIT Press, Cambridge (1990)
9. Deming, R.W.: Acyclic orientations of a graph and chromatic and independence numbers. J. Comb. Theory Ser. B **26**, 101–110 (1979)
10. Fujito, T.: A unified approximation algorithm for node-deletion problems. Discrete Appl. Math. **86**(2–3), 213–231 (1998)
11. Garey, M., Johnson, D.: Computers and Intractability - A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
12. Khoshkhah, K.: On finding orientations with the fewest number of vertices with small out-degree. Discrete Appl. Math. **194**, 163–166 (2015)
13. Orlin, J.B.: Max flows in $O(nm)$ time, or better. In: Proceedings of STOC 2013, pp. 765–774. ACM (2013)
14. Venkateswaran, V.: Minimizing maximum indegree. Discrete Appl. Math. **143**(1–3), 374–378 (2004)
15. Zuckerman, D.: Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. Theory Comput. **3**(1), 103–128 (2007)